Scheduling in Service Systems with Impatient Customers and Insights on Mass-casualty Triage

Evin Uzun Jacobson

A thesis submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research.

Chapel Hill
2010

Approved by

Nilay Tanık Argon

Vidyadhar G. Kulkarni

Adam Mersereau

Haipeng Shen

Serhan Ziya

Abstract

EVIN UZUN JACOBSON: Scheduling on Service Systems with Impatient Customers and Insights on Mass-casualty Triage
(Under the supervision of Nilay Tanık Argon)

In this research, we study a resource allocation problem among competing customers who may differ in their tolerance for wait. If a customer waits longer than his/her tolerance for wait (which we call the "lifetime"), then he/she leaves the system without receiving any service. On the other hand, if a customer enters service, a random reward is earned. The decision maker knows the type of the customer, which determines the lifetime, service time, and reward distributions for that customer. The objective is to obtain dynamic scheduling priority policies that maximize the total (or average) reward collected.

Our motivation for this study is a resource allocation problem commonly observed in the aftermath of mass-casualty events, where the medical resources are overwhelmed with the nearly simultaneous arrivals of large numbers of patients. In such situations, the common practice is to first triage the casualties, i.e., categorize them into priority groups based on only the type of the injuries. In this dissertation, we study the benefits of taking into account the number of patients, the available resources, and the changes that occur with time while giving prioritization decisions during a mass-casualty event. We formulate the problem as a priority assignment problem for a queueing system with multiple types of impatient jobs (patients). We study the problem under two main scenarios: (i) the case with a fixed number of jobs to be cleared (no future arrivals), (ii) the case with job arrivals. In either case, the objective is to maximize the reward (either total or long-run average). For the clearing problem, we consider the multi-server case under the assumption that service times are identically distributed, and when we relax this assumption, we restrict our attention to the single server case. In the analysis of both cases, we use sample path methods and stochastic dynamic programming to characterize structures of "good" scheduling policies. For example, we show that a job is prioritized irrespective of the number of other jobs, if it comes from the job type that brings the highest reward and that has the shortest lifetime in some stochastic sense. We also provide analytical results that show how the optimal policy might depend on the state of the system when such conditions do not hold. Based on these partial characterizations of the optimal policy, we develop state-dependent and state-independent heuristic policies, and test the performance of these policies by a numerical study. Finally, we extend the clearing model by considering job arrivals after time zero and allowing jobs to change type while waiting in the queue.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Many service systems give prioritization decisions on a regular basis for the allocation of scarce resources among their customers to fulfill one or more objectives. In service systems such as call centers and health care systems where customers may renege if they wait too long, the tolerance for wait can be different for different customer groups. Taking into account these differences for delay intolerance as well as other customer characteristics (such as expected rewards brought) while assigning priorities could enhance the overall performance of the service system. In this research, we study the priority assignment problem for such service systems by modeling them as queueing systems with multiple classes of impatient customers.

In our most general queueing model, there are multiple identical parallel servers, each serving one customer at a time. Each customer has a certain tolerance for wait, which we call the lifetime of that customer. If customers wait longer than their lifetimes, then they abandon the system without receiving any service. The lifetimes of the customers are unknown to the decision maker, but the decision maker knows the type of the customers which determines the service time and lifetime distributions. A customer that is taken into service brings a random reward, and the reward distribution is type-dependent as well. We study this problem under two main scenarios: (i) the case with a fixed number of customers to be cleared (no future arrivals), (ii) the case with customer arrivals. In the first case, the problem is essentially a stochastic job scheduling problem, whereas in the second case the problem is more of a queueing control problem. In both cases, our objective is to determine optimal or near-optimal policies that maximize the reward (either total or long-run average) that the customers bring to the system.

The main motivation behind this research is a basic resource allocation problem that is commonly encountered in the aftermath of a mass-casualty event. In the aftermath of mass-casualty events and disasters, critical resources such as ambulances, rescue vehicles, operating rooms, and physicians are typically overwhelmed by the sudden jump in demand for their services. In a matter of minutes to hours, these resources become insufficient in numbers to provide immediate relief to all that are in need and therefore their efficient allocation is essential for the eventual success of the emergency response effort. However, making these allocation decisions is a very difficult task as it requires simultaneous consideration of multiple factors. Furthermore, one needs to act fast as there is only a short period of time during which lives can be saved. The first step of a response effort is typically to determine (at least roughly) the urgency of different "jobs" to which the resources need to be assigned. (Here a job could be a single patient, a group of patients, or a rescue mission involving a large number of individuals.) Once that is done, one reasonable policy could be to start from the most urgent jobs and move onto less urgent ones as resources become available. However, what complicates the problem is that normally, the expected "payoff" from jobs at different urgency levels are different from each other. For example, in the case of mass-casualty incidents with traumatic injuries, most patients with shorter life expectancies have lower chances of going through a successful operation, i.e., lower expected payoffs. Furthermore, the service times of jobs at different urgency levels could be different as well. One of the objectives of this thesis is to investigate these trade-offs between urgency, payoff, and faster service and identify "good" resource allocation policies that are simple enough to be implemented during chaotic situations.

Triage, the practice of rationing medical resources depending on the severity of the patients' conditions, dates back to Napoleonic Wars. Since then it has been widely adopted not only in wars but also in civilian life in case of mass-casualty events or even in daily emergencies. In the medical literature, triage is defined as a brief clinical assessment that determines the order in which patients should be seen in the Emergency Department or, if in the field, the speed of transport and choice of hospital destination [65]. (In this research, we define triage as the decision process associated with determining the order which patients be served based on the information about the system.) There are several proposed and adopted triage systems in the emergency medicine literature. One common mass-casualty triage method is Simple Triage And Rapid Treatment (START), which separates the injured into four groups based on the type of the injury with each group marked by a color; see, e.g., Nocera and Garner [54]. However, to our knowledge, there has not been any comprehensive study on whether or not using these systems improves the outcome of emergency response efforts. In fact, more recently, adopted practices have been criticized for being too short-sighted. Several researchers from the emergency medicine community have argued that when making prioritization decisions, unlike the current practice, scarcity of the resources should be taken into account and called for more research on how that should be done (see, e.g., Frykberg [29] and Sacco [66]).

With this thesis, we aim to contribute to this discussion by providing insights on how resource limitations can be taken into account when determining patient priorities in mass-casualty events and the associated potential benefits. We do not attempt to develop a decision support tool that can readily be used in real time. The goal rather is to develop a relatively simple model that captures the most essential components of the decision problem, identify basic principles and rules-of-thumb that work well, and provide some guidance to the emergency response community in their efforts to devise practical and efficient policies.

Since the nature of the system under consideration includes factors that are hard to quantify such as loss of life, it is not easy to find an appropriate performance measure for the analysis of the underlying queuing model. Most of the Operations Research work that considers allocation of resources in health care systems defines the performance measure as the average utilization of resources or the queue waiting times. In the aftermath of a large-scale emergency event where the decision may involve life or death, these performance measures may not be appropriate. Therefore, we decided to let our performance measure be the expected reward that can be earned by serving patients. The reward associated with each patient can have various interpretations. If the objective of the emergency response effort is to save as many patients as possible, then the reward for a patient can be seen as the probability that the patient will survive when the required resource is provided. If the objective is to maximize the total QALY (Quality Adjusted Life Year) score, then the reward can be seen as the expected QALY that would be gained by allocating the resource to the patient. In case of prioritizing rescue missions, if the objective is to maximize the number of survivors, then the reward can be the number of disaster victims who would survive as a result of the associated rescue mission.

Although the triage problem is our main motivation, prioritization decisions can arise at various other applications and our results in this thesis are not exclusive to any specific setting. These applications include communication systems where data need to be transmitted by a given time [12]; and call centers where impatient customers change their patterns of waiting, e.g., customers may decide to abandon the queue before they receive service if their waiting time exceeds a certain threshold [17]. Hence, to keep the general appeal of our results, we adopt a general terminology throughout the thesis, which also allows us to emphasize the relevance of our findings to the classical scheduling literature. For example, we use “jobs” that are impatient instead of patients with finite lifetimes and “servers” that provide service to these jobs instead of ambulances or operating rooms. However, throughout the thesis, we will interpret our results and provide insights mainly within the context of prioritization decisions during emergency response to a disaster or a mass-casualty incident.

The outline and a brief summary of this dissertation are as follows. In Chapter 2, we provide a review of the literature on scheduling in clearing systems with and without deadlines, queues with reneging, and relevant work in emergency response. In this review, we observe that although the literature on queueing systems with impatient customers is vast, there are only a handful of articles on the dynamic prioritization of different classes of customers with different reneging patterns. In Chapter 3, we present our clearing model with multiple identical servers and a fixed number of impatient jobs that are initially present in the system. We consider two main trade-offs in the analysis of the clearing model: (i) lifetime vs. reward (urgency vs. payoff), and (ii) lifetime vs. service time (urgency vs. fast service). In Chapter 3, we study the first trade-off by assuming that service times are identically distributed for all jobs. This is a reasonable assumption for the patient triage problem when the service constitutes the transportation of the patients from the field to the hospitals. The analysis of the second trade-off is more challenging, but we were able to consider both trade-offs in the same model by restricting the number of servers to one. Our work on this is presented in Chapter 4.

In the theoretical analysis of our clearing model, we use sample path arguments, stochastic orders, and stochastic dynamic programming to characterize structures of “good” scheduling policies. In particular, we identify conditions under which some simple state-independent policies are optimal, and for the cases when these conditions are not satisfied, we show how the optimal policy depends on the system state. For example, one of our analytical results shows that when there are two types of customers with identical service times distributions, it is optimal to serve the class of customers who brings higher rewards when the total number of customers exceeds a certain threshold value. Based on the knowledge obtained from such analytical results, we develop easy-to-implement state-dependent heuristic policies that can be used effectively for patient triage. By means of numerical experiments, we compare these heuristics to the common practice and other proposed heuristics from the literature.

In Chapter 5, we study two main extensions to our base clearing model. Firstly, we allow arrival of jobs after time zero. Secondly, we consider the model where each customer’s lifetime consists of multiple stages and the decision maker knows which stage each customer is in at any given time. We use an approach similar to that used for the clearing model to obtain insights on efficient prioritization policies. Finally, we present our concluding remarks in Chapter 6.

CHAPTER 2

Literature review

In this chapter, we review the related literature in four main categories. Although some of the papers that we review fall into more than one of these categories, we will discuss them in only one category.

2.1 Scheduling in clearing systems

There is a vast literature on stochastic scheduling in clearing systems, where the objective is to determine the order of the processing of jobs that are all available at time zero so as to optimize certain performance measures. We here review only the most relevant work and refer the interested reader to a popular textbook on scheduling by Pinedo [60] for an overview of deterministic and stochastic scheduling in clearing systems and issues about implementing these models. Within the stochastic scheduling literature, we are aware of only four papers that discuss scheduling in a clearing system with impatient jobs. These articles are Argon, Ziya, and Righter [2], Glazebrook, Ansell, Dunn, and Lumley [31], Li and Glazebrook [51], and Childers, Visagamurthy, and Taaffe [21]. As we do in this thesis, these four articles seek a solution to the problem of allocating service capacity to impatient jobs in a setting where all jobs are present at time zero and no additional jobs are expected to arrive. However, our work differs from these four articles in a number of ways. One common difference is that they all consider models with a single resource while we allow the number of resources to be possibly more than one. This is an important generalization since in many emergency response settings there is usually more than one resource available (e.g., when the resources are ambulances). We next review these four related articles in more detail.

Among these four articles, the closest to our work is the one by Argon et al. [2]. The authors consider a formulation where patients who belong to one of two different types (which determine their lifetime and service time distributions) receive service from a single server. The objective is to determine the optimal policy that maximizes the total expected number of survivors. Along with a number of analytical results that characterize the optimal policy, the authors propose two state-dependent heuristic policies that give priority to jobs with smaller mean service times but longer mean lifetimes when the system is heavily congested. In this thesis, we consider formulations that generalize the model of Argon et al. [2] in several ways making it a much more realistic representation of the actual system. First, as we stated before, the number of servers can be greater than one. Second, patients can belong to more than two different types. Third, unlike in the model of Argon et al. [2] not all patients who receive service bring the same reward; the rewards may depend on the type of the patient. This generalization significantly enriches the model. For example, it allows us to incorporate survival probabilities that differ across patient types.

Although not motivated by priority decisions during emergency response, Glazebrook et al. [31] study a model that is highly relevant. Specifically, the authors consider a general job-scheduling formulation of a multi-class single server clearing system with impatient jobs having exponential lifetimes under the objective of maximizing the expected total reward accumulated. They propose a simple state-independent policy resembling the “$c\mu$ rule” and prove that this policy is asymptotically optimal in the class of non-preemptive policies as the death rates approach to zero, i.e., as the mean lifetimes go to infinity. The authors also provide a brief numerical study on the performance of the suggested policy. However, as Argon et al. [2] and Li and Glazebrook [51] demonstrate later, this simple policy does not perform well when death rates are sufficiently large.

Li and Glazebrook [51] consider a formulation that is very similar to that of Argon et al. [2] except that they allow more than two patient types. The objective of the work is developing a heuristic method that could be executed in real-time to produce a near-optimal solution. With this objective, the authors use the idea of applying a single-step of the policy improvement algorithm (for Markov decision processes) on the state-dependent policy proposed by Glazebrook et al. [31]. They also use a fluid approximation for computing value functions needed in the policy improvement algorithm. By a numerical study, the authors show that this method produces a solution that is close to the optimal performance.

In a numerical study, Childers et al. [21] consider a similar job-scheduling problem with impatient jobs with the motivation of ordering patients for transport in case of a health-care facility evacuation. In their model, the patients are classified into two types (critical and non-critical) and there is a final due date common to all patients. They study the problem under two objectives: maximizing the number of lives saved and minimizing the holding cost of patients. Consistent with the results by Argon et al. [2], Childers et al. [21] conclude that when the resources are severely limited, the evacuation should start with non-critical patients first and switch to critical patients as the number of patients in need decreases.

Finally, there are many other articles on traditional job scheduling problems but with jobs that do not renege from the system after their due date. See, for example, Boxma and Forst [15], Coffman, Flatto, Garey, and Weber [23], Emmons and Pinedo [27], Righter [63], Weber, Varaiya, and Walrand [74], and Weiss and Pinedo [76]. The articles by Boxma and Forst [15] and Emmons and Pinedo [27] are the most relevant as their models also have multiple servers and their objective is to minimize the weighted number of tardy jobs (i.e., jobs for which the deadline expires while waiting in the queue). In these models, "weights" can be seen as "rewards" in our formulation, but unlike our work, the weights of jobs are deterministic. Furthermore, the work by Boxma and Frost [15] differs from ours in that they consider only static policies under the assumption that the due dates are independent and identically distributed (i.i.d.). As only static policies are considered, all jobs are scheduled at time zero, and hence a tardy job can be taken into service although it is not optimal to do so. One of the results by Boxma and Frost [15] shows that if all due dates are i.i.d. and processing times are stochastically ordered, then the jobs with stochastically shortest processing times should be processed first. Emmons and Pinedo [27], on the other hand, consider dynamic scheduling policies as we do in this thesis. One of their results states that if the processing times are i.i.d., and the due dates are either i.i.d. or have the same value, then the optimal non-preemptive dynamic policy is to process the job with the largest weight. In Chapter 3, we prove a similar result but without the assumption on i.i.d. due dates and deterministic weights. They also investigate the system under preemptive service discipline. They prove that if the processing times are i.i.d. exponential random variables, and the due dates are independent and can be ordered according to their failure rates, then the optimal preemptive dynamic policy is to process the jobs in the increasing order of their failure rates.

2.2 Scheduling in queueing systems with deadlines

In this section, we first review five relevant papers that consider scheduling in multi-class queueing systems with deadlines. The main difference of this section from Section 2.1 is the arrival of customers after time zero. Bhattacharya and Ephremides [11, 13] assume that the stochastic due date of a job is announced upon the arrival of the job and show that a form of the “shortest-time-to-extinction” policy is optimal under certain conditions. Moreover, Pandelis and Teneketzis [56] establish sufficient conditions under which serving one type of job is optimal at all decision epochs. The studies of Bhattacharya and Ephremides [11, 13] and Pandelis and Teneketzis [56] differ from our work as the due dates are announced upon arrival and their performance measure of interest is the (expected) discounted tardiness (and/or earliness) and/or long-run average tardiness (and/or earliness) per customer. Finally, Liu [53] investigates the scheduling of a multi-class queueing system with deterministic deadlines by considering fixed and dynamic prioritization policies with preemption, and the performance measure is server utilization. In addition to the difference in the performance measure of interest, our work also differs from Liu [53] since we consider random deadlines (i.e., lifetimes).

Among the studies on scheduling in a single-class queueing system with random deadlines, the papers by Bhattacharya and Ephremides [12], Panwar, Towsley and Wolf [57] and Pinedo [59] are the most relevant to our problem mainly because the performance measure of interest in these papers is the (weighted) number of tardy jobs. Bhattacharya and Ephremides [12] show that under the assumption of i.i.d. lifetimes, i.i.d. service times, and i.i.d. interarrival times (that are all mutually independent), the “earliest-arrival” policy is optimal if the lifetime distribution has a non-decreasing failure rate. Panwar et al. [57] show that a form of the “shortest-time-to-extinction” policy is optimal under certain conditions if the due date of a job is known upon arrival, and they compare the performance of the “shortest-time-to-extinction” policy with the first-come, first-served policy for various scenarios. Pinedo [59] considers only list scheduling policies, i.e., the decision maker arranges all jobs into a list at time zero, and is not allowed to change this list thereafter. Hence, when a list scheduling policy is applied, all jobs (even those jobs that are tardy) are processed. It is shown that if the processing times of jobs are independent and exponentially distributed, their release dates (i.e., the times that the jobs are available for processing) are random, and their due dates are identically distributed, then the optimal static list policy sequences jobs in increasing order of mean processing times when the system has a single server.

2.3 Queueing systems with reneging

There is a vast literature on the analysis of queueing systems with impatient customers (reneging or abandonments). The most relevant one is the study by Down, Koole and Lewis [26]. They consider a single server Markovian queueing system with two types of customers and both types of customers have equal service rates. They analyze both the discounted holding cost minimization and the long-run average reward maximization problems, and they formulate each of the two problems as a continuous-time MDP. One of their objectives is to identify the cases where a static policy is optimal. Their main result states that if type 1 jobs have higher reneging rates and rewards, then it is optimal to serve type 1 jobs. We obtain a similar result for our clearing problem in Chapters 3 and 4, but our results hold with more generality as we allow more than two types of jobs, general lifetime and service time distributions, and multiple servers in Chapter 3, and type-dependent service rates in Chapter 4. By means of a numerical study, they identify the conditions under which the $c\mu$-rule's deviation from optimality is significant.

We are also aware of three studies on heavy traffic approximations of the multi-server multi-type queueing systems with impatient customers; namely, Atar, Giat and Shimkin [6], Ghamami and Ward [30], and Perry and Whitt [58]. Under the Markovian assumption, Atar et al. [6] propose a policy called "$c\mu/\theta$ rule," where c, μ, and θ denote the holding cost rate, the service rate, and the abandonment rate, respectively. They show that the $c\mu/\theta$ rule is asymptotically optimal for the long-run average cost minimization problem. They also provide a counterexample that shows that the $c\mu/\theta$ rule is not necessarily asymptotically optimal for a finite horizon version of the cost function. Ghamami and Ward [30] consider the dynamic control of a system with two job types and two parallel servers, one of which, namely server 2, can serve both types of jobs, and server 1 can only serve type 1 jobs. (This system is usually called the N-system.) Customers from each class arrive according to a renewal process and the lifetime of a customer is exponentially distributed. The main result shows the asymptotic optimality of a two-threshold policy that uses one threshold on the total number of customers and another threshold on type 1 jobs to determine which job server 2 should serve. Their objective is to minimize the expected infinite horizon discounted holding and reneging cost of jobs. Finally, Perry and Whitt [58] consider a multi-class Markovian queueing problem, where each class has a separate queue that is served by a pool of multiple servers. They approximate the problem by a deterministic fluid model and propose a policy that balances the workload by sharing a server pool among various queues when the workload is high.

They test the performance of the proposed policy using simulation.

There are other papers that examine queueing systems with two classes of customers, where only class-1 customers are impatient and have higher priority over the class-2 customers, see, e.g., [17, 19, 22, 40, 3, 4]. These papers assume preemptive service, and therefore the dynamics of class-1 customers are not affected by class-2 customers. Thus, the dynamics of class-1 customers reduce to a single class of customers with impatience, which is well investigated by several authors (partly as special cases of more general models), see, e.g., [7, 8, 9, 14, 16, 18, 37, 73] and references therein. Moreover, in some of these studies, a customer may abandon the system not only while waiting for service, but also during his/her service, see, e.g., [9, 22, 37]. In that case, some of the service will not be useful. Among the work that examines the characteristics of class-2 customers, Choi et al. [22] study an $M/M/1$ queueing system, where class-1 customers have impatience of constant duration. The main results are on the stability condition, the probability generating function of the distribution of number of class-2 customers, and Laplace-Stieltjes transform of the sojourn time of class-2 customers. Brandt and Brandt [19] generalize this model by considering impatience with a general distribution. They obtain the distribution of the number of customers in service or in class-1 queue. They develop an approximate method for obtaining the moments of the number of customers in class-2 queue. Furthermore, Brandt and Brandt [17] analyze the case where class-1 customers may join the class-2 queue or leave the system if the random maximal waiting time exceeds a given deterministic time. They propose a birth-and-death queueing model for a call center with impatient class-1 customers and patient class-2 customers. If class-1 customers wait in the queue beyond a given threshold, they become class-2 customers. Class-2 customers are served when no class-1 customers are waiting and the number of idle agents exceeds a threshold. Iravani and Balcıoğlu [40] consider three separate problems. In the first problem, all jobs are impatient and the server follows a preempt-resume policy, and in the second problem, only the high-priority class customers are impatient and their service is performed in a non-preemptive manner. In the third problem, there are multiple servers and in addition to leaving the system due to reneging, a customer can leave the system without joining the queue (balking) if he/she knows his/her prospective waiting time upon arrival, and this time exceeds the maximal waiting time. Our work differs from the studies reviewed in this paragraph, as they analyze the performance measures of interest for a fixed policy, whereas we aim to characterize the optimal policy for the performance measure of interest.

Similar to the third problem in [40], the articles by Armony and Maglaras [3, 4] consider a multiple-server problem where the system provides information about waiting times upon arrival, and after receiving the information, customers can balk, join the high-priority queue, or request a call-back. The information provided upon arrival includes the waiting time in the high-priority queue and a guaranteed amount time within which the system will call them back. With an objective of minimizing the delay in the high-priority queue, Armony and Maglaras [3] show that the proposed policy is better than the policy that only gives the information on the waiting time in the high-priority queue asymptotically. Furthermore, Armony and Maglaras [3] investigate the optimal staffing levels that satisfies a set of constraints on the system performance under heavy traffic regime. Finally, in addition to two-class priority queueing models that are discussed in [40, 3, 4], there are other queueing systems in the literature where customer balking is investigated, see, e.g., [5, 44, 77]. Additionally, other types of departures from the system without service completion can be due to admission and expulsion decisions. For examples of this work, see [20, 42, 45, 64, 78, 79, 80] and the references therein.

2.4 Operations Research work on emergency response management

Even though patient triage has long been practiced, interestingly, there has not appeared any comprehensive study on how useful existing triage systems are or in fact whether or not triage is useful at all (Jenkins et al. [41] and Lerner et al. [50]). More recently, a number of authors (e.g., Frykberg [29]) discussed the limitations of existing practices and argued in support of making triage and priority decisions while taking into account resource limitations. However, to the best of our knowledge, there is only one work from the emergency medicine literature (Sacco et al. [66, 67]) that proposes a prioritization method (called the *Sacco Triage Method* (STM)) that takes into account system conditions. More specifically, Sacco and his coauthors propose a linear-programming-based method for determining priorities when dispatching patients to the hospitals. In their model, patients are categorized into twelve criticality levels upon arrival, the planning horizon is divided into a fixed number of periods, and a decision about which patients to transport to the hospital is made at the beginning of each time period. Transportation times are deterministic and patients become deterministically more critical with time. The survival probability of a patient depends on the criticality level at the given time. The idea is to solve a linear program at the beginning of the response effort and perhaps repeatedly thereafter as the conditions change. Then, the results are compared to START by means of a numerical study. Their results show that the difference in the performance of the proposed method and START is very small when patients are not very critical but the difference becomes significant when patients become more critical. Moreover, when the resources are overwhelmed, the less critical patients with higher survival probabilities should be given priority. Then, their main conclusions state that the current procedures of triage do not take into account resource limitations and have too few categories, so patients have very different survival probabilities within a category, and they point out that a method that better predicts the condition of the patient and considers resource limitations while giving prioritization decisions is needed to improve the expected number of survivors. In addition to the fact that STM largely ignores the randomness inherent in the actual system, the method has been criticized as being impractical as it suggests using a real-time solution, which might differ drastically from one event to the other, and it highly relies on perfect system information and communication within the disaster area; see Cone and MacMillan [24]. Our objective is not to propose a real-time solution method like STM, but instead to identify basic rules and principles that the emergency response community can use in the development of simple and effective prioritization policies.

Although not very relevant, we would like to mention that in the context of emergency response planning, excluding patient triage, there is also some early work that used multi-server queueing models for optimal dispatching of police patrols in New York City, see [32, 33, 35, 48, 68, 69]. Green [32] proposes a multiple-car dispatch model. The model is a multi-server, multi-priority Markovian queueing model, and the number of servers needed by each type of service call is given by a probability distribution. The service times are assumed to be i.i.d., and the performance measures of interest include the probability of delay, mean delay for each type of call, and the average number of available servers. The comparison of this model with several other queueing models is discussed in Green and Kolesar [33]. Furthermore, in the study by Green and Kolesar [35], the validity of this model is tested. Schack and Larson [68, 69] also consider a multi-server, multi-priority queueing system motivated by dispatching of police patrol cars, assuming that the service times are i.i.d. They derive some system statistics including the waiting time distributions for each type of calls. Another study by Larson [48] investigates the effects of increasing the service area of police patrols and concludes that travel times do not necessarily increase when the service area increases, especially at the medium utilization of police patrols.

For comprehensive reviews of the Operations Research work on emergency response, the interested reader is referred to [1, 36, 46, 49, 72].

CHAPTER 3

Scheduling of impatient customers in a clearing system with multiple servers and i.i.d. service times

In this chapter, we investigate a problem that is similar to a traditional job scheduling problem although with some important differences. Very broadly, the problem can be described as follows: There are different types of jobs each having a stochastic due date, which is unknown to the decision-maker, and an associated expected reward that will be earned if the job is taken into service before its due date. Each job has a stochastic processing time distributed identically for all jobs. The objective is to maximize the total expected reward by dynamically determining the order according to which jobs will be processed.

The outline of the chapter is as follows. We start with our model description in Section 3.1. In Section 3.2, we use a sample-path argument to show that urgent jobs that bring high rewards should be prioritized at all times. In the absence of such a condition, we need to make other simplifying assumptions for analytical tractability. Hence, in Sections 3.3 and 3.4, we assume that service time and lifetime (time until the due date) for each job are exponentially distributed, and then formulate the problem as a stochastic dynamic program. Using this formulation, we prove several structural results that characterize the optimal policy under certain conditions. These analytical results not only help us generate useful insights on the characteristics of "good" policies but also provide analytical support for the development of three heuristic methods that we propose in Section 3.5. Finally, in Section 3.6, we test the performance of our heuristic policies by means of a numerical study and observe that it is possible to design simple policies that perform well.

3.1 Model description

In our model, we assume that at time zero there are N jobs that are in need of receiving service from one of the M identical parallel servers, where $N > M \geq 1$. (The problem is trivial when $N \leq M$.) Jobs are impatient in the sense that if a job's waiting time in the queue exceeds its "lifetime," it reneges, i.e., it leaves the system without receiving any service. Jobs that enter service do not renege while in service. We assume that there will not be any future job arrivals so that the problem is over as soon as all of the N jobs in the system are cleared either after they receive service or after their lifetime expires. A job that is taken into service brings a random reward.

In the context of a mass casualty event, jobs can refer to any group of tasks that require the same set of scarce resources during an emergency response effort. For example, in case of a bombing, jobs can be injured patients who are waiting to be transported to a hospital; or in case of a natural disaster, they can be already hospitalized patients who are waiting to be transferred to safer locations from areas affected by the disaster. In these two examples, the scarce resource would be ground or air transportation vehicles. Similarly, jobs can be patients with traumatic injuries that are brought to a hospital following an emergency event and the scarce resource can be the operating rooms of the hospital. In each one of these cases, there is a random due date for each job since patients can die before they are safely transported and/or provided with the required medical care. Moreover, the reward of a job can be seen as the probability that the patient will survive after the given service or the patient's QALY. In the case of prioritization of rescue missions, where a limited resource needs to be allocated among several rescue missions, the reward can be seen as the number of potential survivors associated with the mission.

Each job in the system is characterized by its lifetime and reward distribution. We assume that the service times for all jobs are i.i.d. One setting where this assumption would be perfectly reasonable is when determining priorities for patients who need to be dispatched to a specific hospital from the disaster area via ambulances. In such a situation, transportation times are not expected to depend on the type of patients. We also assume that the service is performed in a non-preemptive manner, i.e., once a server starts processing a job, it cannot start working on another job before completing the processing of the job that is already in service.

Let Y_i be the lifetime of job i at time zero, Z_i be the non-negative reward earned when job i is taken into service, and S_i be the service time for job i for $i = 1, \ldots, N$. We assume that $\{Y_i\}_{i=1}^{N}$,

$\{Z_i\}_{i=1}^N$, and $\{S_i\}_{i=1}^N$ are sequences of independent random variables and that these three sequences are independent from each other. We also assume that $\{S_i\}_{i=1}^N$ is a sequence of identically distributed random variables. We let Π be the set of all dynamic and non-preemptive scheduling (prioritization) policies. Here, a dynamic prioritization policy is a collection of rules that determine which job is taken into service at any given decision epoch based on the state of the system, i.e., the time of the decision epoch and the collection of jobs in the system. We also define $C_\pi(t)$ to be the total reward earned by time $t \geq 0$ when policy $\pi \in \Pi$ is applied. Our objective is to identify characteristics of policies that either maximize $C_\pi(t)$ stochastically or its expectation by the time the system is cleared.

In Sections 3.2, 3.3, and 3.4, we study the characteristics of the solution to this optimization problem. Before we proceed with the analysis, we first note an intuitive result, which is proved in the Appendix.

Proposition 3.1.1. *Any idling policy is suboptimal in the sense of maximizing $C_\pi(t)$ along any given sample path.*

Based on Proposition 3.1.1, in the rest of the chapter, we only consider non-idling policies. Note that since idling can never be optimal and preemption is not allowed, the decision epochs for our dynamic control problem are time zero and service completion instants. At time zero, all N servers are available and hence the decision is to assign all these servers to jobs. From then on, new jobs are allocated at service completion instants, i.e., at times when servers become available.

3.2 When more urgent jobs have higher rewards

We first study settings where jobs with earlier lifetimes (and thus are more urgent) have higher associated rewards. In this section, we do not make any distributional assumptions on service times, lifetimes, and rewards, and our objective is to maximize $C_\pi(t)$ stochastically. We start by providing the definitions of three stochastic orders used throughout this dissertation.

Suppose that X and Y are two random variables that are either discrete or continuous. If $\Pr\{X > u\} \leq \Pr\{Y > u\}$, for all $u \in (-\infty, \infty)$, then X is said to be smaller than Y in the sense of *usual stochastic orders* (denoted by $X \leq_{st} Y$). On the other hand, if $\Pr\{X - v > u | X > v\} \leq \Pr\{Y - v > u | Y > v\}$, for all $u, v \geq 0$, then X is said to be smaller than Y in the sense of *hazard rate orders* (denoted by $X \leq_{hr} Y$). Finally, let $f(t)$ and $g(t)$ be the densities or probability mass functions of X

and Y, respectively. If $f(t)/g(t)$ is decreasing in t over the union of the supports of X and Y, then X is said to be smaller than Y in the sense of *likelihood ratio orders* (denoted by $X \leq_{lr} Y$). Note that $X \leq_{lr} Y \Rightarrow X \leq_{hr} Y \Rightarrow X \leq_{st} Y$. For more on these stochastic orders, see Shaked and Shanthikumar (2007).

Proposition 3.2.1. *Consider a decision epoch $t_0 \geq 0$ at which jobs i and j are available for service. If $Y_i \leq_{hr} Y_j$ and $Z_i \geq_{lr} Z_j$, then a policy $\pi \in \Pi$ that serves job j at time t_0 can be improved (in the sense of stochastically increasing $C_\pi(t)$ for all $t \geq t_0$) by serving job i instead of job j at time t_0.*

Proposition 3.2.1, which is proved in the Appendix, states that if the lifetimes and rewards of any two jobs can be ordered according to the hazard rate and likelihood ratio orders respectively, then giving priority to the job with a shorter lifetime and larger reward increases the total reward stochastically, and as a result, also in expectation. Thus, when determining which job to serve next, a job can be eliminated from consideration if it is dominated by another job whose lifetime is longer in the sense of hazard rate ordering and whose reward is smaller in the sense of likelihood ratio ordering. If these two orderings hold for any job pair, then the optimal policy can be completely characterized. Hence, Proposition 3.2.1 directly leads to the following result.

Corollary 3.2.1. *If $Y_1 \leq_{hr} Y_2 \leq_{hr} \cdots \leq_{hr} Y_N$ and $Z_1 \geq_{lr} Z_2 \geq_{lr} \cdots \geq_{lr} Z_N$, then a non-idling policy that prioritizes the job with the smallest index at every decision epoch maximizes $C_\pi(t)$ in the sense of usual stochastic orders for every $t \geq 0$.*

Corollary 3.2.1 indicates that giving priority to the job with the shortest lifetime (in the sense of hazard rate orders) maximizes the total reward earned if that job also brings the highest reward (in the sense of likelihood ratio orders).

Both Proposition 3.2.1 and Corollary 3.2.1 make intuitive sense as it is reasonable to believe that high-reward jobs with short patience times (e.g., patients with shorter life expectancies and higher survival probabilities) should get higher priority. These results are important in that they provide specific ordering conditions under which this intuition holds. In the context of emergency response, the results imply that if a group of patients have a higher chance of survival but a shorter life expectancy in terms of the stochastic orders given in Proposition 3.2.1 and Corollary 3.2.1, then that group should receive priority no matter how many resources (e.g., ambulances) are available and how many patients there are at any point in time during the response effort. However, at least in the case of mass-casualty triage, the situations where these conditions hold are not common since survival probabilities for patients with longer life expectancies are typically higher (see, e.g., Sacco et al. 2005). Therefore, in the following sections, we mainly focus on cases where more urgent jobs bring lower rewards.

3.3 When more urgent jobs have lower rewards

In most mass-casualty incidents, patients who have shorter life expectancies also have smaller chances of survival. Thus, investigating priority decisions for the case where more urgent jobs bring lower rewards is crucial. However, in this case, even a partial characterization of the optimal policy appears to be very difficult if not impossible for general service time and lifetime distributions. If we assume that service times and lifetimes are exponentially distributed, then we can obtain partial characterizations of optimal policies and gain insights into policies that perform well. Furthermore, these characterizations lead to simple heuristic policies that can be used in non-exponential settings as well.

Our claim here is not that in reality (at least in scheduling problems that arise during emergency response efforts) service times or lifetimes are exponentially distributed. Although, to the best of our knowledge, no prior work has studied what particular distributions would be good fits, there is also no reason to expect that the exponential distribution would be a good choice neither for lifetimes nor service times. However, the assumption of exponentially distributed lifetimes and service times (which we refer to as the Markovian assumption) allows some mathematical analysis and helps us develop insights into what kind of policies are likely to work well in practice. In fact, as we demonstrate in Section 3.6, the heuristic methods that are developed based on our analysis of the Markovian case perform well even under settings when the exponential assumptions do not hold. Thus, the main insights that come out of our analysis appear to be valid under conditions that are more general than the Markovian setup we assume here.

In the following, we assume that jobs are classified into K different job types, each type being characterized by its mean lifetime and reward, where $2 \leq K \leq N$. These job types can be seen as triage classes for patients with different injury characteristics or more generally patients with different health conditions. Let $\mu > 0$ be the service rate for all jobs. Also, for $i = 1, \ldots, K$, let $r_i > 0$ be the abandonment rate (i.e., the reciprocal of the mean lifetime) and $\alpha_i > 0$ be the expected reward for a type i job. Finally, if we let Z_i denote the reward of a type i job for $i = 1, \ldots, K$, we assume that Z_i comes from a distribution such that $\alpha_i \leq \alpha_j$ implies that $Z_i \leq_{lr} Z_j$ for all $i, j \in \{1, \ldots, K\}$. We let $D_\pi(m_1, \ldots, m_K)$ be the expected total reward accumulated after all jobs are cleared when prioritization policy $\pi \in \Pi$ is applied and m_i jobs from type $i \in \{1, \ldots, K\}$ are initially in the system, where $\sum_{i=1}^K m_i = N$. We use a dynamic programming formulation to characterize the solution of the optimization problem, which is stated as

$$\max_{\pi \in \Pi} D_\pi(m_1, \ldots, m_K).$$

We define the state of the system with the vector $(\mathbf{q}; s)$, where $\mathbf{q} := (q_1, \ldots, q_K)$, q_i is the number of type i jobs waiting (excluding the ones in service), and s is the number of jobs in service. The decision epochs are time zero and service completion times at which there exist at least two job types say i and j such that $q_i, q_j \geq 1$. At time zero, the state of the system is $(m_1, \ldots, m_K; 0)$ and the decision is to determine the number of servers to be allocated to each job type, i.e., to determine the vector $\mathbf{n} := (n_1, \ldots, n_K)$, where $n_i \in \{0, 1, \ldots, m_i\}$ for $i = 1, \ldots, K$ and $\sum_{j=1}^K n_j = M$. On the other hand, at service completion times at which there is at least one job waiting for service, s will be equal to $M - 1$ and the decision is to determine the type of job to be allocated to the available server among those types for which $q_i > 0$ for $i = 1, \ldots, K$. We next present the dynamic programming equations.

Let $V(\mathbf{q}; s)$ be the value function at state $(\mathbf{q}; s)$, i.e., the maximum expected total reward starting from state $(\mathbf{q}; s)$. Also let $\mathbf{e}_i$ denote the vector with a one in the ith position and zeroes elsewhere and $\mathbb{I}_A$ denote the indicator function of event A, i.e., $\mathbb{I}_A = 1$ if A is true, and $\mathbb{I}_A = 0$ otherwise. The dynamic programming equations are given as follows:

$$V(m_1, \ldots, m_K; 0) = \max_{(n_1, \ldots, n_K) \in \Phi} \left\{ \sum_{i=1}^K \alpha_i n_i + V(m_1 - n_1, \ldots, m_K - n_K; M) \right\}, \tag{3.3.1}$$

where

$$\Phi = \left\{ (n_1, \ldots, n_K) : n_i = 0, 1, \ldots, m_i, i = 1, \ldots, K;\ \sum_{j=1}^K n_j = M \right\}.$$

For $\mathbf{q} \in \{(q_1, \ldots, q_K) : q_i = 0, 1, \ldots, m_i, i = 1, \ldots, K; \sum_{j=1}^K q_j \leq N - M\}$, we have

$$V(\mathbf{q}; M-1) = \max_{i=1,\ldots,K} \{\mathbb{I}_{\{q_i \geq 1\}} \alpha_i + V(\mathbf{q} - \mathbf{e}_i; M)\} \text{ and } \tag{3.3.2}$$

$$V(\mathbf{q}; M) = \frac{M\mu V(\mathbf{q}; M-1) + \sum_{i=1}^K q_i r_i V(\mathbf{q} - \mathbf{e}_i; M)}{M\mu + \sum_{i=1}^K q_i r_i}. \tag{3.3.3}$$

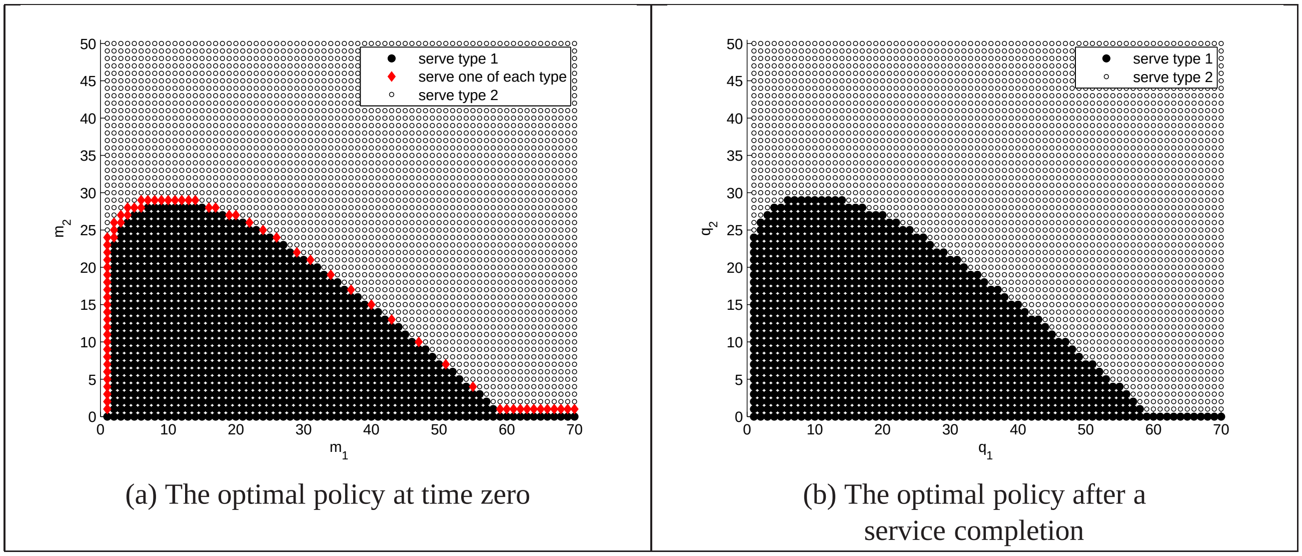
Finally, we let $V(\mathbf{q}; s) = 0$ if $\min\{q_1, \ldots, q_K, s\} < 0$, or $q_i = 0$ for all $i = 1, \ldots, K$ and $s = 0, 1, \ldots, M$.

In the remainder of this section, we use this dynamic programming formulation to obtain results on the characteristics of optimal policies. Without loss of generality, assume that $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_K$. From Corollary 3.2.1, we already know that if $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_K$ and $r_1 \leq r_2 \leq \cdots \leq r_K$, i.e., when the expected rewards and abandonment rates are agreeably ordered, then it is optimal to prioritize type K jobs. What if jobs with higher rewards do not necessarily have shorter lifetimes? How should we set priorities in that case? In this section and Section 3.4, we will provide some answers to these questions.

3.3.1 Structure of the optimal policy

In order to give the reader some idea about how the optimal policies look like in general we start with two examples. First, suppose that there are two types of jobs and two servers. Figure 3.1 presents the shape of the optimal policy for a specific example where $\alpha_2 > \alpha_1$ and $r_2 < r_1$, i.e., type 2 jobs have higher expected rewards and longer lifetimes. We selected this particular example since it demonstrates the general structure for the optimal policy that we observed from several numerical examples.

Figure 3.1 (a) shows the optimal allocation of the two servers at time zero for various values of m_1 and m_2. Figure 3.1 (b) demonstrates the optimal allocation of the server at a service completion instant for different values of q_1 and q_2. As it can be seen from both plots, the optimal policy gives priority to less time-critical jobs that bring a higher reward (i.e., type 2 jobs) when the number of jobs waiting is sufficiently large. To better understand the reason behind this, it might help to think of the extreme hypothetical case where there is an infinite supply of type 1 and type 2 jobs. In this case, one can see that it is always preferable to serve high-reward (type 2) jobs since there is simply no advantage in serving a



(a) The optimal policy at time zero

(b) The optimal policy after a service completion

Figure 3.1: The optimal policy for the case where $M = K = 2$, $\alpha_1 = 1.000$, $\alpha_2 = 1.001$, $\mu = 0.9009$, $r_1 = 0.9091$, and $r_2 = 0.9009$.

type 1 job instead. When there are fewer jobs however, delaying service to type 2 jobs becomes a better strategy since one can "afford" to serve at least some of the type 1 jobs before switching to type 2 jobs, which have longer lifetimes. In the context of emergency response, this observation suggests that giving priority to less time-critical patients with a higher survival probability might be better when there are many patients in need of treatment. When the number of casualties is significantly high and it is clear that a large percentage of them is likely to die because of resource limitations, it makes more sense to use the resources to serve those with higher rewards, e.g., those who are more likely to go through a successful service. However, when there are so few patients, then it makes more sense to give priority to those with shorter life expectancies even though the chances of saving them are smaller since there is enough time to get back to less time-critical patients later.

The optimal policy shown in Figure 3.1 does not possess some of the desired monotonicity properties that would make describing and determining optimal policies easier. In particular, for a fixed number of type 2 jobs, the optimal policy is not monotone in the number of type 1 jobs. For example, if there are 25 type 2 jobs, as the number of type 1 jobs decreases, the optimal decision switches from serving type 2 jobs to serving type 1 jobs and then back to serving type 2 jobs again.

For the second example, suppose that there are three job types, two servers, $\alpha_3 > \alpha_2 > \alpha_1$, and $r_3 < r_2 < r_1$. Thus, type 3 jobs bring the largest expected rewards and have the longest mean lifetimes while type 1 jobs bring the lowest expected rewards and have the shortest mean lifetimes. Figure 3.2 gives a rough description of the optimal policy for this example. As the reader can observe from the figure, the state space is divided into three regions. The first region is shaded in grey and is a polyhedron defined by corner points A, B, C, and O. The second region is another polyhedron defined by the corner points A, B, C, D, and E. Finally, the third region includes all the other remaining points in the state space. The optimal policy for this case prioritizes type i jobs at the end of every service completion if the state falls in the ith region, for $i = 1, 2, 3$. Hence, as in the previous example, when the number of jobs is large, the optimal policy prefers jobs with larger expected rewards and longer lifetimes, and when the number of jobs is small, the optimal policy prefers jobs with smaller expected rewards and shorter lifetimes.

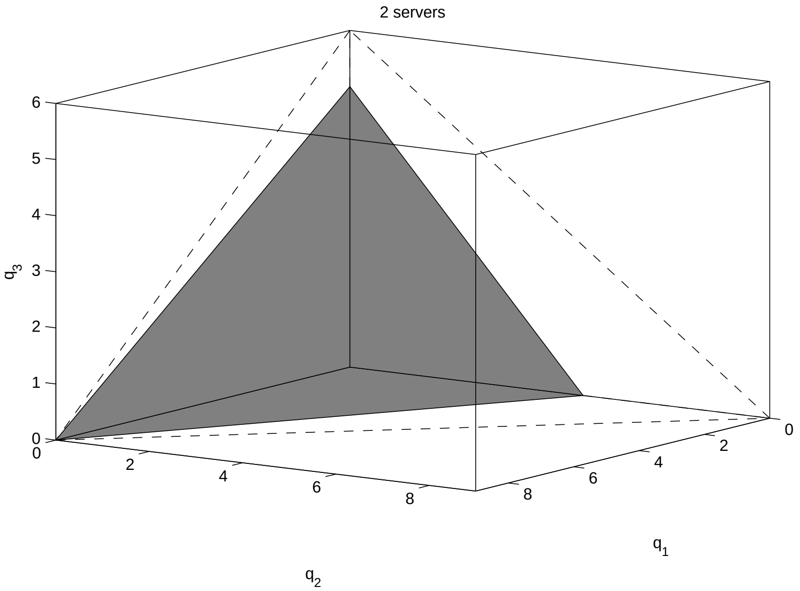


Figure 3.2: The structure of the optimal decisions at service completion instants for the case where $M = 2$, $K = 3$, $\alpha_1 = 1.0000$, $\alpha_2 = 1.0018$, $\alpha_3 = 1.0020$, $\mu = 0.9009$, $r_1 = 0.9091$, $r_2 = 0.9009$, and $r_3 = 0.9001$.

As the two examples clearly demonstrate, when the expected rewards and abandonment rates are not agreeably ordered, the optimal policy may be a state-dependent policy, i.e., a policy where the prioritization decisions depend on the number of jobs from each type that are waiting to receive service. However, even though the numerical computation of the optimal policy is straightforward, its structure can be quite complex, and therefore coming up with a complete characterization of the optimal policy appears to be a significant challenge. We can however show that the optimal policy possesses a particular type of monotonic structure under certain conditions. This structure is rigorously described in the following proposition, which is proved in the Appendix.

Proposition 3.3.1. *Consider a job type $j \in \{1, \dots, K\}$ and a state $(\mathbf{q}; M-1)$, where $q_j \geq 1$. Suppose that for all $k \in \{1, \dots, K\} \setminus \{j\}$ such that $q_k \geq 1$, an optimal action in state $(\mathbf{q} - \mathbf{e_k}; M-1)$ is to serve a type j job, and if $q_j \geq 2$, an optimal action in state $(\mathbf{q} - \mathbf{e_j}; M-1)$ is to serve a type j job. If*

$$(\alpha_j r_j - \alpha_i r_i) \sum_{k=1}^{K} q_k r_k + (r_i - r_j) \sum_{k=1}^{K} \alpha_k q_k r_k \geq \alpha_i r_i (M\mu - r_j) - \alpha_j r_j (M\mu - r_i) \tag{3.3.4}$$

and

$$r_i \geq r_j \text{ (when } K \geq 3\text{)} \tag{3.3.5}$$

for every $i \in \{1, \dots, K\} \setminus \{j\}$ such that $q_i \geq 1$, then an optimal action in state $(\mathbf{q}; M-1)$ is to serve a type j job.

Proposition 3.3.1 essentially says that under Conditions (3.3.4) and (3.3.5), it is optimal to give priority to a job of type j in a given state if it is also optimal to give priority to a type j job in all states with one less job of any particular type. Proposition 3.3.1 is important not only because it gives a partial characterization of the optimal policy but also because it serves as a backbone for the proofs of a number of insightful results on the structure of optimal policies (Propositions 3.4.1 and 3.4.2 in particular), which in turn form the basis for one of our heuristic methods described in Section 3.5. Furthermore, Proposition 3.3.1 leads to more sufficient conditions under which *index policies* are optimal, specifically Corollaries 3.3.1 and 3.3.2, and Proposition 3.4.4.

3.3.2 Optimality of index policies

An *index policy* is a set of state-independent decision rules that assign priorities based only on job types at any given state. Index policies have clear practical advantages over state-dependent policies. They are easier to implement since under such policies the priority relation among types of jobs does not change with time and system state, and also there is no need to keep track of the number of jobs from each type. In this section, we study index policies more closely.

In Proposition 3.2.1, we identified a set of conditions under which a job type should be prioritized over another regardless of the system state. For the Markovian case, these conditions imply that if $\alpha_i \leq \alpha_j$ and $r_i \leq r_j$, then job j should receive higher priority than job i at all decision epochs. We now identify a condition under which a job with the smallest abandonment rate receives the highest priority independently of the system state. Proofs of all propositions presented in this section are deferred to the Appendix.

Proposition 3.3.2. *Suppose that there exists a job type $j \in \{1, \ldots, K\}$ such that $r_i \geq r_j$ and $\alpha_i r_i \leq \alpha_j r_j$ for all $i = 1, \ldots, K$. Then, the optimal policy gives priority to type j jobs at all decision epochs.*

According to Proposition 3.3.2, non-urgent jobs can receive priority at all decision epochs regardless of the system state if their rewards are sufficiently high. This means that in the context of emergency response, if there is a particular type of patients, say type j, who have long life expectancies (i.e., $r_j \leq r_i$ for all $i = 1, \ldots, K$), they should nevertheless get the highest priority regardless of the system state if their expected reward is significantly large (i.e., $\alpha_j \geq \alpha_i r_i / r_j$ for all $i = 1, \ldots, K$).

Proposition 3.3.3. *Suppose that there exists a job type $j \in \{1, \ldots, K\}$ such that $\alpha_i \leq \alpha_j$ and $\alpha_i r_i \leq \alpha_j r_j$ for all $i = 1, \ldots, K$. Then, the optimal policy gives priority to type j jobs at all decision epochs.*

Proposition 3.3.3 states that jobs with the highest reward should receive priority at all decision epochs regardless of the system state if they also abandon the system at a sufficiently high rate. In the context of emergency response, this means that the type of patients, say type j, who bring the highest expected reward (i.e., $\alpha_j \geq \alpha_i$ for all $i = 1, \ldots, K$) should get the highest priority regardless of the system state if their life expectancies are significantly short (i.e., $r_j \geq \alpha_i r_i / \alpha_j$ for all $i = 1, \ldots, K$).

Propositions 3.3.2 and 3.3.3 also lead to complete characterizations of the optimal policy under two sets of conditions. More specifically, applying Propositions 3.3.2 and 3.3.3 multiple times, we obtain Corollaries 3.3.1 and 3.3.2, respectively.

Corollary 3.3.1. *If $r_1 \geq r_2 \geq \cdots \geq r_K$ and $\alpha_1 r_1 \leq \alpha_2 r_2 \leq \cdots \leq \alpha_K r_K$, then a non-idling policy that prioritizes the type of jobs with the highest index at every decision epoch is optimal.*

Corollary 3.3.2. *If $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_K$ and $\alpha_1 r_1 \leq \alpha_2 r_2 \leq \cdots \leq \alpha_K r_K$, then a non-idling policy that prioritizes the type of jobs with the highest index at every decision epoch is optimal.*

Corollaries 3.2.1, 3.3.1, and 3.3.2 provide us three sets of sufficient conditions that lead to the optimality of index policies. They are not necessary conditions however, and index policies might be optimal even when none of these conditions hold. Although it does not appear to be possible to identify necessary conditions, by applying a simple argument, we can characterize the structure of the "best" index policy given that there is an index policy that is optimal among all policies in Π.

Proposition 3.3.4. *If an index policy is optimal among all policies in* Π, *then it must give priority to the job with the largest value of* $\alpha_i r_i/(M\mu + r_i)$.

Proposition 3.3.4 describes the optimal policy under the condition that there is an index policy that is optimal. This condition does not hold in general as it can be clearly observed from Figures 3.1 and 3.2. However, this index policy can still perform well and thus can be used as a heuristic policy even though it may not be optimal. One important reason for expecting a reasonably good performance from this policy is that it is "myopically" optimal. To be specific, note that if a particular job is not taken into service at a decision epoch then the probability that it will not be available at the next decision epoch is $r_i/(M\mu+r_i)$ if it is of type i. Consequently, $\alpha_i r_i/(M\mu+r_i)$ can be seen as the "immediate opportunity cost" of not providing service to that particular job. The index policy given in Proposition 3.3.4 simply gives priority to the job with the largest immediate opportunity cost.

Remark 3.3.1. The index policies described in Corollaries 3.2.1, 3.3.1, and 3.3.2 all agree with the index policy identified in Proposition 3.3.4.

3.4 When more urgent jobs have lower rewards: The case with two types of jobs

In this section, we study a special Markovian case where jobs are categorized into two classes, i.e., $K = 2$. This simplification helps us push the analytical results further, get a better understanding of optimal policies, and develop heuristic methods of assigning priorities. More importantly, priority decisions during emergency response mainly concern two groups of jobs/patients. For example, according to START – a widely adopted triage system – the casualties are categorized into four groups but the most important decision concerns the priority ordering between critically injured patients who need to be taken care of as soon as possible (classified as *immediate*) and those who also have serious injuries but can wait a little longer (classified as *delayed*). Other patients, i.e., those with minor injuries (classified as *minor*) and those with injuries that are so severe that chances of survival are almost zero (classified as *expectant*), have the lowest priority. It is clear that as long as patients are correctly classified, there is no point in giving priority to either minor patients or expectant patients. However, the priority decision between the immediate and delayed patients is not clear at all. Even though the general understanding (and the current practice) is that immediate patients should have a higher priority than delayed patients, some in the emergency response community (e.g., Frykberg 2002) have suggested that this decision should ideally depend on the number of casualties and the scarcity of the available resources.

We start by assuming without loss of generality that $\alpha_2 \geq \alpha_1$. When $\alpha_2 = \alpha_1$, Proposition 3.2.1 implies that it is optimal to serve the job with the highest abandonment rate, and hence it is sufficient to only consider the case where $\alpha_2 > \alpha_1$. Proposition 3.2.1 also characterizes the optimal policy when $\alpha_2 > \alpha_1$ and $r_2 \geq r_1$. Hence, in this section, we will only focus on the case where $\alpha_2 > \alpha_1$ and $r_2 < r_1$, i.e., type 1 jobs have shorter life expectancies and thus are in more critical condition and their expected rewards are smaller.

Figure 3.1 presents a typical shape for the optimal policy when $K = 2$. The figure suggests that in general the optimal policy divides the state space into two regions separated by a single curve. A complete characterization of this curve, i.e., a complete description of the optimal policy, does not seem to be possible under all cases. Therefore, our objective here is to identify some structural properties of the optimal policy, with the ultimate goal of developing heuristic policies that nicely approximate the optimal policy, i.e., the curve that separates the two regions in Figure 3.1. Now, since it appears that the optimal policy has a lot to do with the total number of jobs waiting to be processed, a reasonable and also easy-to-implement policy would be of the form: serve a type 1 job if the total number of patients $q_1 + q_2$ is less than or equal to some threshold and serve a type 2 job otherwise. It is clear from Figure 3.1 that such a policy is not optimal in general. However, one can also see that if that threshold is carefully chosen, such a policy has the potential to be a reasonable alternative to the optimal policy. With this motivation, we next identify conditions under which the optimal action can be determined by simply comparing the total number of jobs with a threshold value.

In the following two propositions, we show that optimal actions at time zero and at service completion instants can be partially characterized by two thresholds. We first present the threshold result for service completion instants. The proofs of all our results in this section are provided in the Appendix.

Proposition 3.4.1. *Suppose that $K = 2$ and $\alpha_2 > \alpha_1$.*

(i) *There exists a threshold*

$$T_1 = \frac{M\mu(\alpha_1 r_1 - \alpha_2 r_2)}{(\alpha_2 - \alpha_1) r_1 r_2} + 1 \tag{3.4.1}$$

such that at all states $(\mathbf{q}; M-1)$, where $q_1 + q_2 \leq T_1$ and $q_1, q_2 \geq 1$, type 1 jobs are prioritized under the optimal policy.

(ii) *If there exists a positive integer $T \geq T_1$ such that at all states $(\mathbf{q}; M-1)$, where $q_1 + q_2 = T$ and $q_1, q_2 \geq 1$, it is optimal to give priority to type 2 jobs, then it is also optimal to prioritize type 2 jobs at all states $(\mathbf{q}; M-1)$ such that $q_1 + q_2 > T$ and $q_1, q_2 \geq 1$.*

To see how Proposition 3.4.1 partially characterizes the optimal policy, we first define

$$\begin{aligned} T_2 \quad = \quad & \inf\{T : T \geq T_1; \alpha_1 + V(\mathbf{q} - \mathbf{e}_1; M-1) \leq \alpha_2 + V(\mathbf{q} - \mathbf{e}_2; M-1), \\ & \quad \forall q_1, q_2 \geq 1, q_1 + q_2 = T\}, \end{aligned} \tag{3.4.2}$$

with the convention that $\inf \emptyset = \infty$. In other words, T_2 is the smallest T that satisfies the condition given in part (ii) of Proposition 3.4.1 if there exists such T; otherwise T_2 is set to infinity. Note that T_2 is always larger than or equal to T_1. We can now see that Proposition 3.4.1 implies that the optimal policy can be characterized partially by at most two thresholds: When the total number of jobs is below T_1, giving priority to type 1 jobs is optimal; and when it exceeds T_2, giving priority to type 2 jobs becomes optimal. Only when the total number of jobs $q_1 + q_2$ is between T_1 and T_2, we do not know what the optimal action is. Hence, Proposition 3.4.1 partially characterizes the optimal structure observed in Figure 3.1 (b), where T_1 is approximately equal to 17.06 and $T_2 = 59$. More importantly, Proposition 3.4.1 provides analytical support to our observation that when the number of patients in need of treatment is large and resources are highly loaded, giving priority to patients who have higher chances of survival might be more preferable.

We can also obtain a similar threshold result for the decision given at time zero, which partially characterizes the structure of the optimal policy observed in Figure 3.1 (a).

Proposition 3.4.2. *Suppose that $K = 2$ and $\alpha_2 > \alpha_1$.*

(i) *If $N \leq T_1 + M - 1$, where T_1 is given by Equation (3.4.1), then the optimal policy allocates as many servers as possible to type 1 jobs at time zero.*

(ii) *If $N \geq T_2 + M - 1$, where T_2 is given by Equation (3.4.2), then the optimal policy allocates as many servers as possible to type 2 jobs at time zero.*

Propositions 3.4.1 and 3.4.2 provide partial yet simple characterizations of the optimal policy providing insights into patient prioritization decisions. The results clearly show that optimal priority decisions can be dependent on the scale of the mass-casualty incident, i.e., the total number of patients in need of treatment. Even though these characterizations do not describe the optimal policy completely, they could be very useful in practice due to their simplicity. For example, in the immediate aftermath of a mass-casualty event, it would be much easier and faster for emergency responders to estimate the total number of casualties rather than the number of casualties at each criticality level. Furthermore, as we demonstrate in Section 3.6, a heuristic policy developed based on Propositions 3.4.1 and 3.4.2, which simply use the total number of jobs to determine priority levels, performs surprisingly well. (See Section 3.5 for the description of this heuristic policy and others.)

In the remainder of this section, we present two sets of conditions under which we can completely characterize optimal policies, both of which turn out to be index policies. We first provide a sufficient condition on the total number of jobs in the system at time zero under which the optimal policy prioritizes type 1 jobs at all decision epochs.

Proposition 3.4.3. *Suppose that $K = 2$ and $\alpha_2 > \alpha_1$. If $N \leq T_1 + M - 1$, then the optimal policy prioritizes type 1 jobs at all decision epochs.*

Proposition 3.4.3 essentially provides a threshold value such that if the total number of patients immediately after the event is below this threshold, patients with smaller chances of survival should get the higher priority at all times. Thus, the standard ordering of the START triage method, which always gives priority to *immediate* patients with lower chances of survival, is reasonable when the scale of the event is relatively small.

Finally, we provide a sufficient condition under which the optimal policy always prioritizes type 2 jobs regardless of the number of jobs waiting for service.

Proposition 3.4.4. *Suppose that $K = 2$ and $\alpha_2 > \alpha_1$. If*

$$\frac{\alpha_2 r_2}{M\mu + r_2} \geq \frac{\alpha_1 r_1}{M\mu + r_1}, \tag{3.4.3}$$

then the optimal policy gives priority to type 2 jobs at all decision epochs.

Similar to Proposition 3.3.3, Proposition 3.4.4 implies that the jobs with the larger expected reward should receive higher priority regardless of the system state if they abandon the system at a rate high enough that Condition (3.4.3) holds. Note however that this condition on the abandonment rate of the job type with the highest expected reward is weaker than the one provided in Proposition 3.3.3. Finally, we would like to point out a subtle but important difference between Propositions 3.3.4 and 3.4.4. Proposition 3.3.4 states that *if there is an index policy that is optimal*, then priorities are determined by the indices $\alpha_i r_i/(M\mu + r_i)$. On the other hand, Proposition 3.4.4 does not assume that the optimal policy is an index policy; it says that if $K = 2$ and the index policy described in Proposition 3.3.4 is agreeable with the highest expected reward rule, then it is optimal.

3.5 Heuristic policies

In Sections 3.2, 3.3, and 3.4, we obtained partial characterizations of the optimal policy and also identified conditions under which simple state-independent policies are optimal. For the remaining cases where the optimal policy is not characterized completely, we develop simple heuristic rules that are expected to perform well under a variety of conditions. To be more specific, in this section, we propose two state-dependent heuristic policies, namely the *2-step* and *threshold heuristics*. These heuristics are designed based on our dynamic programming formulation and structural results (particularly Proposition 3.4.1) presented in Sections 3.3 and 3.4. We also propose an index policy, which we call the *myopic policy*, based on Propositions 3.3.4 and 3.4.4. We finally discuss two other index policies, namely the *$\alpha r\mu$-rule* and *time-critical-first rule*, which will later serve as benchmark policies in our numerical study.

Below, we describe these heuristic policies under the assumption that the service times and lifetimes are exponentially distributed. However, as we explain later in Section 3.6, they can be also applied in more general settings. When describing the heuristics in the following, we assume, without loss of generality, that $m_i \geq 1$ and $q_i \geq 1$ for all $i \in \{1, \ldots, K\}$ because when the number of jobs is zero for a job type, then the problem essentially reduces to a problem with one less job type.

1. **2-step policy:** At every decision epoch, this heuristic maximizes the expected total reward over the next two periods. (Here, the period means the time between two consecutive event completion

times.) More precisely, to obtain this heuristic, we solve the dynamic programming equations (3.3.1), (3.3.2), and (3.3.3) assuming that the problem horizon is of two periods length. This gives us the following policy: At time zero, pick the allocation $(n_1^*, \ldots, n_K^*)$ that attains the following maximum:

$$\max_{(n_1,\ldots,n_K)\in\Phi} \left\{ \sum_{i=1}^{K} \alpha_i n_i + \frac{M\mu \max_{j\in\{1,\ldots,K\}} \left\{\mathbb{I}_{\{m_j - n_j \geq 1\}} \alpha_j\right\}}{M\mu + \sum_{j=1}^{K}(m_j - n_j) r_j} \right\}.$$

Similarly, at a service completion, i.e., when the system is in state $(\mathbf{q}; M-1)$, serve type i^* such that

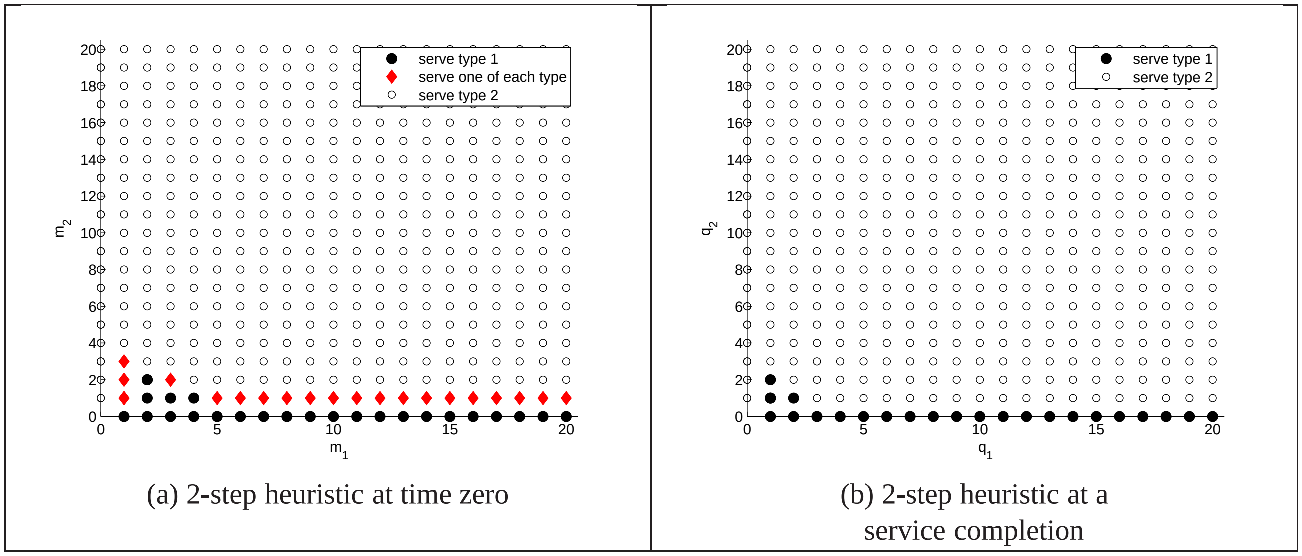
$$i^* = \arg \max_{i\in\{1,\ldots,K\}} \left\{ \alpha_i + \frac{M\mu \max\left\{\mathbb{I}_{\{q_i \geq 2\}}\alpha_i, \max_{j\in\{1,\ldots,K\}\setminus\{i\}}\{\alpha_j\}\right\}}{M\mu - r_i + \sum_{j=1}^{K} q_j r_j} \right\}.$$

(In case of ties, we arbitrarily let i^* be the smallest index that attains the maximum.)

Figure 3.3 shows the structure of the 2-step heuristic for the same experimental setting used in Figure 3.1. From the figure, we observe that for larger numbers of type 1 and 2 jobs, the heuristic prioritizes type 2 jobs, which is consistent with the optimal policy. Note however that the structure of the curves separating the state space differs between the 2-step heuristic and the optimal policy. Indeed, we can show that the curve that separates the state space into two regions under the 2-step policy at a service completion instant is a non-increasing function of q_1 when $K = 2$ and $r_1 > r_2$ (as in Figure 3.3 (b)), which is not true for the optimal policy as shown in Figure 3.1 (b).

2. **Threshold policy:** A quick examination of Figures 3.1 and 3.2 suggests that the optimal policy can possibly be approximated by a set of threshold values on the total number of jobs. For example, in Figure 3.1, a line that passes through points $(q_1 = 0, q_2 = 50)$ and $(q_1 = 50, q_2 = 0)$ could be used as the boundary between the set of states in which type 1 jobs are served and those in which type 2 jobs are served. This policy is clearly not optimal but it is expected to perform well.

   More generally, the heuristic policy we propose is described by (at most) $K-1$ thresholds $\{\mathcal{T}_1, \ldots, \mathcal{T}_{K-1}\}$, where $\mathcal{T}_1 \leq \mathcal{T}_2 \leq \cdots \leq \mathcal{T}_{K-1}$. It is specifically designed for the case where $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_K$ and $r_1 \geq r_2 \geq \cdots \geq r_K$ although it is possible to use it in other parameter regions as well. The heuristic works as follows: At a service completion, for $j = 1, \ldots, K$, type j jobs are prioritized if $\mathcal{T}_{j-1} < \sum_{i=1}^{K} q_i \leq \mathcal{T}_j$, where $\mathcal{T}_0 = -\infty$ and $\mathcal{T}_K = \infty$. Similarly, at time



(a) 2-step heuristic at time zero

(b) 2-step heuristic at a service completion

Figure 3.3: A sample structure for the 2-step heuristic when $\alpha_1 = 1.000$, $\alpha_2 = 1.001$, $\mu = 0.9009$, $r_1 = 0.9091$, $r_2 = 0.9009$, and $M = K = 2$.

zero, the threshold policy gives priority to type j jobs if $\mathcal{T}_{j-1} + M - 1 < N \leq \mathcal{T}_j + M - 1$, for $j = 1, \ldots, K$. To be more specific, if type j is the preferred type based on the thresholds, then M type j jobs are taken into service at time zero if $m_j \geq M$. Otherwise, the remaining $M - m_j$ servers are allocated to the job types with the closest index, starting from type $j + 1$ and continuing with type $j - 1$, type $j + 2$, and so on.

Now, the question is how one should pick the thresholds. We propose two different methods. In the first method, for each pair of job types i and j, where $j = 2, \ldots, K$ and $i = 1, \ldots, j - 1$, we compute

$$T_{i,j} = \frac{M\mu(\alpha_i r_i - \alpha_j r_j)}{(\alpha_j - \alpha_i) r_i r_j} + 1, \tag{3.5.1}$$

which is identical to the threshold expression given in Proposition 3.4.1. Then, we let

$$\mathcal{T}_j = \min\{\mathcal{T}_{j+1}, \max_{i \in \{1,\ldots,j\}} \{T_{i,j+1}\}\}, \text{for } j = 1, \ldots, K - 1. \tag{3.5.2}$$

In the second method, we use the 2-step policy to obtain $T_{i,j}$'s. To be specific, consider the 2-step policy when there are two types of jobs, namely type i and type j jobs. Then, the equation of the *switching curve* for the 2-step policy is given by

$$\alpha_i + \frac{M\mu \max\left\{\mathbb{I}_{\{q_i\geq 2\}}\alpha_i, \alpha_j\right\}}{M\mu + (q_i - 1)r_i + q_j r_j} - \alpha_j - \frac{M\mu \max\left\{\alpha_i, \mathbb{I}_{\{q_j\geq 2\}}\alpha_j\right\}}{M\mu + q_i r_i + (q_j - 1)r_j} = 0, \tag{3.5.3}$$

for $q_i, q_j \geq 1$. We let $q_i = 1$ in Equation (3.5.3) and solve for q_j (the largest solution is denoted by q_j^*); and similarly, we let $q_j = 1$ in the same equation and solve for q_i (the largest solution is denoted by q_i^*). If a solution is found for both equations, then we let $T_{i,j} = \max\{q_i^*, q_j^*\}$ based on our observation that the 2-step policy tends to underestimate the area under the switching curve for the optimal policy when $K = 2$. If a solution does not exist for one of the two equations, then we let $T_{i,j} = 0$. Finally, thresholds $\mathcal{T}_j$'s are determined using (3.5.2) as in the first method.

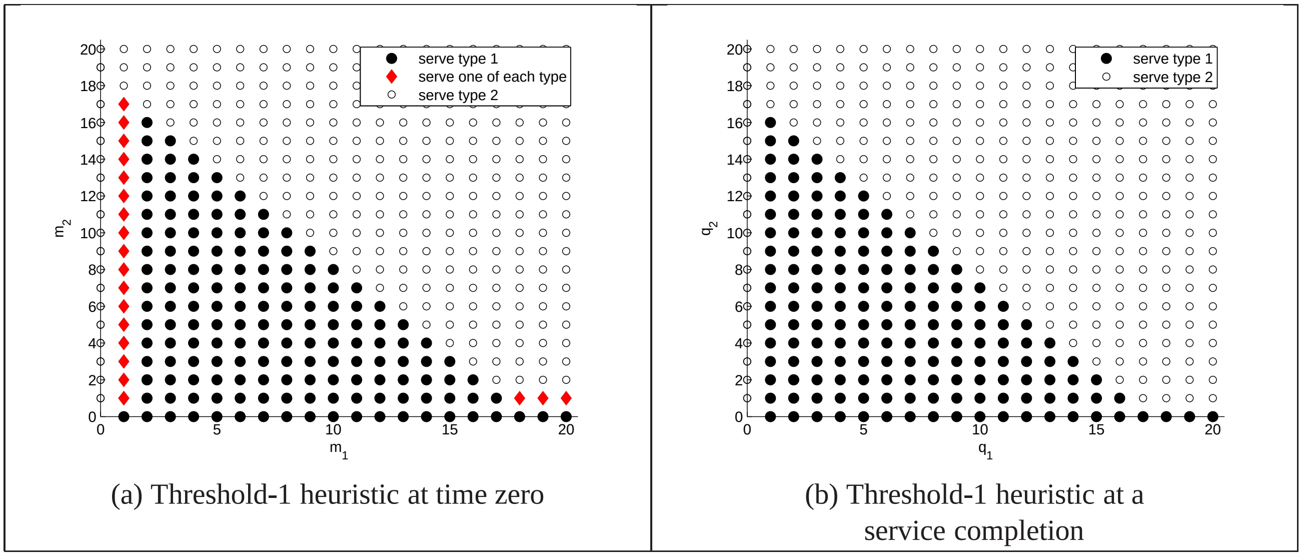
Thus, we have two different threshold-type policies depending on which method is used when computing the thresholds. When they are calculated using (3.5.1) [(3.5.3)], we call the policy the Threshold-1 [Threshold-2] heuristic.

One nice property of the threshold heuristic is its simple structure as it is completely characterized by at most $K - 1$ thresholds and the only required information is the total number of jobs in the system. In order to see the basic idea behind this heuristic, consider the simplest case where $K = 2$. Given Proposition 3.4.1, it would be reasonable to expect a relatively good performance from a policy that gives priority to more urgent jobs when the total number of jobs is below a certain threshold and to less urgent ones otherwise. This is precisely what the threshold policy does. When $K = 2$, the policy is defined by a single threshold value $\mathcal{T}_1 = T_{1,2}$ that divides the state space into two regions. As shown in Figure 3.4, the structure of the threshold policy is similar to that of the optimal policy in that the heuristic gives priority to type 2 jobs when the number of jobs in the system is large. The threshold policy simply generalizes this basic structure to any $K \geq 2$.

3. **Myopic policy:** This index policy, which is based on Proposition 3.3.4, prioritizes type i^* at all decision epochs where

$$i^* = \arg \max_{i\in\{1,\ldots,K\}} \left\{\frac{\alpha_i r_i}{M\mu + r_i}\right\}.$$

This policy can be seen as prioritizing the job with the largest "immediate opportunity cost" of not providing service. For more on the justification of this policy, see our discussion following



(a) Threshold-1 heuristic at time zero

(b) Threshold-1 heuristic at a service completion

Figure 3.4: A sample structure for the Threshold-1 heuristic when $\alpha_1 = 1.000$, $\alpha_2 = 1.001$, $\mu = 0.9009$, $r_1 = 0.9091$, $r_2 = 0.9009$, and $M = K = 2$.

Proposition 3.3.4.

4. $\alpha r \mu$**-rule:** This index policy, which is proposed by Glazebrook et al. (2004), prioritizes type i^* such that

$$i^* = \arg \max_{i \in \{1,\ldots,K\}} \{\alpha_i r_i \mu_i\},$$

where μ_i is the service rate for type i jobs. Glazebrook et al. show that when the lifetimes are exponentially distributed and there is a single server, the $\alpha r \mu$-rule is asymptotically optimal as abandonment rates approach zero. More specifically, if the abandonment rates are defined as $r_i = \theta \nu_i$ for all $i = 1, \ldots, K$, then the $\alpha r \mu$-rule is asymptotically optimal as $\theta \to 0$. For our case, where $\mu_i = \mu$ for all $i = 1, \ldots, K$, the $\alpha r \mu$-rule essentially becomes the αr-rule. Furthermore, when the service rates are equal for all jobs, the $\alpha r \mu$-rule and the myopic policy behave similarly under some additional conditions. To see this, consider the ratio of $\alpha_i r_i \mu_i$ to $\alpha_i r_i/(M\mu + r_i)$:

$$\frac{\alpha_i r_i \mu}{\alpha_i r_i/(M\mu + r_i)} = M\mu^2 + r_i \mu,$$

for $i = 1, \ldots, K$. This shows that the $\alpha r \mu$-rule and the myopic policy will behave similarly when r_i's are either very close to each other or very close to zero for all $i = 1, \ldots, K$. Moreover, using the asymptotic optimality of the $\alpha r \mu$-rule for small θ, we can conclude that the performance of the myopic policy will be very close to that of the optimal policy for small θ under the assumption that $M = 1$ and the lifetimes are exponentially distributed.

5. **Time-Critical-First (TCF) rule:** This index policy is based on the common practice for patient triage during daily emergencies that always gives priority to the most time-critical patients. To be more precise, this heuristic prioritizes type i^* such that

   $$i^* = \arg \max_{i \in \{1,\ldots,K\}} \{r_i\}.$$

   Although this rule is expected to perform poorly in general, we still include it in our numerical analysis due to its common use in daily triage.

Among the five heuristics described in this section, the TCF rule is likely to be the easiest to implement as it simply requires an ordering of the patients with respect to their remaining life expectancies. The $\alpha r \mu$-rule and the myopic policies are also simple policies although in addition to life expectancies these heuristics require estimates on “rewards” such as survival probabilities. In comparison, the 2-step and the threshold policies are more sophisticated since they both prescribe state-dependent rules. However, they are also relatively easy to implement, arguably among the simplest state-dependent policies which can be expected to perform well. One of the desirable aspects of these policies is that they do not use any distributional properties other than the mean values of remaining lifetimes and rewards, which means that they can be immediately adapted to settings where Markovian assumptions do not hold. The threshold policies are even simpler in that they only need to keep track of the total number of patients as opposed to the number of patients of each type.

Finally, in this section, we present a result that shows that all of the heuristics proposed in this chapter (i.e., the 2-step, Threshold-1, Threshold-2, and myopic policies) agree with the optimal policy for all conditions under which these heuristics are defined and we were able to characterize the optimal policy. The proof of Proposition 3.5.1 is provided in the Appendix.

Proposition 3.5.1. *Suppose that the Markovian assumption holds.*

(i) *If $r_1 \leq r_2 \leq \cdots \leq r_K$ and $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_K$, then the 2-step policy, myopic policy, $\alpha r \mu$-rule, and TCF rule are optimal.*[1]

[1] Threshold policies are not defined for these conditions.

(ii) *If $r_1 \geq r_2 \geq \cdots \geq r_K$ and $\alpha_1 r_1 \leq \alpha_2 r_2 \leq \cdots \leq \alpha_K r_K$, then the 2-step policy, Threshold-1 and Threshold-2 policies, myopic policy, and $\alpha r \mu$-rule are optimal.*

(iii) *If $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_K$ and $\alpha_1 r_1 \leq \alpha_2 r_2 \leq \cdots \leq \alpha_K r_K$, then the 2-step policy, Threshold-1 and Threshold-2 policies, myopic policy, and $\alpha r \mu$-rule are optimal.*

(iv) *If $K = 2$, $\alpha_1 < \alpha_2$, and $\alpha_1 r_1/(M\mu + r_1) \leq \alpha_2 r_2/(M\mu + r_2)$, then the 2-step, Threshold-1, Threshold-2, and myopic policies are optimal.*

3.6 Numerical results

In this section, we present our numerical results on the performance of the heuristics discussed in Section 3.5. We consider two cases: (i) the case where lifetimes and service times are exponentially distributed; and (ii) the case where lifetimes come from a Weibull distribution and service times are deterministic. In both settings, we can compute the performance of the optimal policy and compare it with those of the heuristic policies.

3.6.1 Exponential lifetimes and service times

In the first part of our numerical analysis, service times are exponentially distributed with rate one (i.e., $\mu = 1$) and lifetimes are exponentially distributed with rate $r_i > 0$ for type $i \in \{1, \ldots, K\}$ jobs. In order to test the heuristics under a variety of conditions, we generated some of the system parameters randomly. More specifically, we generated the initial numbers of jobs m_i, for $i = 1, \ldots, K$, independently and uniformly over the set $\{1, 2, \ldots, 100\}$ and the rewards α_i, for $i = 1, \ldots, K$, independently from a uniform distribution with range $[0, 1]$. We considered five subsets of experiments depending on the range of the abandonment rates r_i, for $i = 1, \ldots, K$, which are generated independently from a uniform distribution with ranges $[2.0, 5.0]$, $[0.5, 2.0]$, $[0.1, 0.5]$, $[0.01, 0.1]$, and $[0.005, 0.001]$. (The first [last] subset corresponds to the case where jobs are most [least] time-critical.) For each subset, we generated 5,000 random scenarios where $\alpha_1 < \cdots < \alpha_K$ and $r_1 > \cdots > r_K$. For every scenario, we calculated the expected total reward collected under all six heuristic policies and the optimal policy. Then, we computed the percentage deviation of the expected total reward under each heuristic from that under the optimal policy. Based on these 5,000 percentage deviations, we constructed a 95% confidence interval (C.I.) on the mean and determined the median and the maximum percentage deviation. We also calculated the number of times each heuristic provided the best performance among the six heuristics in each subset. The results for $M = K = 2$ and $M = K = 3$ are presented in Table 3.1.

Table 3.1: Performance of the heuristic policies (in terms of the percentage deviation from the optimal performance) when the service times and lifetimes are exponentially distributed and $m_i \sim$ Uniform$\{1,\ldots,100\}$ for $i = 1, \ldots, K$.

	$M = K = 2$				$M = K = 3$			
Heuristic	95% **C.I.**	**Median**	**Maximum**	$\#$ **of times best**	95% **C.I.**	**Median**	**Maximum**	$\#$ **of times best**
	r_i ~Uniform[2.0,5.0]							
2-step	0.03 ± 0.00	0.00	3.30	4423	0.13 ± 0.01	0.00	4.31	2573
Threshold-1	0.02 ± 0.00	0.00	1.95	4097	0.06 ± 0.00	0.02	3.47	2023
Threshold-2	0.44 ± 0.01	0.34	12.11	232	0.33 ± 0.01	0.21	5.40	40
Myopic	0.50 ± 0.05	0.00	14.42	4345	2.56 ± 0.11	0.31	20.49	2238
$\alpha r \mu$	2.39 ± 0.15	0.00	36.71	3667	9.89 ± 0.23	7.99	39.28	261
TCF	35.81 ± 0.72	32.62	98.35	90	36.93 ± 0.56	34.64	93.70	8
	r_i ~Uniform[0.5,2.0]							
2-step	0.22 ± 0.02	0.00	11.29	3679	0.66 ± 0.03	0.10	11.58	1018
Threshold-1	0.04 ± 0.01	0.00	4.55	4164	0.15 ± 0.01	0.01	6.30	3404
Threshold-2	0.35 ± 0.02	0.18	7.75	294	0.66 ± 0.03	0.18	10.21	38
Myopic	1.54 ± 0.10	0.00	21.39	3792	5.11 ± 0.17	2.72	29.13	1374
$\alpha r \mu$	2.80 ± 0.16	0.00	31.64	3385	8.66 ± 0.22	6.60	36.33	579
TCF	27.81 ± 0.63	22.95	95.64	308	27.71 ± 0.45	25.70	87.86	37
	r_i ~Uniform[0.1,0.5]							
2-step	0.83 ± 0.06	0.00	15.85	3115	1.83 ± 0.07	0.62	16.33	472
Threshold-1	0.08 ± 0.01	0.00	5.68	4465	0.23 ± 0.01	0.02	5.19	3908
Threshold-2	0.78 ± 0.05	0.03	13.81	468	1.78 ± 0.07	0.60	15.47	66
Myopic	1.32 ± 0.09	0.00	22.98	3749	3.64 ± 0.13	1.56	23.01	1574
$\alpha r \mu$	1.55 ± 0.10	0.00	25.54	3656	4.22 ± 0.14	2.06	23.78	1371
TCF	19.11 ± 0.52	13.26	86.46	743	17.47 ± 0.36	14.95	77.68	202
	r_i ~Uniform[0.01,0.1]							
2-step	2.06 ± 0.10	0.00	21.98	2666	3.21 ± 0.10	1.79	22.68	320
Threshold-1	0.04 ± 0.01	0.00	2.39	4631	0.07 ± 0.01	0.00	2.27	3996
Threshold-2	1.99 ± 0.10	0.00	21.36	2091	3.17 ± 0.10	1.77	21.72	270
Myopic	0.23 ± 0.03	0.00	11.48	4432	0.43 ± 0.03	0.00	9.45	3187
$\alpha r \mu$	0.24 ± 0.03	0.00	12.49	4421	0.46 ± 0.03	0.00	9.45	3147
TCF	6.86 ± 0.27	1.87	60.78	1888	4.87 ± 0.17	2.37	45.82	849
	r_i ~Uniform[0.005,0.01]							
2-step	0.19 ± 0.01	0.00	5.11	3931	0.49 ± 0.02	0.27	4.47	387
Threshold-1	0.00 ± 0.00	0.00	0.15	4930	0.00 ± 0.00	0.00	0.34	4475
Threshold-2	0.18 ± 0.01	0.00	5.11	3706	0.49 ± 0.02	0.27	4.47	383
Myopic	0.00 ± 0.00	0.00	0.76	4893	0.02 ± 0.00	0.00	0.79	4159
$\alpha r \mu$	0.00 ± 0.00	0.00	0.76	4892	0.02 ± 0.00	0.00	0.79	4147
TCF	4.04 ± 0.14	2.09	27.74	1029	2.91 ± 0.09	1.56	18.97	481

From Table 3.1, we observe that Threshold-1 achieves the best performance among all heuristic policies and across all parameters with a significant margin in some cases. The only exception is the case where $r_i \in [2.0, 5.0]$, i.e., when jobs are very time-critical. In this case, the 2-step policy has the smallest median and the 2-step and myopic policies yield the best performance in slightly higher numbers of scenarios. However, the Threshold-1 policy still provides a better average performance mainly because when the 2-step and myopic policies deviate from the optimal performance, the deviation is significant enough that their average performances are worse than that of Threshold-1. When we consider all the scenarios in this analysis, the Threshold-1 policy is at most 6.3% worse than the optimal policy. Thus, Threshold-1 policy not only performs well on the average but also appears to be robust with respect to changes in the system parameters.

Considering the other state-dependent heuristics, namely the 2-step and Threshold-2 policies, we see that they perform similar to one another for small abandonment rates but the 2-step policy is in general better when jobs are time-critical. However, neither of these two heuristic policies come close to the superior performance of the Threshold-1 policy (except in the first subset when jobs are very time-critical).

Among the three index policies considered, the myopic policy is the best across all parameter sets, and it is significantly better than the other two when jobs are time-critical. As expected, the $\alpha r \mu$-rule and the myopic policy provide near-optimal performances when the abandonment rates approach to zero.

In summary, Table 3.1 suggests that it is possible to find a simple and very effective policy (such as the Threshold-1 policy) that achieves a near-optimal performance across a variety of parameter regions by only using the information on the total number of jobs. It is especially important to use such a state-dependent policy when jobs are time-critical, i.e., their abandonment rates are high. When jobs are not time-critical, all policies (except for TCF) yield a performance similar to the optimal performance since regardless of which policy is used few jobs reach the end of their life while waiting to get service. Hence, when the abandonment rates are high, one could as well use one of the simple state-independent policies such as the myopic policy. Finally, if conditions do not allow using a state-dependent policy, regardless of whether or not abandonment rates are high, it might be best to choose the myopic policy since its performance is either comparable to or better (in some cases significantly better) than the performances of the alternative index policies.

3.6.2 Weibull lifetimes and deterministic service times

In this section, we test the performance of the heuristics discussed in Section 3.5 under a non-exponential setting. We assume that the lifetimes come from a Weibull distribution with shape parameter $\theta_i > 0$ and scale parameter $\beta_i > 0$. (Weibull is a commonly used distribution for modeling the lifetimes of humans; see, e.g., Section 2.2.2 in Hougaard 2000.) Then, the abandonment rates are given by

$r_i = \theta_i/[\beta_i\Gamma(1/\theta_i)]$ for $i = 1, \dots, K$, where $\Gamma(\cdot)$ is the gamma function. We assume that the service times are deterministic with $\mu = 1$, i.e., each service takes exactly one unit of time. The deterministic service time assumption allows us to compute the performance of the optimal policy.

We next discuss how we adapt the heuristics we described in Section 3.5 to this non-exponential setting. When lifetimes are not exponentially distributed, the abandonment rates change with time. When implementing the heuristics, one can either ignore that and simply use the abandonment rates of time zero at all times, or update them with time. In this study, we use the updated rates as in Argon et al. (2008). It can be shown that the updated abandonment rate for job type $i \in \{1, \dots, K\}$ at time $t \geq 0$ is given by

$$r_i(t) := \frac{\theta_i}{\beta_i\Gamma(1/\theta_i, (t/\beta_i)^{\theta_i})} e^{-(t/\beta_i)^{\theta_i}},$$

where $\Gamma(a, b) := \int_b^\infty u^{a-1}e^{-u}du$, for $a > 0$ and $b \geq 0$, is the incomplete gamma function. At each decision epoch after time zero, these updated abandonment rates are used instead of the initial abandonment rate r_i. (Note that $r_i(0) = r_i$.) Also, since the service times are equal to one time unit for all jobs, the decision epochs take place at times $0, 1, 2, \dots$, and all servers become available at every decision epoch. Hence, at all decision epochs where there are more than M jobs in queue, the decision is to determine which M jobs will be taken into service. Thus, in this deterministic-service setting, the heuristics use time-zero server allocation decisions (as described in Section 3.5) at every decision epoch.

For the numerical experiments, we set the initial number of jobs m_i to ten and let $\theta_i = 1.5$ for all $i = 1, \dots, K$. (Unfortunately, due to the computational complexity of this non-exponential case, we could not use the same experimental setting of Section 3.6.1.) We then generated the initial abandonment rate $r_i(0)$ from a uniform distribution with five different ranges: $[2.0, 5.0]$, $[0.5, 2.0]$, $[0.1, 0.5]$, $[0.01, 0.1]$, and $[0.005, 0.001]$. For each of these five subsets of experiments, we generated 5,000 random scenarios where $\alpha_1 < \cdots < \alpha_K$ and $r_1(0) > \cdots > r_K(0)$. (Since the shape parameter is the same for all types of jobs, having $r_i(0) > r_j(0)$ implies that $r_i(t) \geq r_j(t)$ for all $t \geq 0$, $i, j \in \{1, \dots, K\}$.) We computed the performance of each heuristic as in Section 3.6.1 and summarized the results for the cases with $M = K = 2$ and $M = K = 3$ in Table 3.2. We also repeated the experiments for the Markovian case with exponentially distributed service times and lifetimes under the same parameter settings in order to observe the effects of distributional assumptions on the performances of the policies. These results are presented in Table 3.3.

Table 3.2: Performance of the heuristic policies (in terms of the percentage deviation from the optimal performance) when the service times are deterministic, lifetimes come from a Weibull distribution, and $m_i = 10$ for $i = 1, \ldots, K$.

	$M = K = 2$				$M = K = 3$			
Heuristic	95% **C.I.**	**Median**	**Maximum**	$\#$ **of times best**	95% **C.I.**	**Median**	**Maximum**	$\#$ **of times best**
	r_i ∼Uniform[2.0,5.0]							
2-step	0.02 ± 0.01	0.00	3.47	4922	0.10 ± 0.02	0.00	6.71	4672
Threshold-1	0.01 ± 0.00	0.00	2.56	4116	0.02 ± 0.00	0.00	2.87	3895
Threshold-2	0.08 ± 0.01	0.00	3.57	1184	0.10 ± 0.01	0.00	6.71	1448
Myopic	0.35 ± 0.04	0.00	15.45	4628	0.92 ± 0.08	0.00	22.76	1085
$\alpha r \mu$	2.42 ± 0.18	0.00	40.68	3731	3.99 ± 0.23	0.00	45.18	1150
TCF	44.00 ± 0.75	43.56	99.10	78	58.44 ± 0.64	62.17	99.10	7
	r_i ∼Uniform[0.5,2.0]							
2-step	0.41 ± 0.04	0.00	9.74	4458	1.31 ± 0.07	0.03	15.50	3197
Threshold-1	0.25 ± 0.01	0.00	2.75	3559	0.68 ± 0.03	0.16	7.16	2337
Threshold-2	1.53 ± 0.03	1.33	5.76	74	2.90 ± 0.06	2.32	15.08	6
Myopic	1.02 ± 0.08	0.00	19.74	4173	2.09 ± 0.07	1.18	14.30	1506
$\alpha r \mu$	2.79 ± 0.15	0.00	29.14	3438	3.01 ± 0.10	1.83	23.93	999
TCF	32.84 ± 0.64	31.41	87.88	308	29.38 ± 0.46	30.08	74.28	190
	r_i ∼Uniform[0.1,0.5]							
2-step	0.77 ± 0.07	0.00	19.22	4394	1.97 ± 0.08	0.58	17.98	2868
Threshold-1	0.86 ± 0.05	0.00	10.74	3695	1.16 ± 0.05	0.27	9.34	3013
Threshold-2	0.75 ± 0.05	0.09	10.52	606	2.22 ± 0.07	1.28	14.57	361
Myopic	2.13 ± 0.14	0.00	25.67	3882	1.78 ± 0.05	1.10	12.20	1722
$\alpha r \mu$	2.82 ± 0.16	0.00	29.51	3675	1.95 ± 0.06	1.17	12.20	1640
TCF	28.63 ± 0.60	26.89	79.99	606	17.04 ± 0.27	16.75	49.16	300
	r_i ∼Uniform[0.01,0.1]							
2-step	0.22 ± 0.02	0.00	8.38	4812	0.07 ± 0.00	0.01	1.71	4049
Threshold-1	3.94 ± 0.20	0.00	39.14	3122	4.34 ± 0.17	0.14	31.47	2372
Threshold-2	0.20 ± 0.02	0.00	6.59	4397	0.28 ± 0.01	0.09	2.44	684
Myopic	4.79 ± 0.23	0.00	41.98	3083	4.59 ± 0.18	0.18	36.12	866
$\alpha r \mu$	4.94 ± 0.23	0.00	41.98	3051	4.66 ± 0.18	0.24	36.12	864
TCF	24.96 ± 0.47	23.22	61.35	188	23.37 ± 0.31	23.21	58.81	0
	r_i ∼Uniform[0.005,0.01]							
2-step	0.00 ± 0.00	0.00	0.00	5000	0.00 ± 0.00	0.00	0.05	4202
Threshold-1	0.80 ± 0.07	0.00	20.38	4171	0.66 ± 0.05	0.00	14.03	3884
Threshold-2	0.01 ± 0.00	0.00	1.24	4864	0.04 ± 0.00	0.00	1.90	161
Myopic	0.86 ± 0.07	0.00	22.77	4162	0.67 ± 0.05	0.01	14.43	781
$\alpha r \mu$	0.86 ± 0.07	0.00	22.86	4159	0.67 ± 0.05	0.01	14.43	782
TCF	26.22 ± 0.47	24.53	59.77	0	23.72 ± 0.32	23.56	59.14	0

One of the most important conclusions from Table 3.2 is that the state-dependent heuristics that are developed for the exponential case also perform reasonably well in a non-exponential setting. In particular, the 2-step policy and at least one of the threshold policies perform significantly better than the state-independent policies for all the parameters tested. To be more specific, when the jobs are time-critical, one of the 2-step or the Threshold-1 policies provides the best performance; when the jobs are not very time critical, either the 2-step policy or the Threshold-2 policy is the best heuristic. This is different than the Markovian case, where Threshold-1 is the best policy across all parameter sets (see Tables 3.1 and 3.3).

Table 3.3: Performance of the heuristic policies (in terms of the percentage deviation from the optimal performance) when the service times and lifetimes are exponentially distributed and $m_i = 10$ for $i = 1, \ldots, K$.

	$M = K = 2$				$M = K = 3$			
Heuristic	95% **C.I.**	**Median**	**Maximum**	$\#$ **of times best**	95% **C.I.**	**Median**	**Maximum**	$\#$ **of times best**
	r_i ∼Uniform[2.0,5.0]							
2-step	0.08 ± 0.01	0.00	3.37	4383	0.37 ± 0.02	0.02	5.11	2366
Threshold-1	0.04 ± 0.00	0.00	1.32	4021	0.14 ± 0.01	0.04	2.82	1819
Threshold-2	0.82 ± 0.02	0.77	2.84	161	0.74 ± 0.02	0.53	5.46	25
Myopic	0.36 ± 0.04	0.00	13.09	4473	1.80 ± 0.08	0.00	16.93	2682
$\alpha r \mu$	2.19 ± 0.14	0.00	33.34	3755	8.47 ± 0.22	6.30	36.33	568
TCF	34.77 ± 0.70	31.94	88.36	233	35.16 ± 0.54	33.13	78.91	20
	r_i ∼Uniform[0.5,2.0]							
2-step	0.54 ± 0.04	0.00	8.43	3604	1.60 ± 0.06	0.46	12.68	849
Threshold-1	0.06 ± 0.01	0.00	1.98	4116	0.26 ± 0.01	0.02	3.68	3013
Threshold-2	0.73 ± 0.03	0.46	5.33	213	1.55 ± 0.06	0.52	9.84	19
Myopic	0.87 ± 0.07	0.00	16.95	4066	2.79 ± 0.11	0.70	19.19	2088
$\alpha r \mu$	1.94 ± 0.12	0.00	26.99	3650	5.54 ± 0.17	3.14	26.33	1208
TCF	24.49 ± 0.58	20.28	76.76	633	22.62 ± 0.40	21.32	61.75	153
	r_i ∼Uniform[0.1,0.5]							
2-step	1.54 ± 0.08	0.00	14.70	3049	2.84 ± 0.09	1.74	17.38	278
Threshold-1	0.04 ± 0.00	0.00	1.01	4494	0.11 ± 0.01	0.00	1.87	3900
Threshold-2	1.37 ± 0.07	0.04	13.28	159	2.69 ± 0.08	1.66	15.01	17
Myopic	0.32 ± 0.03	0.00	9.13	4348	0.85 ± 0.04	0.03	8.66	2726
$\alpha r \mu$	0.45 ± 0.04	0.00	11.48	4242	1.11 ± 0.05	0.12	10.47	2482
TCF	12.50 ± 0.37	8.08	55.50	1403	9.27 ± 0.23	7.27	36.75	532
	r_i ∼Uniform[0.01,0.1]							
2-step	1.48 ± 0.06	0.00	11.62	2533	1.90 ± 0.05	1.24	10.71	129
Threshold-1	0.00 ± 0.00	0.00	0.15	4913	0.00 ± 0.00	0.00	0.17	4674
Threshold-2	1.37 ± 0.06	0.00	11.29	2292	1.82 ± 0.05	1.19	10.20	159
Myopic	0.01 ± 0.00	0.00	1.29	4853	0.02 ± 0.00	0.00	0.87	4413
$\alpha r \mu$	0.01 ± 0.00	0.00	1.94	4838	0.02 ± 0.00	0.00	1.01	4368
TCF	2.40 ± 0.10	0.18	25.76	2387	1.32 ± 0.05	0.53	14.63	1303
	r_i ∼Uniform[0.005,0.01]							
2-step	0.07 ± 0.00	0.00	0.93	3808	0.16 ± 0.00	0.12	0.88	79
Threshold-1	0.00 ± 0.00	0.00	0.00	4990	0.00 ± 0.00	0.00	0.01	4897
Threshold-2	0.06 ± 0.00	0.00	0.92	3927	0.15 ± 0.00	0.11	0.80	210
Myopic	0.00 ± 0.00	0.00	0.03	4983	0.00 ± 0.00	0.00	0.02	4831
$\alpha r \mu$	0.00 ± 0.00	0.00	0.03	4982	0.00 ± 0.00	0.00	0.02	4820
TCF	1.06 ± 0.03	0.78	4.26	1195	0.67 ± 0.02	0.50	2.72	624

Among all the index policies considered, myopic policy is the best and the $\alpha r \mu$-rule performs similarly well for small abandonment rates as in the Markovian case. However, when compared with the performances under the Markovian case reported in Table 3.3, the overall performances of the index policies are relatively worse.

One needs to be careful about carrying over every insight from our numerical study to practice directly as the actual problem in emergency response is more complicated than any mathematical model that can be analyzed. For example, without further study, it would not be reasonable to claim any one of the heuristic policies to be superior than the others for practical purposes or that their performances will actually be as close to optimality as the numerical study suggests. Nevertheless, we believe that our numerical study suggests a number of general insights that can be useful for emergency response practitioners. First, there can be significant benefits of taking resource limitations and casualty numbers into account while giving prioritization decisions, especially when patients' life expectancies are short. Second, these state-dependent policies need not be very complex; policies that simply keep track of the total number of patients and prioritize patients accordingly (as in our threshold policy) can perform quite well. Finally, when patients' conditions are not very critical, state-independent policies perform reasonably well and thus can be preferred over state-dependent policies because of their simplicity. However, the choice of the state-independent policy is important as the superiority of the myopic policy across all parameter regions, particularly over the TCF policy, clearly indicates.

CHAPTER 4

Scheduling of impatient customers in a clearing system with a single server and type-dependent service times

In this chapter, we extend the problem in Chapter 3 such that jobs differ not only in their lifetime and reward distributions but also in their service time distributions. The notation and the modeling assumptions of Chapter 3 are still valid in this chapter unless they are redefined. Let S_i be the service time for job $i \in \{1, \dots, N\}$. We assume that $\{Y_i\}_{i=1}^{N}$, $\{Z_i\}_{i=1}^{N}$, and $\{S_i\}_{i=1}^{N}$ are sequences of independent random variables and that these three sequences are independent from each other. One can see from the proof of Proposition 3.1.1 in the Appendix that idling is still suboptimal when the service times are type-dependent. Hence, the decision epochs are time zero and the service completion instants. Our objective is to identify characteristics of policies that maximize $C_\pi(t)$ stochastically, and thereby maximize its expected value.

We briefly outline the contents of this chapter. In Section 4.1, a sample-path argument is used to show that if urgent jobs are also faster to serve and bring higher rewards, then they should always be prioritized in a system with a single server. Without such a condition, other simplifying assumptions are needed to ensure analytical tractability. Therefore, in Sections 4.2 and 4.3, we assume that the service time and lifetime for each job are exponentially distributed random variables, and prove a number of structural results for the optimal policy. Finally, based on these analytical results, we propose some heuristic policies in Section 4.4, and present a numerical study on the performances of these heuristic policies in Section 4.5.

4.1 When more urgent jobs have higher rewards and shorter service times

In this section, we investigate the case in which jobs with shorter lifetimes also have shorter service times but higher rewards. Our objective is to maximize $C_\pi(t)$ stochastically. Throughout this section we do not make any distributional assumptions on service times, lifetimes, or rewards. The following proposition is the main result of this section and it generalizes Proposition 3.2.1 to type-dependent service times but under the condition that there is a single server. The proof for this result is given in the Appendix.

Proposition 4.1.1. *Suppose that $M = 1$ and consider a decision epoch $t_0 \geq 0$ at which jobs i and j are available for service. If $Y_i \leq_{hr} Y_j$, $S_i \leq_{lr} S_j$, and $Z_i \geq_{lr} Z_j$, then a policy $\pi \in \Pi$ that serves job j at time t_0 can be improved (in the sense of stochastically increasing $C_\pi(t)$ for all $t \geq t_0$) by serving job i instead of job j at time t_0.*

Proposition 4.1.1 can be used to partially characterize the optimal policy when there is a single server and at least two jobs that satisfy the "agreeability" conditions on service times, lifetimes, and rewards. More specifically, Proposition 4.1.1 implies that serving a job that has a shorter service time (in the sense of likelihood ratio orders), a shorter lifetime (in the sense of hazard rate orders), and a higher reward (in the sense of likelihood ratio orders) increases the total reward (in the sense of usual stochastic orders). In the special case where all jobs are agreeably ordered, Proposition 4.1.1 gives a complete characterization of the optimal policy as stated in the following corollary.

Corollary 4.1.1. *If $M = 1$, $Y_1 \leq_{hr} Y_2 \leq_{hr} \cdots \leq_{hr} Y_N$, $S_1 \leq_{lr} S_2 \leq_{lr} \cdots \leq_{lr} S_N$, and $Z_1 \geq_{lr} Z_2 \geq_{lr} \cdots \geq_{lr} Z_N$, then a non-idling policy that prioritizes the job with the smallest index at every decision epoch maximizes $C_\pi(t)$ in the sense of usual stochastic orders at every $t \geq 0$.*

Note that Corollary 4.1.1 generalizes Theorem 1 in Argon et al. [2] by relaxing the assumption on deterministic rewards. One may expect that Proposition 4.1.1 and Corollary 4.1.1 may also hold for $M \geq 2$. The following example shows that this is not true in general.

Example 4.1.1. Consider a clearing system with two parallel servers and five jobs at time zero. Suppose that the service times of jobs 1 and 2 are equal to 1 time unit and the service times of jobs 3, 4, and 5 are equal to 2 time units; the lifetime for each job is $5/2$ time units; and the reward for each job is equal to 1. If a policy π follows the policy described in Proposition 4.1.1 and takes jobs 1 and 2 into service at time zero, then at most four jobs could be taken into service by any given time t, i.e., $C_\pi(t) \leq 4$ for all $t \geq 0$. On the other hand, policy γ that assigns jobs 1, 2, and 3 (in the given order) to one server, and jobs 4 and 5 to the other server, will yield $C_\gamma(t) = 5$ for $t \geq 2$.

This example shows that when the service times are not i.i.d. for all jobs, then optimally assigning multiple servers to jobs can be complex and it may involve some not-so-intuitive actions. Thus, in this chapter, we will only focus on the single-server case ($M = 1$).

4.2 When more urgent jobs have lower rewards and longer service times

In the aftermath of a mass-casualty event, patients whose conditions are more urgent are expected to have a longer service time and a lower chance of successfully completing their treatment. Therefore, investigating priority decisions for this case, that is, the case where more urgent jobs bring lower rewards and have longer service times, is crucial. However, characterizing the optimal policy for this case is difficult under general service time and lifetime distributions. Therefore, in this section, we assume that service times and lifetimes are exponentially distributed to obtain partial characterizations of the optimal policy that will lead to insights into policies that perform well. The issues related to this assumption in the context of patient triage are discussed in detail in Chapter 3, and hence will not be repeated here.

We categorize jobs into K types based on their service times, lifetimes, and rewards, where $2 \leq K \leq N$. For $i = 1, \ldots, K$, let $\mu_i > 0$, $r_i > 0$, and $\alpha_i > 0$ be the service rate, abandonment rate, and the expected reward for a type i job, respectively. Similar to Chapter 3, we let Z_i denote the reward of a type i job for $i = 1, \ldots, K$, and we assume that Z_i comes from a distribution such that $\alpha_i \leq \alpha_j$ implies that $Z_i \leq_{lr} Z_j$ for all $i, j \in \{1, \ldots, K\}$.

Next, we let $D_\pi(m_1, \ldots, m_K)$ be the expected total reward accumulated when scheduling (prioritization) policy $\pi \in \Pi$ is applied and m_i jobs from type $i \in \{1, \ldots, K\}$ are initially in the system, where $\sum_{j=1}^{K} m_j = N$. We will use dynamic programming to characterize the solution of the optimization problem stated as

$$\max_{\pi \in \Pi} D_\pi(m_1, \ldots, m_K)$$

for the model with a single server and type-dependent service times. The state of the system is defined with the vector $(\mathbf{q}; Q)$, where $\mathbf{q} := (q_1, \cdots, q_K)$, q_i is the number of type i jobs in queue, and $Q \in \{P_1, \ldots, P_K, R\}$ is the status of the server. Here, $Q = P_i$ indicates that the server is busy processing a job of type $i \in \{1, \ldots, K\}$, and $Q = R$ indicates that the server is idle and ready to begin processing a new job. The decision epochs are time zero and service completion times. At a decision epoch, that is, when Q is equal to R, the possible actions are allocating the server to a job from type $i \in \{1, \ldots, K\}$ such that $q_i > 0$. We next present the dynamic programming equations.

Let $V(\mathbf{q}; Q)$ be the maximum expected reward earned starting from state $(\mathbf{q}; Q)$. Then, using the convention that $V(\mathbf{q}; Q) = 0$ if $q_i = 0$ for all $i \in \{1, \ldots, K\}$ or $\min\{q_1, \ldots, q_K\} < 0$ for all $Q \in \{P_1, \ldots, P_K, R\}$, we have:

$$\begin{aligned} V(\mathbf{q}; R) &= \max_{i=1,\ldots,K} \{\mathbb{I}_{\{q_i>0\}}\alpha_i + V(\mathbf{q} - \mathbf{e}_i; P_i)\}, && (4.2.1) \\ V(\mathbf{q}; P_i) &= \frac{\mu_i V(\mathbf{q}; R) + \sum_{j=1}^{K} q_j r_j V(\mathbf{q} - \mathbf{e}_j; P_i)}{\mu_i + \sum_{j=1}^{K} q_j r_j}, \\ \forall \quad \mathbf{q} &\in \left\{(q_1, \cdots, q_K) : q_i = 0, 1, \ldots, m_i, i = 1, \ldots, K; \sum_{j=1}^{K} q_j \leq N\right\}. && (4.2.2) \end{aligned}$$

We next use this dynamic programming formulation to obtain conditions under which the optimal policy can be characterized for the single-server case with type-dependent service times. Note that Proposition 4.1.1 and Corollary 4.1.1 already provide some sufficient conditions under which the optimal policy is characterized. In particular, Corollary 4.1.1 says that if $\alpha_K \leq \alpha_{K-1} \leq \cdots \leq \alpha_1$, $\mu_K \leq \mu_{K-1} \leq \cdots \leq \mu_1$, and $r_K \leq r_{K-1} \leq \cdots \leq r_1$, then it is optimal to prioritize type 1 jobs. In other words, if the most urgent job has the highest reward and the shortest mean service time, then it is optimal to serve that job regardless of the system state. However, the more interesting and realistic case is when more urgent jobs have smaller rewards and longer mean service times, that is, when $\alpha_K \geq \alpha_{K-1} \geq \cdots \geq \alpha_1$, $\mu_K \geq \mu_{K-1} \geq \cdots \geq \mu_1$, and $r_K \leq r_{K-1} \leq \cdots \leq r_1$. Hence, in the remainder of this section, we will focus on this case.

We next present Proposition 4.2.1, which gives a set of conditions for the monotonicity of the policy in the number of jobs in the queue. To be more specific, it follows from Proposition 4.2.1 that under certain conditions it is optimal to serve a type j job at $(\mathbf{q}; R)$ if it is optimal to serve a type j job in states $(\mathbf{q} - \mathbf{e}_k; R)$ for all $k = 1, \ldots, K$. Proposition 4.2.1 is an extension of Proposition 3 in Argon et al. [2]

to more than two job types and type-dependent rewards. The proof of Proposition 4.2.1 as well as all the other propositions presented in this section are provided in the Appendix.

Proposition 4.2.1. *Consider a job type $j \in \{1, \ldots, K\}$ and a state $(\mathbf{q}; R)$, where $q_j \geq 1$. Suppose that an optimal action in state $(\mathbf{q} - \mathbf{e_k}; R)$ is to serve a type j job for all $k \in \{1, \ldots, K\} \setminus \{j\}$ such that $q_k \geq 1$ and also for $k = j$ if $q_j \geq 2$. If*

$$(\alpha_j r_j - \alpha_i r_i) \sum_{k=1}^{K} q_k r_k + (r_i - \mu_i - r_j + \mu_j) \sum_{k=1}^{K} \alpha_k q_k r_k \geq \alpha_i r_i (\mu_j - r_j) - \alpha_j r_j (\mu_i - r_i), \tag{4.2.3}$$

$$(\mu_j - \mu_i) \left[(r_i - \mu_i) q_j r_j + (r_j - \mu_j)(\mu_i - r_i + \sum_{k=1, k \neq j}^{K} q_k r_k) \right] \geq 0, \tag{4.2.4}$$

$$r_i - \mu_i \geq r_j - \mu_j \text{ (when } K \geq 3), \text{ and} \tag{4.2.5}$$

$$\mu_i = \mu \text{ (when } K \geq 3) \tag{4.2.6}$$

for some $\mu > 0$ and every $i \in \{1, \ldots, K\} \setminus \{j\}$ such that $q_i \geq 1$, then an optimal action in state $(\mathbf{q}; R)$ is to serve a type j job.

Proposition 4.2.1 and Proposition 3.3.1 are similar results but neither one follows from the other as Proposition 4.2.1 considers the problem with a single server but type-dependent service times, whereas Proposition 3.3.1 considers the problem with multiple parallel servers but equal service rates. On the other hand, they are consistent because if we let $\mu_j = \mu$ in Proposition 4.2.1, then the conditions in Proposition 4.2.1 diminish to the conditions for the case with $M = 1$ in Proposition 3.3.1.

We next use Proposition 4.2.1 to determine sufficient conditions for the optimality of simple state-independent policies. Furthermore, later in Section 4.3, we will use Proposition 4.2.1 to partially characterize the structure of optimal policies that are possibly state-dependent.

Proposition 4.2.2. *Suppose that there exists a type $j \in \{1, \ldots, K\}$ such that $r_j \geq \mu_j > \mu_i$, $\alpha_j \geq \alpha_i$, and $\alpha_j r_j \geq \alpha_i r_i$ for all $i \in \{1, \ldots, K\} \setminus \{j\}$, where $\mu_i = \mu$ for some $\mu > 0$ and every $i \in \{1, \ldots, K\} \setminus \{j\}$. Then, the optimal policy gives priority to type j jobs at all decision epochs.*

Similar to Proposition 4.1.1, Proposition 4.2.2 provides conditions under which one type of job should always have priority over the others. In particular, Proposition 4.2.2 implies that if the job with the highest reward also has the fastest service and a sufficiently fast abandonment rate (which can possibly be smaller than the rates for other jobs), then it is optimal to give priority to that job. Furthermore, together with Corollary 3.3.2, the partial characterization in Proposition 4.2.2 immediately yields a complete characterization for the optimal policy, which is an *index policy*, under certain conditions:

Corollary 4.2.1. *If $\alpha_K \geq \alpha_{K-1} \geq \cdots \geq \alpha_1$, $r_K \geq \mu_K > \mu_{K-1} = \cdots = \mu_1$, and $\alpha_K r_K \geq \alpha_{K-1} r_{K-1} \geq \cdots \geq \alpha_1 r_1$, then the optimal policy gives priority to the jobs with the highest index at all decision epochs.*

Recall that an index policy is a set of state-independent decision rules that assigns priorities based only on job types at any given state. The main advantage of index policies over state-dependent policies is the ease in implementation since the priority relation among types of jobs does not change with time and system state. Therefore, Corollaries 4.1.1 and 4.2.1 are important as they provide conditions under which we can safely apply these simple policies. Our next result identifies other potentially good index policies for two special cases.

Proposition 4.2.3. *Suppose that there is an optimal policy among the set of all index policies.*
(i) If $\alpha_i = \alpha$ for some $\alpha > 0$ and for all $i = 1, \ldots, K$, then the optimal policy gives priority to the job with the largest value of $r_i \mu_i$.
(ii) If $r_i = r$ for some $r > 0$ and for all $i = 1, \ldots, K$, then the optimal policy gives priority to the job with the largest value of $\alpha_i(\mu_i + r)$.

Proposition 4.2.3 characterizes the best index policy when it is known that an index policy is optimal in Π. In particular, it tells that $r\mu$-rule and $\alpha(\mu + r)$-rule are the optimal index rules given that an index policy is optimal when $\alpha_i = \alpha$ and $r_i = r$, respectively, for all $i = 1, \ldots, K$. Note that, since $\alpha_i \geq \alpha_j$, $\mu_i \geq \mu_j$, and $r_i \geq r_j$ imply that $\alpha_i r_i \geq \alpha_j r_j$ and $\alpha_i(\mu_i + r) \geq \alpha_j(\mu_j + r)$, Proposition 4.2.3 is consistent with Proposition 4.1.1 and Corollary 4.1.1. Moreover, Proposition 4.2.3 (i) extends Proposition 2 of Argon et al. [2] to multiple types of jobs.

4.3 The case with two types of jobs

We now study the special case where jobs are categorized into two types, i.e., $K = 2$. This simplification helps us obtain more analytical results and get a better understanding of the structure of the optimal policy. These results are later used in the development of effective heuristic policies for the case with $K > 2$.

First of all, we assume that $\mu_2 \geq \mu_1$ without loss of generality. Moreover, as the case with $\mu_2 = \mu_1$ is considered in Chapter 3, we will focus on the case where $\mu_2 > \mu_1$ in this section. We start our discussion by first observing the structure of the optimal policy for a specific example, where $\alpha_2 > \alpha_1$ and $r_2 < r_1$, i.e., type 2 jobs that have shorter mean service times also have higher expected rewards and longer mean lifetimes. This example is selected as it demonstrates the most general structure for the optimal policy that we observed in a wide range of numerical examples.

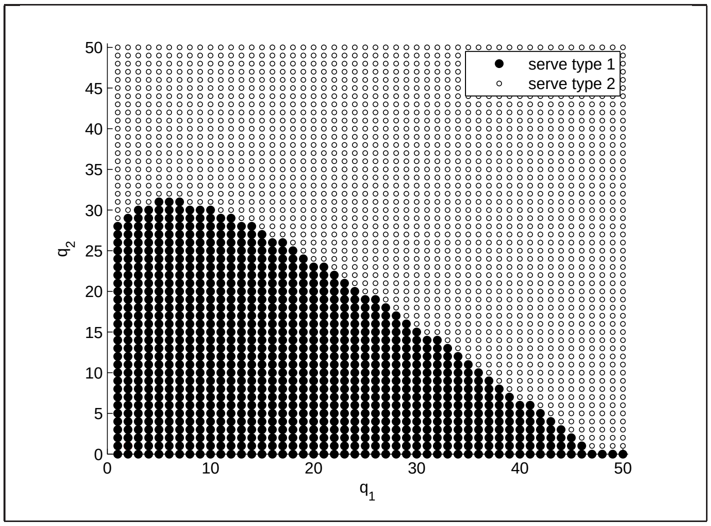


Figure 4.1: The optimal policy for the case where $K = 2$, $\alpha_1 = 1.00$, $\alpha_2 = 1.10$, $\mu_1 = 1.10$, $\mu_2 = 1.22$, $r_1 = 1.11$, and $r_2 = 1.10$.

Figure 4.1 illustrates the optimal allocation of the server at a decision epoch for various values of q_1 and q_2. Similar to Figure 3.1, the optimal policy gives priority to less time-critical type 2 jobs that bring a higher reward when the number of jobs waiting is sufficiently large, with the addition that type 2 jobs have also shorter service times. Considering the emergency response context, one interpretation of this observation is that, when there are many patients in need of treatment, it is best to give priority to faster to treat patients with a higher survival probability, even though those patients are less time-critical. However, if the number of patients is small, giving priority to more urgent patients makes more sense even though they are slower to treat and the chances of saving them are smaller, as there will be enough time to get back to less time-critical patients later.

We next present a result that partially characterizes the structure of the optimal policy that is observed in Figure 4.1 under certain conditions.

Proposition 4.3.1. *Suppose that $\alpha_1 \leq \alpha_2$.*

(i) If $r_1 \leq \mu_1$, then for every $q_1 \geq 1$, the optimal policy has a threshold

$$t(q_1) = \mathbb{I}_{\{r_1 > r_2\}} \left[\frac{\alpha_2 r_2 (r_1 - \mu_1) - \alpha_1 r_1 (r_2 - \mu_2)}{r_2 [r_1(\alpha_2 - \alpha_1) + \alpha_2(\mu_2 - \mu_1)]} - q_1 \frac{r_1 [r_2(\alpha_2 - \alpha_1) + \alpha_1(\mu_2 - \mu_1)]}{r_2 [r_1(\alpha_2 - \alpha_1) + \alpha_2(\mu_2 - \mu_1)]} \right] \quad (4.3.1)$$

such that for all $q_2 \leq t(q_1)$, it is optimal to serve a type 1 job.

(ii) If $r_2 \geq \mu_2$, then there exists a threshold $\tilde{t}(q_1)$, which is greater than or equal to $t(q_1)$ and possibly infinite, such that it is optimal to serve a type 2 job for all $q_2 \geq \tilde{t}(q_1)$ and $q_2 \geq 1$.

Proposition 4.3.1 implies that under certain conditions, when the number of type 2 jobs is lower than a threshold, it is optimal to give priority to type 1 jobs that are slower to serve and that bring a lower reward; and when the number of type 2 jobs is higher than another threshold, it is optimal to give priority to type 2 jobs that are faster to serve and that bring a higher reward. Note that parts (i) and (ii) of Proposition 4.3.1 generalize Propositions 6 and 4 in Argon et al. [2] to type-dependent rewards, respectively.

Our next proposition provides conditions under which an index policy is optimal.

Proposition 4.3.2. *If $\alpha_1 r_1(\mu_1 + r_2) \leq \alpha_2 r_2(\mu_2 + r_1)$, $\alpha_i(\mu_1 - \mu_2) \leq (\alpha_2 - \alpha_1) r_{3-i}$, and $\mu_i \leq r_i$ for $i = 1, 2$, then the optimal policy gives priority to type 2 jobs at all decision epochs.*

Note that Proposition 4.3.2 generalizes Proposition 5 in Argon et al. [2] to type-dependent rewards. Furthermore, by Proposition 4.3.2, we can obtain the following corollary that implies that if type 2 jobs, which have faster service by definition, also bring a higher expected reward, have a higher $\alpha r \mu$ value, and abandon the system at a sufficiently fast rate, then it is optimal to give priority to them at every decision epoch.

Corollary 4.3.1. *If $\alpha_1 \leq \alpha_2$, $\alpha_1 r_1 \mu_1 \leq \alpha_2 r_2 \mu_2$, and $\mu_2 \leq r_2$, then the optimal policy gives priority to type 2 jobs at all decision epochs.*

Corollary 4.3.1 implies that we should give priority to jobs with higher reward and faster service if their abandonment rate is sufficiently high (not necessarily higher than the abandonment rate of the other type). Finally, for the case with $K = 2$, we provide a result that characterizes the optimal index policy when it is known that an index policy is optimal in Π.

Proposition 4.3.3. *If there is an optimal policy among the set of all index policies, then it gives priority to the job with the largest value of* $\alpha_i r_i \mu_i + \alpha_i r_1 r_2$.

Proposition 4.3.3 generalizes Proposition 2 in Argon et al. [2] to type-dependent rewards, which states that the $r\mu$-rule is optimal if there is an optimal index policy. It is interesting to see that the $\alpha r \mu$-rule is not the policy that generalizes Argon et al.'s result. We test the performances of both the index given in Proposition 4.3.3 and the $\alpha r \mu$-rule in Section 4.5.

4.4 Heuristic policies

In Chapter 3, for the multiple-server problem with equal service rates, we developed three heuristic policies and also considered two index policies from the literature as benchmark policies. In this section, we modify the three heuristics developed in Chapter 3 for the single-server problem with type-dependent service rates using our dynamic programming formulation and structural results presented in Section 4.3.

Below, we describe these heuristic policies under the assumption that the service times and lifetimes are exponentially distributed. Note, however, that they can also be applied in more general settings as we explain later in Section 4.5. Also, since the problem essentially reduces to a problem with one less job type when q_i is zero for a job type i, when describing our heuristics we will, without loss of generality, assume that $q_i \geq 1$ for all $i = 1, \ldots, K$.

1. **2-step policy:** At every decision epoch, this heuristic chooses an action that maximizes the expected total rewards over the next two periods. Hence, in order to obtain this policy, we solve the dynamic programming equations (4.2.1) and (4.2.2) assuming that the problem horizon is of two periods length. This gives us the following policy. At a decision epoch, serve type i^* such that

$$i^* = \arg \max_{i \in \{1,\ldots,K\}} \left\{ \alpha_i + \frac{\mu_i \max\left\{\mathbb{I}_{\{q_i \geq 2\}}\alpha_i, \max_{j \in \{1,\ldots,K\}\setminus\{i\}}\{\alpha_j\}\right\}}{\mu_i - r_i + \sum_{j=1}^{K} q_j r_j} \right\}.$$

   (We arbitrarily let i^* be the smallest index that attains the maximum in case of ties.)

2. **Threshold policy:** Threshold heuristic is described by (at most) $K-1$ thresholds $\{\mathcal{T}_1, \ldots, \mathcal{T}_{K-1}\}$, where $\mathcal{T}_1 \leq \mathcal{T}_2 \leq \cdots \leq \mathcal{T}_{K-1}$. For the case where $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_K$, $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_K$,

and $r_1 \geq r_2 \geq \cdots \geq r_K$, this heuristic can be described as follows: At any decision epoch, type j jobs are prioritized if $\mathcal{T}_{j-1} < \sum_{i=1}^{K} q_i \leq \mathcal{T}_j$ for $j = 1, \ldots, K$, where $\mathcal{T}_0 = -\infty$ and $\mathcal{T}_K = \infty$. We use Proposition 4.3.1 in defining the thresholds $\{\mathcal{T}_1, \ldots, \mathcal{T}_{K-1}\}$. More precisely, for any pair of job types i and j, for $i = 1, \ldots, K-1$ and $i = 1, \ldots, j$, we consider the equation $t_{i,j}(q_i) = q_j$, where

$$t_{i,j}(q_i) = \frac{\alpha_j r_j (r_i - \mu_i) - \alpha_i r_i (r_j - \mu_j)}{r_j [r_i(\alpha_j - \alpha_i) + \alpha_j(\mu_j - \mu_i)]} - q_i \frac{r_i [r_j(\alpha_j - \alpha_i) + \alpha_i(\mu_j - \mu_i)]}{r_j [r_i(\alpha_j - \alpha_i) + \alpha_j(\mu_j - \mu_i)]}.$$

First, we let $q_i = 1$ in this equation and solve for q_j (the solution is denoted by q_j^*); and similarly, we let $q_j = 1$ in the same equation and solve for q_i (the solution is denoted by q_i^*). We get

$$q_i^* = \frac{\mu_j(\alpha_i r_i - \alpha_j r_j)}{r_i[r_j(\alpha_j - \alpha_i) + \alpha_i(\mu_j - \mu_i)]} \text{ and } q_j^* = \frac{\mu_i(\alpha_i r_i - \alpha_j r_j)}{r_j[r_i(\alpha_j - \alpha_i) + \alpha_j(\mu_j - \mu_i)]}.$$

Then, we let $T_{i,j} = \max\{q_i^*, q_j^*\}$. Finally, we obtain our thresholds as follows:

$$\mathcal{T}_j = \min\{\mathcal{T}_{j+1}, \max_{i \in \{1,\ldots,j\}} \{T_{i,j+1}\}\}, \text{for } j = 1, \ldots, K-1.$$

3. **Myopic policy:** Proposition 4.3.3 states that, for the case with $K = 2$, the policy which gives priority to the job with the largest value of $\alpha_i r_i \mu_i + \alpha_i r_1 r_2$ index is optimal given that there is an optimal index policy. Myopic policy generalizes the index given in Proposition 4.3.3 to more than two types, that is, it prioritizes type i^* at all decision epochs where

$$i^* = \arg \max_{i \in \{1,\ldots,K\}} \left\{ \alpha_i r_i \mu_i + \alpha_i \prod_{j=1}^{K} r_j \right\}.$$

Note that the $\alpha r \mu$-rule and the myopic policy will behave similarly when the abandonment rate of at least one type is very close to zero.

4.5 Numerical results

In this section, we test the performance of the heuristic policies discussed in Section 4.4 under the assumption that, for type $i \in \{1, \ldots, K\}$ jobs, service times and lifetimes are exponentially distributed with rate $\mu_i > 0$ and $r_i > 0$, respectively. In order to cover as many different scenarios as possible, we used random samples of the system parameters. More specifically, we generated the initial numbers of jobs m_i independently and uniformly over the set $\{1, 2, \ldots, 100\}$ for $i = 1, \ldots, K$. Moreover, for $i = 1, \ldots, K$, we generated the expected rewards α_i and service rates μ_i independently from a uniform distribution with ranges $[0, 1]$ and $[0.5, 2.0]$, respectively. Based on the range of abandonment rates r_i, for $i = 1, \ldots, K$, we conducted five subsets of experiments, and the first subset corresponds to the case where jobs are most time-critical, as r_i's are generated independently from a uniform distribution with range $[2.0, 5.0]$, followed by the other four subsets in decreasing time-criticality order with ranges $[0.5, 2.0]$, $[0.1, 0.5]$, $[0.01, 0.1]$, and $[0.005, 0.001]$. For each subset, we generated 5,000 random scenarios where $\alpha_1 < \cdots < \alpha_K$, $\mu_1 < \cdots < \mu_K$, and $r_1 > \cdots > r_K$. For each scenario, we calculated the expected total reward collected under each of the five heuristic policies and the optimal policy. Then, we computed the percentage deviation of the expected total reward of each heuristic from that of the optimal policy, constructed a 95% confidence interval (C.I.) on the mean of these 5,000 percentage deviations, and calculated the median and the maximum percentage deviation. Finally, we calculated the number of times each heuristic provided the best performance among the five heuristics. The results for $K = 2$ and $K = 3$ are presented in Table 4.1.

From Table 4.1, we observe that, across all parameters, state-dependent policies perform very well and they are significantly better than the index policies when jobs are time-critical, especially in terms of the worst performance. Among the five subsets that we consider, the only subset where a state-dependent policy does not perform best is when r_i ∼Uniform[0.005,0.001], i.e., when jobs are least time-critical. In this case, the $\alpha r \mu$-rule and the myopic policy perform slightly better than the state-dependent policies, but all policies (except for TCF) perform very well, being at most 5.30% worse than the optimal policy.

Comparing the two state-dependent heuristics, namely the 2-step policy and Threshold policy, we see that Threshold policy performs better across all parameters. Comparing the three index policies, the myopic policy performs the best for all parameters, and the difference is significant when jobs are time-critical. When the abandonment rates approach to zero, the $\alpha r \mu$ rule and the myopic policy provide near-optimal performances as expected.

Overall, similar to Chapter 3, by examining Table 4.1, we conclude that using a simple state-dependent policy such as the Threshold policy may improve the system performance significantly. This

Table 4.1: Performance of the heuristic policies (in terms of the percentage deviation from the optimal performance) when the service times and lifetimes are exponentially distributed and $m_i \sim$ Uniform$\{1,\ldots,85\}$ for $i = 1, \ldots, K$.

	$K = 2$				$K = 3$			
Heuristic	95% **C.I.**	**Median**	**Maximum**	$\#$ **of times best**	95% **C.I.**	**Median**	**Maximum**	$\#$ **of times best**
	r_i ∼Uniform[2.0,5.0]							
2-step	0.00 ± 0.00	0.00	0.47	4973	0.00 ± 0.00	0.00	0.49	4920
Threshold	0.00 ± 0.00	0.00	0.47	4995	0.00 ± 0.00	0.00	0.22	4995
Myopic	0.15 ± 0.04	0.00	22.62	4908	0.05 ± 0.02	0.00	17.20	4784
$\alpha r \mu$	1.23 ± 0.13	0.00	36.19	4615	1.24 ± 0.13	0.00	36.88	4219
TCF	49.19 ± 0.62	50.35	99.19	106	64.65 ± 0.47	67.25	96.51	0
	r_i ∼Uniform[0.5,2.0]							
2-step	0.01 ± 0.00	0.00	6.96	4843	0.01 ± 0.00	0.00	2.91	4654
Threshold	0.00 ± 0.00	0.00	4.82	4987	0.01 ± 0.00	0.00	1.31	4960
Myopic	0.85 ± 0.10	0.00	28.63	4641	0.81 ± 0.09	0.00	35.28	4246
$\alpha r \mu$	2.18 ± 0.18	0.00	38.93	4339	2.46 ± 0.18	0.00	39.62	3715
TCF	42.56 ± 0.58	42.87	97.77	114	55.99 ± 0.47	57.49	93.73	4
	r_i ∼Uniform[0.1,0.5]							
2-step	0.08 ± 0.02	0.00	13.55	4588	0.10 ± 0.02	0.00	11.31	4215
Threshold	0.04 ± 0.01	0.00	5.77	4943	0.06 ± 0.01	0.00	6.99	4800
Myopic	1.25 ± 0.11	0.00	29.05	4383	1.97 ± 0.14	0.00	30.29	3681
$\alpha r \mu$	1.76 ± 0.14	0.00	30.65	4244	2.17 ± 0.15	0.00	30.97	3629
TCF	32.52 ± 0.53	31.63	92.50	168	45.23 ± 0.46	46.05	88.12	10
	r_i ∼Uniform[0.01,0.1]							
2-step	0.52 ± 0.05	0.00	20.50	4238	0.53 ± 0.05	0.00	18.64	3579
Threshold	0.40 ± 0.04	0.00	15.76	4485	0.43 ± 0.04	0.00	13.91	4216
Myopic	0.57 ± 0.05	0.00	18.42	4386	1.09 ± 0.07	0.00	21.33	3679
$\alpha r \mu$	0.61 ± 0.06	0.00	22.15	4365	1.09 ± 0.07	0.00	21.33	3679
TCF	13.96 ± 0.35	11.25	74.22	644	22.75 ± 0.36	21.77	69.65	97
	r_i ∼Uniform[0.005,0.01]							
2-step	0.03 ± 0.01	0.00	5.30	4842	0.04 ± 0.01	0.00	3.65	4656
Threshold	0.03 ± 0.01	0.00	5.29	4849	0.04 ± 0.01	0.00	3.65	4663
Myopic	0.02 ± 0.00	0.00	3.76	4897	0.01 ± 0.00	0.00	2.26	4824
$\alpha r \mu$	0.02 ± 0.00	0.00	3.76	4892	0.01 ± 0.00	0.00	2.26	4824
TCF	9.64 ± 0.23	7.67	47.68	258	17.29 ± 0.27	16.36	49.85	15

difference is especially high when jobs are time-critical, that is, when their abandonment rates are high. However, when jobs are not time-critical, since few jobs reach the end of their life while waiting to get service, all five policies that we consider perform similarly (except for TCF). Hence, the selection of the heuristic policy becomes less important in this case, and the index policies, preferably the myopic policy, can also be chosen. The future work includes testing the performance of the heuristics under a non-exponential setting and generalizing our heuristics to multiple servers, hence combining the two problems in this chapter and in Chapter 3.

CHAPTER 5

Extensions

In this chapter, we study two extensions to the base clearing model studied in Chapters 3 and 4. In Section 5.1, we discuss the case where the arrivals of jobs after time zero are allowed. In Section 5.2, we consider the case where a job goes through multiple stages of its lifetime while waiting for service, and at the end of the last stage of its lifetime, it reneges from the system. In each section, we redefine the notation and obtain conditions under which simple state-independent policies are optimal.

5.1 Job arrivals

We consider a single server queueing system where jobs are impatient and classified into $K \geq 2$ types based on their lifetime and service time distributions as well as the rewards that they bring. The lifetime of a type i job, which begins at the time of its arrival to the system, is independent and exponentially distributed with rate $\gamma_i > 0$, for $i = 1, \ldots, K$. The service is performed in a preemptive manner and the service time of a type i job is exponentially distributed with rate μ_i, for $i = 1, \ldots, K$. Jobs from type i arrive to the system according to a Poisson process with rate $\lambda_i > 0$, for $i = 1, \ldots, K$. We let $0 \leq R_i < \infty$ be the expected reward earned when a type i job completes service, for $i = 1, \ldots, K$. We formulate this problem as a MDP and we seek dynamic policies that determine which jobs should be prioritized for service to maximize the long-run average expected reward.

The extended queueing model is inspired by resource allocation problems observed in the aftermath of mass-casualty events such as bioterror attacks, pandemics, or nuclear attacks, in which new patients may arrive to the system as time passes. We relax the assumption that all patients are available at time zero since in such events with longer effects, patients may need medical attention days after the initial event. Hence, together with our analysis of the clearing problem, we can distinguish between mass-casualty events such as bombings that do not involve a significant number of future arrivals of patients after the incident, and mass-casualty events such as a bioterror attack that would involve ongoing arrivals of patients after the initial outburst. The expected arrival interval of all victims in such events is longer than that in mass trauma events like bombings and earthquakes, where the majority of the cases require care within hours of the initial event, but is shorter than that of daily emergency cases which can be modeled as a (possibly non-stationary) stochastic process in steady-state.

We first studied the model described above under a non-preemptive discipline and found quickly that this case is quite difficult to analyze. To demonstrate, consider the following example:

Example 5.1.1. Suppose that jobs 1 and 2 are in the system at time zero, and jobs 3, 4, and 5 arrive at 5, 7, and 9 hours after time zero, respectively. No arrivals are observed after 9 hours. Suppose that the service time of each job is equal to 4 hours; and the reward for serving each job is equal to 1. The lifetime for jobs 1, 3, and 5 is 1 hour and the lifetime for jobs 2 and 4 is 10 hours. Under a policy that takes job 1 into service at time zero (job 1 has a shorter lifetime than job 2), the service of a total of three jobs could be completed. On the other hand, under a policy that assigns job 2 at time zero, idles as there is no one in the system during time interval [4, 5] hours, and serves jobs 3, 5, and 4 (in the given order), the total number of service completions be four, which is optimal.

Example 5.1.1 shows that the policy given in Proposition 4.1.1 is no longer optimal, even for a simple deterministic system with equal service times and equal rewards. It rather involves a more complex structure that is counterintuitive in that it gives priority to a less time-critical job. Therefore, we will consider the case where preemption is allowed, that is, the service of a job can be interrupted.

Since preemption is allowed in our model, one can show that idling is suboptimal by a simple sample path argument (see Down, Koole, and Lewis [26]). Because for any policy that idles, we can construct a non-idling policy that takes all the same actions at the same time as the idling policy, and serves a job waiting in the queue during the idling periods. Then, the non-idling policy will serve all the jobs served under the idling policy or more. Therefore, in the remainder of this section, we only consider non-idling policies.

Remark 5.1.1. We made several attempts to obtain structural results using sample-path arguments. Initially, we considered our problem with no distributional assumptions on lifetimes, service times, and interarrival times. First, we had to restrict our attention to the preemptive service discipline since even simple models under non-preemptive service discipline had very complicated and counter-intuitive optimal policies as illustrated in Example 5.1.1. On the other hand, we also observed that the sample-path analysis of problems under preemptive service discipline can be complicated. Hence, we focused on the Markovian case, which still requires non-trivial arguments due to the addition of arrivals.

We next define our problem more rigourously. Let Π be the set of prioritization policies under consideration. For all $\pi \in \Pi$ and $t \geq 0$, we let $D_\pi(t)$ be the total reward and $\Gamma_\pi(t) = \mathbb{E}[D_\pi(t)]/t$ be the expected average reward up to time t under π. We are interested in solving the following optimization problem:

$$\max_{\pi \in \Pi} \lim_{t \to \infty} \Gamma_\pi(t). \tag{5.1.1}$$

For all $\pi \in \Pi$ and $t \geq 0$, we let $X_\pi(t) = (X_{\pi,1}(t), \ldots, X_{\pi,K}(t))$, where $X_{\pi,i}(t)$ denotes the number of type i jobs in the system at time t under π for $i = 1. \ldots, K$. It is clear that for a fixed $\pi \in \Pi$, $X_\pi(t)$ is a continuous-time Markov chain.

In an attempt to specify the optimal policy for our continuous-time problem, we consider a discrete-time equivalent, applying uniformization in the spirit of Lippman [52]. In order to apply uniformization, we need to be able to identify a finite uniformization constant, we will achieve this by limiting the number of jobs in the system. Hence, for technical reasons, we consider two types of capacity restrictions, namely, a capacity on the number of each type of job and a capacity on the total number of jobs. For the sake of briefness, in this section, we explain our model with a system capacity on the number of each type only and we explain the model with a capacity on the total number of jobs in the Appendix. Hence, we assume that capacity for the number of type i jobs is equal to $C_i < \infty$, for $i = 1, \ldots, K$. In other words, if a type i job arrives to the system when there are C_i type i jobs already in the system, then that job is lost. After uniformization, the times between transitions are exponentially distributed with a constant rate and transitions that do not result in a change of state are allowed. Let ψ denote the uniformization constant, which is given by

$$\psi = \sum_{i=1}^{K} \lambda_i + \max_{i=1,\ldots,K} \mu_i + \sum_{i=1}^{K} C_i \gamma_i.$$

Without loss of generality, we assume that the uniformization constant is equal to one, thus the transition rates (arrival, service, and reneging rates) can be interpreted as probabilities. We let $Y_\pi(n) = (Y_{\pi,1}(n), \ldots, Y_{\pi,K}(n))$, where $Y_{\pi,i}(n)$ denotes the number of type i jobs in the system at period $n \in \{0, 1, \ldots\}$ under π, for $i = 1, \ldots, K$. Then $\{Y_\pi(n)\}$ is a discrete-time Markov chain for each policy π obtained by uniformizing $\{X_\pi(t)\}$ with the finite state space $\mathcal{S}$, which is given by

$$\mathcal{S} = \{\, s = (s_1, \cdots, s_K) : s_i \in \{0, 1, \ldots, C_i\} \text{ for } i = 1, \ldots, K\,\},$$

where s_i is the number of type i jobs in the system.

For the infinite horizon average reward optimality problem given by (5.1.1) after uniformization, the set of all feasible actions consists of serving a type i job, where $i = 1, \ldots, K$. Note that the action space is finite as the number of job types K is finite, and $\mathcal{A}(s)$, the set of all admissible actions when the state of the system is s, is given by

$$\mathcal{A}(s) = \{\, a_i : s_i > 0,\ s = (s_1, \cdots, s_K) \in \mathcal{S}\},$$

where a_i is the action of serving a type i job, for $i = 1, \ldots, K$. As proved in Theorem 9.18 in Puterman [61], there exists a stationary policy for the MDP under consideration, since the state space and the action space are finite. Hence, for the remainder of this section, we assume that the class Π of prioritization policies under consideration consists of all Markovian stationary deterministic policies.

A policy $\pi = \{d_1, d_2, \ldots\}$ is defined as a sequence of decision rules, where a decision rule is a mapping from state space to action space, so that $d_n(s) \in \mathcal{A}(s)$, where $d_n(s)$ is the action to be taken at period $n \in \{0, 1, \ldots\}$ and $s \in \mathcal{S}$. Then, a stationary policy $\pi \in \Pi$ is a sequence of decisions $\pi = \{d, d, \ldots\}$, where $d_n(s) = d(s)$ for all $n \in \{0, 1, \ldots\}$ and $s \in \mathcal{S}$. Moreover, the random reward function is denoted by $r_n(s, a)$, which is the total reward earned at period $n \in \{0, 1, \ldots\}$ starting at state $s \in \mathcal{S}$ and taking action $a \in A(s)$. The reward function is additive in the sense that the reward incurred at period n accumulates over time. Then, the expected long-run average reward per period for policy π is

$$J_\pi(s_0) = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}\left[\sum_{n=0}^{N-1} r(Y_\pi(n), d(Y_\pi(n)))\right],$$

where the system is in state $s_0 \in \mathcal{S}$ at the first decision epoch. Therefore, the optimal average expected reward is given by

$$J(s_0) = \min_{\pi \in \Pi} J_\pi(s_0).$$

Bellman's optimality equation for the above problem takes the form:

$$g + v(s) = \min_{a_i \in \mathcal{A}(s)} \{v(s, a_i)\},$$

where g is the optimal long-run average reward per period, and $v(s)$ can be interpreted as a relative or differential reward for each state $s \in \mathcal{S}$. Let e_i be a row vector of K components consisting of zeros except for a one in the ith position and $\mathbf{1}_{\{s_i < C_i\}}$ be the indicator function of the set $\{s_i < C_i\}$, for $i = 1, \ldots, K$. Then, Bellman's equation is given by:

$$\begin{aligned} g + v(s) = & \sum_{j=1}^{K} \mathbf{1}_{\{s_j < C_j\}} \lambda_j v(s + e_j) + \sum_{j=1}^{K} s_j \gamma_j v(s - e_j) \\ & + \left[1 - \sum_{j=1}^{K} \mathbf{1}_{\{s_j < C_j\}} \lambda_j - \sum_{j=1}^{K} s_j \gamma_j \right] v(s) + \max_{a_i \in \mathcal{A}(s)} \{M(s, a_i)\}, \end{aligned}$$

where

$$M(s, a_i) = R_i \mu_i + (\gamma_i - \mu_i) \left[v(s) - v(s - e_i) \right], \quad \text{for } i = 1, \ldots, K.$$

Using these equations, we obtain Propositions 5.1.1 and 5.1.2, which are proved in the Appendix.

Proposition 5.1.1. *Suppose that there is a capacity restriction on the number of each type of jobs. Then, in the class of Markovian stationary deterministic policies* Π, *the policy that serves type* i *jobs, where* $i = 1, \ldots, K$, *is the optimal solution to Problem (5.1.1) if* $\gamma_i \geq \mu_i$, *and* $R_i \mu_i \geq R_j \mu_j$ *and* $\mu_j \geq \gamma_j$ *for all* $j = 1, \ldots, K$ *and* $j \neq i$.

We can obtain the following insights from Proposition 5.1.1:

1. **Equal rewards:** For $R_i = R_j$, $\forall i, j \in \{1, \ldots, K\}$, the conditions given in Proposition 5.1.1 diminish to $\gamma_i \geq \mu_i \geq \mu_j \geq \gamma_j$. Note that this result is consistent with Proposition 4.1.1 as the job that has a shorter service time and lifetime is given priority under the optimal policy.

2. **Equal abandonment rates:** When we let $\gamma_j := \gamma$ for all $j \in \{1, \ldots, K\}$, the conditions given

in Proposition 5.1.1 diminish to the condition $\mu_j \geq \gamma \geq \mu_i \geq \frac{R_j}{R_i}\mu_j$, which means that it is optimal to serve type i jobs at all decision epochs if $R_i \geq R_j$, $\mu_i \in [\frac{R_j}{R_i}\mu_j, \mu_j]$, and $\gamma \in [\mu_i, \mu_j]$. This implies that a job with a sufficiently large reward and a slow service (compared to abandonments) should be given priority over a job with a smaller reward and a faster service (compared to abandonments).

3. **Equal service rates:** Down, Koole, and Lewis [26] consider a reward model for two types of jobs with equal service rates, and Proposition 5.1.1 is consistent with their main result, which states that if $R_i \geq R_j$ and $\gamma_i \geq \gamma_j$, it is optimal to serve type i jobs, for $i, j \in \{1, 2\}$. When we let $\mu_i = \mu_j$, the conditions given in Proposition 5.1.1 become $R_i \geq R_j$ and $\gamma_i \geq \mu \geq \gamma_j$. Hence, both results are consistent with each other but neither one implies the other. Moreover, our result is also consistent with Proposition 3.2.1 as the job that has a higher reward and a shorter lifetime is given priority under the optimal policy.

Proposition 5.1.2. *Suppose that there is a capacity restriction on the total number of jobs and* $K = 2$*. In the class of Markovian stationary deterministic policies* Π*, the policy that serves type* i *jobs, where* $i = 1, 2$*, is the optimal solution to Problem (5.1.1) if* $\mu_i \geq \gamma_i$*,* $R_1\mu_1 = R_2\mu_2$*, and* $\gamma_i - \gamma_j \geq \mu_i - \mu_j \geq 0$ *for* $j = 1, 2$ *and* $j \neq i$*.*

Note that, for $\mu_2 \geq \mu_1$, the conditions of Proposition 5.1.2 imply that if $R_1 \geq R_2$ (so that $R_1\mu_1 = R_2\mu_2$), $\gamma_2 \geq \gamma_1$, and $0 \geq \gamma_2 - \mu_2 \geq \gamma_1 - \mu_1$, then type 2 jobs should be prioritized. For the patient triage problem, this corresponds to the case where urgent patients are faster to serve but have lower chances of survival. Then, those patients are given priority if they abandon the system at a faster rate than their service and this difference is larger than the other type. One important remark is that no condition involving the arrivals is needed for Propositions 5.1.1 and 5.1.2.

5.2 Multiple stages of lifetime

In this part of our research, we assume that the lifetime of a job consists of multiple stages that may affect the service time and reward distributions. A job that goes through all stages of its lifetime while still in queue reneges from the system before receiving any service. Furthermore, we assume that jobs are monitored so that their classification is continuously updated according to their current condition. In the context of patient triage, this corresponds to the patients' going through various stages of a disease or a condition with unique care requirements. Patients pass to the next stage of their lifetime as time passes, and that changes the chance of survival and time required for their treatment. It is important to take into account the changes in the patients' condition with time. For example, after the Oklahoma City Bombing in 1995, the critical patients were given higher priority to be dispatched to hospitals, but as time passed the condition of the so called non-critical patients started to deteriorate, and unfortunately these patients did not receive treatment for many hours as they were labeled "non-critical" during the initial triage.

The lifetime of a job has $K \geq 2$ stages, each of which is independent and exponentially distributed with rate $\gamma_i > 0$, for $i = 1, \ldots, K$. Jobs which are in the ith stage of their lifetime arrive to the system according to a Poisson process with rate $\lambda_i > 0$, for $i = 1, \ldots, K$. If a job is in the ith stage of its lifetime at the start of its service, then the service time is exponentially distributed with rate μ_i, for $i = 1, \ldots, K$. Moreover, for $i = 1, \ldots, K$, we let $0 \leq R_i < \infty$ be the expected reward earned when a job in the ith stage of its lifetime completes its service. Similar to Section 5.1, we assume that there is a single server, the service is performed in a preemptive manner, and and jobs do not renege while they are in service. Finally, we assume that the system capacity for the number of jobs at the ith stage of their lifetime is equal to $C_i < \infty$, that is, if the capacity for the number of jobs at a particular stage is reached, jobs arriving to that stage exogenously and from the previous stage are lost. We seek dynamic policies that determine which jobs are prioritized for service with the objective of maximizing the long-run average reward.

Let $\widetilde{\Pi}$ be the set of prioritization policies under consideration. For all $\pi \in \widetilde{\Pi}$ and $t \geq 0$, we let $D_\pi(t)$ denote the total reward and $\Gamma_\pi(t) = \mathbb{E}[D_\pi(t)]/t$ be the expected average reward up to time t under policy π. We are interested in solving the following optimization problem:

$$\max_{\pi \in \widetilde{\Pi}} \lim_{t \to \infty} \Gamma_\pi(t). \tag{5.2.1}$$

For all $\pi \in \widetilde{\Pi}$ and $t \geq 0$, we let $\tilde{X}_\pi(t) = (\tilde{X}_{\pi,1}(t), \ldots, \tilde{X}_{\pi,K}(t))$, where $\tilde{X}_{\pi,i}(t)$ denotes the number of jobs in the system at the ith stage of its lifetime at time t under policy π for $i = 1, \ldots, K$. It is clear that for a fixed $\pi \in \widetilde{\Pi}$, $\{\tilde{X}_\pi(t)\}$ is a continuous-time Markov chain with the state space $\mathcal{S}$, which is given by

$$\mathcal{S} = \{\, s = (s_1, \cdots, s_K) : s_i \in \{0, 1, \ldots, C_i\} \text{ for } i = 1, \ldots, K \} \,.$$

As in Section 5.1, we uniformize this chain with a uniformization constant

$$\psi = \sum_{i=1}^{K} \lambda_i + \max_{i=1,\ldots,K} \mu_i + \sum_{i=1}^{K} C_i \gamma_i < \infty.$$

Without loss of generality, we assume that the uniformization constant is equal to one. We let $\tilde{Y}_\pi(n) = (\tilde{Y}_{\pi,1}(n), \ldots, \tilde{Y}_{\pi,K}(n))$, where $\tilde{Y}_{\pi,i}(n)$ denotes the number of jobs in the system at the ith stage of their lifetimes at period n under policy π, for $i = 1, \ldots, K$. Then $\{\tilde{Y}_\pi(n)\}$ is a discrete-time Markov chain for each policy π obtained by uniformizing $\{\tilde{X}_\pi(t)\}$.

For the infinite horizon average reward optimality problem given by (5.2.1) after uniformization, the set of all feasible actions consists of serving a job at the ith stage of its lifetime, where $i = 1, \ldots, K$. Note that the action space is finite, and $\mathcal{A}(s)$, the set of all admissible actions when the state of the system is s, is given by

$$\mathcal{A}(s) = \{\, a_i : s_i > 0,\ s = (s_1, \cdots, s_K) \in \mathcal{S}\},$$

where a_i is the action of serving a job at the ith stage, for $i = 1, \ldots, K$. Again by Theorem 9.18 in Puterman [61], there exists a stationary policy for the MDP under consideration. Hence, for the remainder of this section, we assume that the class $\widetilde{\Pi}$ of prioritization policies under consideration consists of all Markovian stationary deterministic policies.

Bellman's optimality equation for the average reward problem takes the form:

$$g + v(s) = \min_{a_i \in \mathcal{A}(s)} \{v(s, a_i)\},$$

where g is the optimal average reward per period, and $v(s)$ can be interpreted as a relative or differential reward for each state $s \in \mathcal{S}$. Again using the notation that e_i is a row vector of K components consisting of zeros except for a one in the ith position and $\mathbf{1}_{\{s_i < C_i\}}$ is the indicator function of the set $\{s_i < C_i\}$, for $i = 1, \ldots, K$, Bellman's equations are given by:

$$\begin{aligned} g + v(s) &= \sum_{j=1}^{K} \mathbf{1}_{\{s_i < C_i\}} \lambda_j v(s + e_j) + \sum_{j=1}^{K-1} s_j \gamma_j v(s - e_j + e_{j+1}) + s_K \gamma_K v(s - e_K) \\ &+ \left[1 - \sum_{j=1}^{K} \mathbf{1}_{\{s_i < C_i\}} \lambda_j - \sum_{j=1}^{K} s_j \gamma_j \right] v(s) + \max_{a_i \in \mathcal{A}(s)} \{M(s, a_i)\}, \end{aligned}$$

where

$$M(s, a_i) = \begin{cases} R_i \mu_i + \gamma_i \left[v(s) - v(s - e_i + e_{i+1}) \right] - \mu_i \left[v(s) - v(s - e_i) \right] & \text{for} \quad i = 1, \ldots, K-1; \\ R_K \mu_K + (\gamma_K - \mu_K) \left[v(s) - v(s - e_K) \right] & \text{for} \quad i = K. \end{cases}$$

Using these equations, we obtain Proposition 5.2.1, which is proved in the Appendix.

Proposition 5.2.1. *In the class of Markovian stationary deterministic policies $\widetilde{\Pi}$, the policy that serves jobs at stage K is the optimal solution to Problem (5.2.1) if $\gamma_K \geq \mu_K$, and $R_K \mu_K \geq R_j \mu_j$ and $\mu_j \geq \gamma_j$ for all $j = 1, \ldots, K-1$.*

The insights obtained from Proposition 5.1.1 are still valid for Proposition 5.2.1 as the required conditions are similar for both results except that Proposition 5.2.1 provide conditions for the optimality of jobs at the last stage of their lifetime whereas Proposition 5.1.1 provide conditions for the optimality of all types of jobs. We need this restriction as this problem with jobs going through stages of their lifetimes is harder to analyze, as the lifetime of a job is not an exponential random variable anymore, instead it is the sum of exponential random variables. Hence, a better approach might be to test some simple policies (such as the heuristic policies that we considered for the clearing problem) using simulation.

CHAPTER 6

Conclusions

In service systems where customers may leave the system without receiving service if their wait exceeds their tolerance, dynamically allocating the limited resources to enhance performance can be a complicated problem. In this thesis we model such systems as queueing systems with multiple classes of impatient customers with the objective of finding effective dynamic scheduling policies that maximize the rewards collected.

Inspired by the debates about response efforts to recent mass-casualty events such as Hurricane Katrina in 2005, our main motivation is a resource allocation problem that may arise in the aftermath of a mass-casualty event. While assigning priorities to injured patients for limited resources, the common practice only uses the time-criticality information of patients. Researchers in the medical community have recognized the potential benefits of also considering the resource limitations in giving prioritization decisions. In this dissertation, we mathematically support the benefits of taking into account the availability of resources, the number of patients, and the type of their injuries in order to optimally allocate limited resources. Although we do not expect precise answers from our mathematical analysis, insights that we obtain from our stylized models can serve as building blocks for policies that can be used in practice.

In our mathematical analysis, the base model is a clearing system where a finite number of jobs are available at the time of the incident (as it would be the case in a mass-trauma event such as a plane crash or bombing in an open space). For the clearing problem, we first consider the multi-server case under the assumption that service times are identically distributed. We later relax this assumption but then restrict our attention to the single server case. For both cases, we used sample-path arguments and dynamic programming to obtain characterizations of the best policies that maximize the expected total reward. In particular, we first identify several conditions under which the system-state information, i.e., the number of available resources and patient counts, can be ignored when determining priorities. For example, when all service times are identically distributed, we showed that if a job with the highest reward (in the sense of likelihood ratio orders) also has the shortest lifetime (in the sense of hazard rate orders), then that job should be prioritized irrespective of the number of other jobs. Second, we partially characterize the optimal policy in cases where the optimal decisions could depend on the system-state. For instance, for the single-server problem, we provide conditions under which giving priority to the type that is faster to serve is optimal if the number of jobs from the type that is slower to serve is less than a threshold value. Third, we demonstrate that one can develop "good" prioritization policies and rules of thumb that only consider the total number of patients as opposed to considering numbers from each type of patient. In particular, with our numerical analysis for the multi-server and single-server problems, we show that a threshold-type policy, which gives priority to time-critical patients if the total number of patients is below a threshold and to less urgent patients otherwise, can perform quite well. We also provide some possible directions for how this threshold can be set. Furthermore, by extending our model to the case with arrivals, we distinguish between mass-casualty events such as bombings that do not involve a significant number of future arrivals after the incident, and mass-casualty events such as a bioterror attack using anthrax or smallpox, that would involve ongoing arrivals of patients after the initial outburst. Moreover, the second extension of our clearing model, in which the criticality levels of patients change with time, corresponds to the case where patients go through multiple stages with unique care requirements which also affects the chance of their survival and the time required for their treatment.

We believe this dissertation provides a common platform of knowledge from which emergency responders (physicians as well as managers) and operations researchers together can build a sound emergency response plan. Moreover, building such a plan requires extensive testing using realistic simulation models, hence, one important future research direction is the development of a simulation test-bed for priority decisions in emergency response, which to the best of our knowledge does not exist at the moment. A realistic simulation model would be of immense practical value to emergency responders around the world in saving more lives in moments of crisis. Part of such a project would also require data gathering on lifetimes for various injury types, as this data would be critical to ensuring both realistic simulation scenarios and to developing effective life-saving policies.

Appendix

In this Appendix, we provide the proofs of our results in the order presented in the thesis.

Proof of Proposition 3.1.1: Assume that a server (say server i) under policy π idles τ time units starting at t_0, while there is at least one job waiting for service. Let γ be another policy and we couple all lifetimes, service times and rewards of all the jobs under both γ and π. All servers under γ follow policy π except that starting at t_0, server i under γ takes the actions that it takes under π starting at $t_0 + \tau$. This is possible since the service completion times of jobs that are taken into service by server i after t_0 under γ are τ units of time earlier than those under π after t_0. Let τ' be the time that server i stops serving jobs under π. If there are any jobs available at time $\tau' - \tau$ under γ, server i serves them in any order until the system is cleared. Thus, we have shown that $C_\gamma(t) - C_\pi(t) \geq 0$ for all $t \geq 0$. $\square$

The following lemma is needed to prove Proposition 3.2.1.

Lemma A.0.1. *(Righter 1994, Lemma 13.D.1; among others) Let X and Y be two independent random variables. Then, $X \leq_{lr} Y$ if and only if $(X|\min\{X,Y\} = \underline{m}, \max\{X,Y\} = \overline{m}) \leq_{st} (Y|\min\{X,Y\} = \underline{m}, \max\{X,Y\} = \overline{m})$ for all $\underline{m} \leq \overline{m}$.*

Lemma A.0.1 can equivalently be stated as follows: Given $\underline{m} = \min\{X,Y\}$ and $\overline{m} = \max\{X,Y\}$, we have that $X \leq_{lr} Y$ if and only if $\Pr\{X = \underline{m}|\underline{m},\overline{m}\} = \Pr\{Y = \overline{m}|\underline{m},\overline{m}\} \geq \Pr\{X = \overline{m}|\underline{m},\overline{m}\} = \Pr\{Y = \underline{m}|\underline{m},\overline{m}\}$.

Proof of Proposition 3.2.1: We will use a coupling argument to prove this result. Let $\widetilde{M}$ be the total number of servers available at time t_0, where $1 \leq \widetilde{M} \leq M$. If $\widetilde{M}$ is greater than or equal to the number of jobs seeking service at time t_0, then all jobs should be taken into service since idling is suboptimal. Otherwise, let $\mathcal{S}_\rho$ be the set of jobs taken into service at time t_0 under policy ρ. Suppose policy π takes job j into service at t_0 while job i is in the queue, i.e., $j \in \mathcal{S}_\pi$ and $i \notin \mathcal{S}_\pi$. We will construct a policy γ which follows policy π between time zero and t_0, but serves job i instead of j at t_0 (i.e., $\mathcal{S}_\gamma = \mathcal{S}_\pi \setminus \{j\} \cup \{i\}$), and for which $C_\pi(t) \leq C_\gamma(t)$ for all $t \geq 0$ along any given sample path.

Let Y_l^ρ denote the remaining lifetime of job l at t_0 under policy ρ, where $l \in \{i,j\}$ and $\rho \in \{\pi,\gamma\}$. Note that by the stochastic ordering relation among the remaining lifetimes of jobs, we can couple the random variables so that $Y_i^\pi = y_i \leq y_j = Y_j^\gamma$. Because policy $\pi(\gamma)$ serves job $j(i)$ at t_0, and the job that is in service will not abandon, we do not need Y_j^π or Y_i^γ. Let $Y_l^\gamma = Y_l^\pi$ for all $l \neq i, j$. Let also S_l^ρ denote the service time of job l under policy $\rho \in \{\pi, \gamma\}$, and let $S_l^\gamma = S_l^\pi$ for all $l \neq i, j$. We can couple (S_i^π, S_j^π) with (S_i^γ, S_j^γ) such that $S_j^\pi = S_i^\gamma := a$ and $S_i^\pi = S_j^\gamma := b$. Finally, let Z_l^ρ denote the reward of taking job l into service under policy $\rho \in \{\pi, \gamma\}$, and let $Z_l^\gamma = Z_l^\pi$ for all $l \neq i, j$. Then, we can couple (Z_i^π, Z_j^π) with (Z_i^γ, Z_j^γ) so that $\min\{Z_i^\pi, Z_j^\pi\} = \min\{Z_i^\gamma, Z_j^\gamma\} \leq \max\{Z_i^\pi, Z_j^\pi\} = \max\{Z_i^\gamma, Z_j^\gamma\}$ and either $Z_j^\pi = Z_i^\gamma$ and $Z_i^\pi = Z_j^\gamma$ or $Z_j^\pi = Z_j^\gamma \leq Z_i^\pi = Z_i^\gamma$. Such a coupling is possible from Lemma A.0.1 and the condition that $Z_j \leq_{lr} Z_i$. Let τ be the time π takes job i into service ($\tau = \infty$ if job i is not taken into service). The following cases exhaust all possibilities:

Case I: We first consider the case where $\tau < \infty$. γ follows π at all decision epochs after t_0 except that it replaces job j with job i at τ. This is possible because $y_i \leq y_j$ and all decision epochs after τ under π and γ take place at the same time with the same set of jobs available for both policies except for jobs i and j. Hence, we have $C_\gamma(t) = C_\pi(t)$ for all $t < t_0$, $C_\gamma(t) - C_\pi(t) = Z_i^\gamma - Z_j^\pi \geq 0$ for all $t_0 \leq t < \tau$, and $C_\gamma(t) - C_\pi(t) = Z_i^\gamma + Z_j^\gamma - Z_i^\pi - Z_j^\pi = 0$ for all $t \geq \tau$.

Case II: Now suppose that $\tau = \infty$. γ follows π after t_0 except that it serves job j last (let the service start time be τ'), if it is still available after all other jobs are cleared. Then, we have $C_\gamma(t) = C_\pi(t)$ for all $t < t_0$, $C_\gamma(t) - C_\pi(t) = Z_i^\gamma - Z_j^\pi \geq 0$ for all $t_0 \leq t < \tau'$, and if $\tau' < \infty$, $C_\gamma(t) - C_\pi(t) = Z_i^\gamma - Z_j^\pi + Z_j^\gamma \geq 0$ for all $t \geq \tau'$. □

Proof of Proposition 3.3.1: We will show that $\alpha_j + V(\mathbf{q} - \mathbf{e}_j; M) \geq \mathbb{I}_{\{q_i \geq 1\}} \alpha_i + V(\mathbf{q} - \mathbf{e}_i; M)$ for all $i = 1, \ldots, K$ under the given conditions. For $i \in \{1, \ldots, K\}$ such that $q_i = 0$, this holds trivially. Hence, we only consider the types of jobs for which $q_i \geq 1$. For $i \in \{1, \ldots, K\} \setminus \{j\}$ such that $q_i \geq 1$, we have

$$\begin{aligned}
&V(\mathbf{q} - \mathbf{e}_j; M) \\
&= \frac{M\mu V(\mathbf{q} - \mathbf{e}_j; M-1) + (q_j - 1) r_j V(\mathbf{q} - 2\mathbf{e}_j; M) + \sum_{k=1, k \neq j}^K q_k r_k V(\mathbf{q} - \mathbf{e_k} - \mathbf{e}_j; M)}{M\mu - r_j + \sum_{k=1}^K q_k r_k} \\
&\geq \frac{M\mu \alpha_i + (\alpha_i - \alpha_j)((q_j - 1) r_j + \sum_{k=1, k \neq i,j}^K q_k r_k)}{M\mu - r_j + \sum_{k=1}^K q_k r_k} \\
&+ \frac{(M\mu + q_i r_i + (q_j - 1) r_j) V(\mathbf{q} - \mathbf{e}_i - \mathbf{e}_j; M) + \sum_{k=1, k \neq i,j}^K q_k r_k V(\mathbf{q} - \mathbf{e}_k - \mathbf{e_i}; M)}{M\mu - r_j + \sum_{k=1}^K q_k r_k}, \quad \text{(A.0.1)}
\end{aligned}$$

where the inequality follows, because, for the first term, $V(\mathbf{q} - \mathbf{e}_j; M-1) \geq \alpha_i + V(\mathbf{q} - \mathbf{e}_i - \mathbf{e}_j; M)$, for the second term, either $q_j = 1$, so the inequality is trivial, or $V(\mathbf{q} - 2\mathbf{e}_j; M) \geq \alpha_i - \alpha_j + V(\mathbf{q} - \mathbf{e}_i - \mathbf{e}_j; M)$, and for the last term, either $q_k = 0$, so the inequality is trivial, or $V(\mathbf{q} - \mathbf{e_k} - \mathbf{e}_j; M) \geq \alpha_i - \alpha_j + V(\mathbf{q} - \mathbf{e_k} - \mathbf{e}_i; M)$, for $k \in \{1, \ldots, K\} \setminus \{i, j\}$. Similarly,

$$\begin{aligned}
&V(\mathbf{q} - \mathbf{e}_i; M) \\
&= \frac{M\mu V(\mathbf{q} - \mathbf{e}_i; M-1) + (q_i - 1) r_i V(\mathbf{q} - 2\mathbf{e}_i; M) + \sum_{k=1, k\neq i}^{K} q_k r_k V(\mathbf{q} - \mathbf{e}_k - \mathbf{e_i}; M)}{M\mu - r_i + \sum_{k=1}^{K} q_k r_k} \\
&\leq \frac{M\mu\alpha_j + (\alpha_j - \alpha_i)(q_i - 1) r_i}{M\mu - r_i + \sum_{k=1}^{K} q_k r_k} + \frac{M\mu + (q_i - 1) r_i + q_j r_j}{M\mu - r_i + \sum_{k=1}^{K} q_k r_k} V(\mathbf{q} - \mathbf{e}_i - \mathbf{e}_j; M) \\
&+ \frac{\sum_{k=1, k\neq i,j}^{K} q_k r_k V(\mathbf{q} - \mathbf{e}_k - \mathbf{e_i}; M)}{M\mu - r_i + \sum_{k=1}^{K} q_k r_k}, \qquad \text{(A.0.2)}
\end{aligned}$$

where the inequality follows, because, for the first term, $V(\mathbf{q} - \mathbf{e}_i; M-1) = \alpha_j + V(\mathbf{q} - \mathbf{e}_i - \mathbf{e}_j; M)$, and for the second term, either $q_i = 1$, so the inequality is trivial, or $V(\mathbf{q} - 2\mathbf{e}_i; M) \leq \alpha_j - \alpha_i + V(\mathbf{q} - \mathbf{e}_i - \mathbf{e}_j; M)$.

Now, from (A.0.1) and (A.0.2), we get

$$\begin{aligned}
&\alpha_j + V(\mathbf{q} - \mathbf{e}_j; M) - \alpha_i - V(\mathbf{q} - \mathbf{e}_i; M) \\
&\geq \alpha_j - \alpha_i + \frac{M\mu\alpha_i + (\alpha_i - \alpha_j)((q_j - 1) r_j + \sum_{k=1, k\neq i,j}^{K} q_k r_k)}{M\mu - r_j + \sum_{k=1}^{K} q_k r_k} - \frac{M\mu\alpha_j + (\alpha_j - \alpha_i)(q_i - 1) r_i}{M\mu - r_i + \sum_{k=1}^{K} q_k r_k}
\end{aligned}$$

$$\begin{aligned}
&+ \left(\frac{M\mu + q_i r_i + (q_j - 1) r_j}{M\mu - r_j + \sum_{k=1}^{K} q_k r_k} - \frac{M\mu + (q_i - 1) r_i + q_j r_j}{M\mu - r_i + \sum_{k=1}^{K} q_k r_k} \right) V(\mathbf{q} - \mathbf{e}_i - \mathbf{e}_j; M) \\
&+ \left(\frac{1}{M\mu - r_j + \sum_{k=1}^{K} q_k r_k} - \frac{1}{M\mu - r_i + \sum_{k=1}^{K} q_k r_k} \right) \sum_{k=1, k\neq i,j}^{K} q_k r_k V(\mathbf{q} - \mathbf{e}_k - \mathbf{e_i}; M) \\
&= \frac{M\mu(\alpha_j r_j - \alpha_i r_i) + (\alpha_j - \alpha_i)((q_i + q_j - 1) r_i r_j + r_i \sum_{k=1, k\neq i,j}^{K} q_k r_k)}{(M\mu - r_j + \sum_{k=1}^{K} q_k r_k)(M\mu - r_i + \sum_{k=1}^{K} q_k r_k)} \\
&+ \frac{(r_i - r_j) \sum_{k=1, k\neq i,j}^{K} q_k r_k \Big(V(\mathbf{q} - \mathbf{e}_i - \mathbf{e}_j; M) - V(\mathbf{q} - \mathbf{e}_k - \mathbf{e_i}; M) \Big)}{(M\mu - r_j + \sum_{k=1}^{K} q_k r_k)(M\mu - r_i + \sum_{k=1}^{K} q_k r_k)} \\
&= \frac{(\alpha_j r_j - \alpha_i r_i) \sum_{k=1}^{K} q_k r_k + (r_i - r_j) \sum_{k=1}^{K} \alpha_k q_k r_k - \alpha_i r_i (M\mu - r_j) + \alpha_j r_j (M\mu - r_i)}{(M\mu - r_j + \sum_{k=1}^{K} q_k r_k)(M\mu - r_i + \sum_{k=1}^{K} q_k r_k)} \\
&+ \frac{(r_i - r_j) \sum_{k=1, k\neq i,j}^{K} q_k r_k \Big(\alpha_j + V(\mathbf{q} - \mathbf{e}_i - \mathbf{e}_j; M) - \alpha_k - V(\mathbf{q} - \mathbf{e}_k - \mathbf{e_i}; M) \Big)}{(M\mu - r_j + \sum_{k=1}^{K} q_k r_k)(M\mu - r_i + \sum_{k=1}^{K} q_k r_k)} \geq 0,
\end{aligned}$$

where the last inequality holds because, for the first term, Condition (3.3.4) holds and for the second term, either $K = 2$ or $q_k = 0$ so that the inequality is trivial, or Condition (3.3.5) holds and $\alpha_j + V(\mathbf{q} - \mathbf{e}_i - \mathbf{e}_j; M) \geq \alpha_k + V(\mathbf{q} - \mathbf{e}_k - \mathbf{e_i}; M)$ for all $k \in \{1, \ldots, K\} \setminus \{i, j\}$. □

Proof of Proposition 3.3.2: We first use Proposition 3.3.1 to prove the result for decision epochs at which a service completion takes place and $q_j \geq 1$. For $i \in \{1, \ldots, K\} \setminus \{j\}$, we rewrite Condition (3.3.4) as

$$(\alpha_j r_j - \alpha_i r_i)\left(M\mu + (q_j - 1)r_j + \sum_{k=1,k\neq j}^{K} q_k r_k\right) + (r_i - r_j)\left(\alpha_j(q_j - 1)r_j + \sum_{k=1,k\neq j}^{K} \alpha_k q_k r_k\right) \geq 0. \tag{A.0.3}$$

Since $r_j \leq r_i$ and $\alpha_j r_j \geq \alpha_i r_i$ for all $i = 1, \ldots, K$, and $q_j \geq 1$, Condition (A.0.3) is satisfied for all $i \in \{1, \ldots, K\} \setminus \{j\}$. Hence, Conditions (3.3.4) and (3.3.5) are satisfied for all $i \in \{1, \ldots, K\} \setminus \{j\}$ such that $q_i \geq 1$.

We will now apply induction on $\sum_{i=1}^{K} q_i$. First, consider the case where $\sum_{i=1}^{K} q_i = 1$ such that $q_j \geq 1$, i.e., $\mathbf{q} = \mathbf{e}_j$. In this case, the result holds trivially. Now, suppose that the result is true for all feasible $\mathbf{q}$ such that $\sum_{i=1}^{K} q_i = a \geq 1$ and $q_j \geq 1$. Then, for any $\mathbf{q}' = (q_1', \ldots, q_K')$ such that $\sum_{i=1}^{K} q_i' = a + 1$ and $q_j' \geq 1$, we have $V(\mathbf{q}'; M - 1) = \alpha_j + V(\mathbf{q}' - \mathbf{e}_j; M)$ by Proposition 3.3.1 since Conditions (3.3.4) and (3.3.5) are satisfied. This shows that at all service completion times where $q_j \geq 1$, job j should receive the highest priority.

We next show that the result also holds for the decision given at time zero. Define

$$H(\mathbf{n}) = \sum_{k=1}^{K} \alpha_k n_k + V(\mathbf{m} - \mathbf{n}; M),$$

where $\mathbf{n} = (n_1, \ldots, n_K)$ and $\mathbf{m} = (m_1, \ldots, m_K)$, so that Equation (3.3.1) can be rewritten as $V(\mathbf{m}; 0) = \max_{\mathbf{n} \in \Phi} H(\mathbf{n})$. For a given $\mathbf{n} \in \Phi$, where $n_i \geq 1$ and $n_j < \min\{M, m_j\}$, let $\Delta_i(\mathbf{n}) = H(\mathbf{n} + \mathbf{e_j} - \mathbf{e_i}) - H(\mathbf{n})$ for $i \in \{1, \ldots, K\} \setminus \{j\}$. Then, for a fixed $i \in \{1, \ldots, K\} \setminus \{j\}$ and a given $n \in \Phi$, where $n_i \geq 1$ and $n_j < \min\{M, m_j\}$, $\Delta_i(\mathbf{n}) \geq 0$ if and only if

$$\alpha_j + V(\mathbf{m} - \mathbf{n} + \mathbf{e_i} - \mathbf{e_j}; M) \geq \alpha_i + V(\mathbf{m} - \mathbf{n}; M). \tag{A.0.4}$$

But in the first part of this proof, we already showed that it is optimal to serve a type j job in all states

$(\mathbf{q}; M-1)$, which implies that Condition (A.0.4) holds and hence $\Delta_i(\mathbf{n}) \geq 0$ for all $i \in \{1,\ldots,K\}\backslash\{j\}$ and $n \in \Phi$ such that $n_i \geq 1$ and $n_j < \min\{M, m_j\}$. This shows that allocating as many resources as possible to type j is optimal at time zero. □

Proof of Proposition 3.3.3: Define $\mathcal{C}_1 \subset \{1,\ldots,K\}$ to be the set of all types of jobs i such that $r_i < r_j$ and let $\mathcal{C}_2 \subseteq \{1,\ldots,K\}$ be the set of all types of jobs i such that $r_i \geq r_j$. Note that $\mathcal{C}_1 \cup \mathcal{C}_2 = \{1,\ldots,K\}$ and $\mathcal{C}_1$ can be an empty set whereas $\mathcal{C}_2$ is never an empty set because it always includes type j. By Proposition 1, we know that type j jobs should be prioritized against all types in $\mathcal{C}_1$ since for all $i \in \mathcal{C}_1$ $\alpha_i \leq \alpha_j$ and $r_i < r_j$. Hence, whenever there is at least one job from type j in the system, we can ignore all other types in $\mathcal{C}_1$. This reduces the problem to the one where the only types of jobs are those in $\mathcal{C}_2$. But for all types in $\mathcal{C}_2$ we have $r_i \geq r_j$ and also $\alpha_i r_i \leq \alpha_j r_j$, and hence by Proposition 3.3.2 type j should receive higher priority than all types in $\mathcal{C}_2$. This shows that type j jobs should receive the highest priority among all types $1,\ldots,K$. □

Proof of Proposition 3.3.4: First, note that if the optimal policy is an index policy, then it is sufficient to show that a type i job will be served under the optimal policy at state $(\mathbf{e}_i + \mathbf{e}_j; M-1)$ if and only if $\alpha_i r_i/(M\mu + r_i) \geq \alpha_j r_j/(M\mu + r_j)$, for $i,j \in \{1,\ldots,K\}$. Using Equations (3.3.2) and (3.3.3) multiple times, we obtain

$$\alpha_i + V(\mathbf{e}_j; M) - \alpha_j - V(\mathbf{e}_i; M) = \alpha_i + \frac{M\mu\alpha_j}{M\mu + r_j} - \alpha_j - \frac{M\mu\alpha_i}{M\mu + r_i} = \frac{\alpha_i r_i}{M\mu + r_i} - \frac{\alpha_j r_j}{M\mu + r_j}.$$

Hence, $V(\mathbf{e}_i + \mathbf{e}_j; M-1) = \alpha_i + V(\mathbf{e}_j; M)$ if and only if $\alpha_i r_i/(M\mu + r_i) \geq \alpha_j r_j(M\mu + r_j)$ for all $i,j \in \{1,\ldots,K\}$, which completes the proof. □

Proof of Proposition 3.4.1: (i) First, note that if $T_1 < 2$, there does not exist a state that would satisfy the condition that $q_1 + q_2 \leq T_1$ and $q_1, q_2 \geq 1$. Thus, consider the case where $T_1 \geq 2$. We will next use Proposition 3.3.1 and induction on $q_1 + q_2$ to prove the result.

When $q_1 + q_2 = 2$, the only state that satisfies the conditions that $q_1 + q_2 \leq T_1$ and $q_1, q_2 \geq 1$ is $(1, 1; M-1)$. In this state, serving a type 1 job is optimal if and only if $T_1 \geq 2 = q_1 + q_2$ (see Equation (3.3.2)), which shows that the result holds when $q_1 + q_2 = 2$. Next, suppose that it is optimal to serve a type 1 job in states $(q_1, q_2; M-1)$ such that $q_1 + q_2 = b$, for some integer b, $2 \leq b \leq T_1$, and $q_1, q_2 \geq 1$. Then, Proposition 3.3.1 implies that it is also optimal to serve a type 1 job at states $(q_1', q_2'; M-1)$, where $q_1' + q_2' = b + 1$ and $q_1', q_2' \geq 1$, if $q_1' + q_2' \leq T_1$. To see this, consider state $(q_1', q_2'; M - 1)$ with $q_1' + q_2' = b + 1$ and $q_1', q_2' \geq 1$. By the induction hypothesis we know that it is optimal to serve a type 1 job in states $(q_1' - 1, q_2'; M - 1)$ (if $q_1' \geq 2$) and $(q_1', q_2' - 1; M - 1)$ since $q_1' + q_2' - 1 = b$. Thus, if $q_1' + q_2' \leq T_1$ (i.e., Condition (3.3.4) for $i = 2$ and $j = 1$ is satisfied), then Proposition 3.3.1 tells that it is also optimal to serve a type 1 job in state $(q_1', q_2'; M - 1)$.

(ii) Suppose that there exists an integer $T \geq T_1$ such that at all states $(\mathbf{q}; M - 1)$, where $q_1 + q_2 = T$ and $q_1, q_2 \geq 1$, it is optimal to serve a type 2 job. Then, Proposition 3.3.1 implies that at all states $(q_1', q_2'; M - 1)$, where $q_1' + q_2' = T + 1$ and $q_1', q_2' \geq 1$, it is optimal to serve a type 2 job. To see this, consider states $(q_1', q_2'; M - 1)$, where $q_1' + q_2' = T + 1$ and $q_1', q_2' \geq 1$. It is given that serving a type 2 job is optimal in states $(q_1' - 1, q_2'; M - 1)$ and $(q_1', q_2' - 1; M - 1)$ (if $q_2' \geq 2$) since $q_1' + q_2' - 1 = T$. Furthermore, Condition (3.3.4) is satisfied for $j = 2$ and $i = 1$ since $q_1' + q_2' = T + 1 > T_1$. Thus, Proposition 3.3.1 tells that it is optimal to serve a type 2 job in states $(q_1', q_2'; M - 1)$, where $q_1' + q_2' = T + 1$ and $q_1', q_2' \geq 1$. Using the same argument successively shows that it is optimal to serve a type 2 job in all states $(q_1', q_2'; M - 1)$ such that $q_1' + q_2' \geq T + 1$ and $q_1', q_2' \geq 1$. □

Proof of Proposition 3.4.2: The optimal decision at time zero is trivial when $N = m_1 + m_2 \leq M$, $m_1 = 0$, or $m_2 = 0$. Hence, consider the case where $N \geq M + 1$, $m_1 \geq 1$, and $m_2 \geq 1$. We first rewrite Equation (3.3.1) as

$$V(m_1, m_2; 0) = \max_{\underline{n} \leq n \leq \overline{n}} G(n),$$

where $\underline{n} = \max\{0, M - m_1\}$, $\overline{n} = \min\{M, m_2\}$, and

$$G(n) = \alpha_1(M - n) + \alpha_2 n + V(m_1 - M + n, m_2 - n; M).$$

Here n is a decision variable that denotes the number of servers allocated to type 2 jobs at time zero. Note that $\underline{n} < \overline{n}$ (since $N \geq M + 1$, $m_1 \geq 1$, and $m_2 \geq 1$), which implies that there are at least two values that n can take. Now let

$$\Delta(n) = G(n + 1) - G(n), \text{ for } \underline{n} \leq n \leq \overline{n} - 1.$$

Using Equation (3.3.2), we make the following observation.

<u>Observation:</u> For a fixed $n \in \{\underline{n}, \ldots, \overline{n} - 1\}$, $\Delta(n) \geq 0$ if and only if serving a type 2 job is optimal in

state $(m_1 - M + n + 1, m_2 - n; M - 1)$. Also, $\Delta(n) = 0$ if and only if serving either a type 1 or type 2 job is optimal in state $(m_1 - M + n + 1, m_2 - n; M - 1)$.

We next use this observation together with Proposition 3.4.1 to complete the proof.

(i) By Proposition 3.4.1, if $m_1 + m_2 - M + 1 \leq T_1$, then serving a type 1 job is optimal in states $(m_1 - M + n + 1, m_2 - n; M - 1)$ for all $\underline{n} \leq n \leq \overline{n} - 1$, which is equivalent to having $\Delta(n) \leq 0$ for all $\underline{n} \leq n \leq \overline{n} - 1$. This implies that at time zero the optimal policy sets $n = \underline{n}$, i.e., allocates as many servers as possible to type 1 jobs.

(ii) By Proposition 3.4.1, if $m_1 + m_2 - M + 1 \geq T_2$, then serving a type 2 job is optimal in states $(m_1 - M + n + 1, m_2 - n; M - 1)$ for all $\underline{n} \leq n \leq \overline{n} - 1$, which is equivalent to having $\Delta(n) \geq 0$ for all $\underline{n} \leq n \leq \overline{n} - 1$. This implies that at time zero the optimal policy sets $n = \overline{n}$, i.e., allocates as many servers as possible to type 2 jobs. □

Proof of Proposition 3.4.3: First note that for a given N, all feasible states $(\mathbf{q}; M - 1)$ satisfy the condition that $q_1 + q_2 \leq N - M$. Hence, if $T_1 \geq N - M$, then by part (i) of Proposition 3.4.1, type 1 jobs should be prioritized at each state $(\mathbf{q}; M - 1)$, where $\mathbf{q} \in \{(q_1, q_2) : q_1 = 1, \ldots, m_1, q_2 = 1, \ldots, m_2; q_1 + q_2 \leq N - M\}$. Combining this with part (i) of Proposition 3.4.2 completes the proof. □

Proof of Proposition 3.4.4: We first show that the result holds for all decision epochs at which a service completion takes place. Condition (3.4.3) implies that it is optimal to serve a type 2 job in state $(1, 1; M - 1)$, see Equation (3.3.2). Condition 3.4.3 also implies that $T_1 \leq 2$. Hence, there exists an integer T, where $T \geq T_1$, such that at all states $(\mathbf{q}; M - 1)$, where $q_1 + q_2 = T$ and $q_1, q_2 \geq 1$, it is optimal to serve a type 2 job. Then from the proof of part (ii) of Proposition 3.4.1, we conclude that at all states $(\mathbf{q}; M - 1)$, where $q_1, q_2 \geq 1$, it is optimal to serve a type 2 job. This also implies that the optimal policy allocates as many servers as possible to type 2 jobs at time zero. To see this, note that $\Delta(n)$, which is defined in the proof of Proposition 3.4.2, is greater than or equal to zero for all n by using the observation made in the proof of Proposition 3.4.2 and the fact that serving a type 2 job is optimal in all states $(\mathbf{q}; M - 1)$, where $q_1, q_2 \geq 1$. This completes the proof. □

Proof of Proposition 3.5.1: (i) Consider a decision epoch such that jobs from type $1, 2, \ldots, j$ are available for service where $j \geq 2$. Suppose that $r_1 \leq r_2 \leq \cdots \leq r_j$ and $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_j$. By Corollary 3.2.1, type j receives the highest priority under the optimal policy. We next show that the

2-step policy, myopic policy, $\alpha r\mu$-rule, and TCF rule all prioritize type j jobs.

1. 2-step policy: For $i = 1, \ldots, j-1$, let

$$\Lambda_{ij}(\mathbf{q}) = \alpha_j + \frac{M\mu \max_j(\mathbf{q})}{M\mu - r_j + \sum_{k=1}^{K} q_k r_k} - \alpha_i - \frac{M\mu \max_i(\mathbf{q})}{M\mu - r_i + \sum_{k=1}^{K} q_k r_k}, \tag{A.0.5}$$

where $\max_k(\mathbf{q}) := \max\left\{\mathbb{I}_{\{q_k \geq 2\}}\alpha_k, \max_{l\in\{1,\ldots,K\}\setminus\{k\}}\{\mathbb{I}_{\{q_l\geq 1\}}\alpha_l\}\right\}$ for $q_i, q_j \geq 1$. We next show that $\Lambda_{ij}(\mathbf{q}) \geq 0$ for all $i < j$, which implies that type j jobs are preferred over type i jobs under the 2-step policy at every service completion instant when $q_i, q_j \geq 1$. First, note that $\alpha_i \leq \alpha_j$ implies that $0 \leq \max_i(\mathbf{q}) - \max_j(\mathbf{q}) \leq \alpha_j - \alpha_i$ for all $\mathbf{q}$ such that $q_i, q_j \geq 1$. Using this inequality together with $r_i \leq r_j$ and $\alpha_i \leq \alpha_j$, we have

$$\begin{aligned}
\Lambda_{ij}(\mathbf{q}) &= \alpha_j - \alpha_i - \frac{M\mu(\max_i(\mathbf{q}) - \max_j(\mathbf{q}))}{M\mu - r_i + \sum_{k=1}^{K} q_k r_k} \\
&\quad + \frac{M\mu \max_j(\mathbf{q})(r_j - r_i)}{(M\mu - r_j + \sum_{k=1}^{K} q_k r_k)(M\mu - r_i + \sum_{k=1}^{K} q_k r_k)} \qquad \text{(A.0.6)} \\
&\geq \alpha_j - \alpha_i - \frac{M\mu(\alpha_j - \alpha_i)}{M\mu - r_i + \sum_{k=1}^{K} q_k r_k} \\
&= \frac{(\alpha_j - \alpha_i)(-r_i + \sum_{k=1}^{K} q_k r_k)}{M\mu - r_i + \sum_{k=1}^{K} q_k r_k} \geq 0. \qquad \text{(A.0.7)}
\end{aligned}$$

Hence, the 2-step policy behaves the same as the optimal policy at all service completions. We next show that type j jobs are preferred over type i jobs at time zero. Let

$$Q(\mathbf{n}) = \sum_{k=1}^{K} \alpha_k n_k + \frac{M\mu \max_{k\in\{1,\ldots,K\}}\{\mathbb{I}_{\{m_k - n_k \geq 1\}}\alpha_k\}}{M\mu + \sum_{k=1}^{K}(m_k - n_k) r_k},$$

where $\mathbf{m} := (m_1, \ldots, m_K)$ is the vector of number of jobs for each type in the system at time zero and $\mathbf{n} = (n_1, \ldots, n_K)$ is the vector of the number of servers allocated to each job type at time zero, where $\mathbf{n} \in \Phi$. Then, allocating a server to a type j job instead of a type i job is preferred under the 2-step policy at time zero if and only if $Q(\mathbf{n}+\mathbf{e_j}-\mathbf{e_i}) \geq Q(\mathbf{n})$ for all $\mathbf{n} \in \Phi$, $m_j - n_j \geq 1$ and $n_i \geq 1$. From Equation (A.0.5), we have $\Lambda_{ij}(\mathbf{m} - \mathbf{n} + \mathbf{e_i}) = Q(\mathbf{n} + \mathbf{e_j} - \mathbf{e_i}) - Q(\mathbf{n})$ for $\mathbf{n} \in \Phi$, $m_j - n_j \geq 1$, and $n_i \geq 1$. Thus, by (A.0.7), we have $Q(\mathbf{n} + \mathbf{e_j} - \mathbf{e_i}) \geq Q(\mathbf{n})$ for all $i < j$ such that $m_j - n_j \geq 1$, $n_i \geq 1$, and $\mathbf{n} \in \Phi$, which implies that type j jobs are preferred over type i jobs also at time zero.

2. Myopic policy: For any $i < j$, we have

$$\frac{\alpha_j r_j}{M\mu + r_j} - \frac{\alpha_i r_i}{M\mu + r_i} = \frac{M\mu(\alpha_j r_j - \alpha_i r_i) + r_i r_j(\alpha_j - \alpha_i)}{(M\mu + r_j)(M\mu + r_i)} \geq 0$$

   since $\alpha_i r_i \leq \alpha_j r_j$ and $\alpha_i \leq \alpha_j$, which implies that the myopic policy prefers type j jobs over type i jobs.

3. $\alpha r\mu$-rule: For any $i < j$, the $\alpha r\mu$-rule will prefer type j jobs over type i jobs because $\alpha_i r_i \leq \alpha_j r_j$.

4. TCF rule: For any $i < j$, the TCF rule will prefer type j jobs over type i jobs since $r_i \leq r_j$.

(ii) Consider a decision epoch such that jobs from type $1, 2, \ldots, j$ are available for service where $j \geq 2$. Suppose that $r_1 \geq r_2 \geq \cdots \geq r_j$ and $\alpha_1 r_1 \leq \alpha_2 r_2 \leq \cdots \leq \alpha_j r_j$ (and hence $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_j$). By Corollary 3.3.1, type j receives the highest priority under the optimal policy. Clearly, the TCF rule is not consistent with this result as it gives priority to the jobs with the largest abandonment rate. On the other hand, the $\alpha r\mu$-rule is consistent because it gives priority to the jobs with the largest αr value. We next show that the remaining four heuristics are consistent with Corollary 3.3.1.

1. 2-step policy: For any $i < j$, we have

$$\begin{aligned} \Lambda_{ij}(\mathbf{e}_i + \mathbf{e}_j) &= \alpha_j + \frac{M\mu\alpha_i}{M\mu + r_i} - \alpha_i - \frac{M\mu\alpha_j}{M\mu + r_j} \\ &= \frac{M\mu(\alpha_j r_j - \alpha_i r_i) + r_i r_j(\alpha_j - \alpha_i)}{(M\mu + r_j)(M\mu + r_i)} \geq 0 \end{aligned} \qquad \text{(A.0.8)}$$

   since $\alpha_i \leq \alpha_j$ and $\alpha_i r_i \leq \alpha_j r_j$. Next, from Equation (A.0.6), we observe that $\Lambda_{ij}(\mathbf{q})$ is increasing in q_k for any $k \in \{1, \ldots, K\}$ as $\alpha_i \leq \alpha_j$ (and hence $\max_i(\mathbf{q}) - \max_j(\mathbf{q}) \geq 0$ for all $\mathbf{q}$ such that $q_i, q_j \geq 1$) and $r_i \geq r_j$. Combining this with (A.0.8), we conclude that $\Lambda_{ij}(\mathbf{q}) \geq 0$ for any $\mathbf{q}$ such that $q_i, q_j \geq 1$, and therefore the 2-step policy is consistent with Corollary 3.3.1 at all service completions. Furthermore, similar to the proof of the consistency with Corollary 3.2.1 at time zero, we have $\Lambda_{ij}(\mathbf{m} - \mathbf{n} + \mathbf{e_i}) = Q(\mathbf{n} + \mathbf{e_j} - \mathbf{e_i}) - Q(\mathbf{n}) \geq 0$ for all $i < j$ such that $m_j - n_j \geq 1$, $n_i \geq 1$, and $\mathbf{n} \in \Phi$, which implies that type j jobs are preferred over type i jobs at time zero.

2. Threshold-1 policy: For any $i < j$, we have

   $$T_{i,j} = \frac{M\mu(\alpha_i r_i - \alpha_j r_j)}{(\alpha_j - \alpha_i) r_i r_j} + 1 \leq 1$$

   since $\alpha_i r_i \leq \alpha_j r_j$ and $\alpha_i < \alpha_j$. This implies that $\mathcal{T}_i \leq 1$,and hence type j jobs are preferred over type i jobs at all decision epochs.

3. Threshold-2 policy: For any $i < j$, we will consider two cases. First, assume that $\alpha_i = \alpha_j$ and $r_i = r_j$, i.e., the two types are essentially identical. In this case, the consistency follows trivially. Next, assume that the two types are not identical. Then, from (A.0.8), we have $\Lambda_{ij}(\mathbf{e_i} + \mathbf{e_j}) > 0$. Furthermore, from Equation (A.0.6) and the conditions that $\alpha_i \leq \alpha_j$ (and hence $\max_i(\mathbf{q}) - \max_j(\mathbf{q}) \geq 0$ for all $\mathbf{q}$ such that $q_i, q_j \geq 1$) and $r_i \geq r_j$, $\Lambda_{ij}(\mathbf{q})$ is increasing in q_k for any $k \in \{1, \ldots, K\}$. Then, we conclude that $\Lambda_{ij}(\mathbf{q}) > 0$ for any $\mathbf{q}$ such that $q_i, q_j \geq 1$. Hence, when we solve $\Lambda_{ij}(\mathbf{q}) = 0$ for q_j [q_i], by letting $q_k = 0$ for all $k \neq i, j$ and $q_i = 1$ [$q_j = 1$], any solution must be less than one. Therefore, $\mathcal{T}_i < 1$ for all $i < j$, which implies that type j jobs are preferred over type i jobs at all decision epochs.

4. Myopic policy: Same proof as in the proof of part (i) also applies here.

(iii) Consider a decision epoch such that jobs from type $1, 2, \ldots, j$ are available for service where $j \geq 2$. Suppose that $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_j$ and $\alpha_1 r_1 \leq \alpha_2 r_2 \leq \cdots \leq \alpha_j r_j$. By Corollary 3.3.2, type j receives the highest priority under the optimal policy. Clearly, the TCF rule is not consistent with this result as it is possible to have $r_i \geq r_j$ for some $i = 1, \ldots, j-1$, in which case it will prioritize type i jobs over type j jobs. We can however show that the remaining five heuristics are consistent with Corollary 3.3.2 as serving a type j job instead of a type i job is the preferred action for all $i < j$ under these five policies. For any fixed $i < j$, consider the following cases:

Case 1 ($r_i \leq r_j$): The result follows from the arguments used in the proof of part (i).

Case 2 ($r_i > r_j$): The result follows from the arguments used in the proof of part (ii).

(iv) Suppose that the conditions of Proposition 3.4.4 hold, and hence it is optimal to give priority to type 2 jobs. Under these conditions, the $\alpha r\mu$-rule [TCF rule] is not necessarily consistent with Proposition 3.4.4 as it is possible to have $\alpha_1 r_1 \geq \alpha_2 r_2$ [$r_1 \geq r_2$], in which case it gives priority to type 1 jobs. On the other hand, the myopic policy agrees with the optimal policy under the conditions of Proposition

3.4.4 by definition. We next show that the remaining three heuristics are consistent with Proposition 3.4.4 as serving a type 2 job instead of a type 1 job is the preferred action under these heuristics. Two cases exhaust all possible scenarios:

Case 1 ($r_2 \geq r_1$): The result trivially follows from the proof of part (i).

Case 2 ($r_2 < r_1$):

1. 2-step policy: By Condition (3.4.3), we have

$$\Lambda_{12}(1,1) = \alpha_2 + \frac{M\mu\alpha_1}{M\mu + r_1} - \alpha_1 - \frac{M\mu\alpha_2}{M\mu + r_2} = \frac{\alpha_2 r_2}{M\mu + r_2} - \frac{\alpha_1 r_1}{M\mu + r_1} \geq 0. \qquad \text{(A.0.9)}$$

   Furthermore, from the proof of part (ii), we know that $\Lambda_{12}(q_1, q_2)$ is increasing in q_1 and q_2 when $\alpha_2 > \alpha_1$ and $r_2 < r_1$. Hence, we conclude that $\Lambda_{12}(q_1, q_2) \geq 0$ for any $q_1, q_2 \geq 1$, which implies that the 2-step policy is consistent with Proposition 3.4.4 when $r_2 < r_1$ at service completion instants. In order to show that the policy agrees with Proposition 3.4.4 also at time zero, we use an argument similar to that used in the proof of part (i). In particular, we have $\Lambda_{12}(m_1 - n_1 + 1, m_2 - n_2) = Q(n_1 - 1, n_2 + 1) - Q(n_1, n_2) \geq 0$ for $m_2 - n_2 \geq 1$, $n_1 \geq 1$, and $(n_1, n_2) \in \Phi$, which implies that type 2 jobs are preferred over type 1 jobs at time zero.

2. Threshold-1 policy: Under the conditions of Proposition 3.4.4, we can show that

$$T_{1,2} = \frac{M\mu(\alpha_1 r_1 - \alpha_2 r_2)}{(\alpha_2 - \alpha_1) r_1 r_2} + 1 \leq 2.$$

   Furthermore, inequality (A.0.9) shows that type 2 jobs are prioritized when $q_1 = q_2 = 1$. Combining these two facts, we conclude that $\mathcal{T}_1 \leq 1$, and hence type 2 jobs are preferred over type 1 jobs at all decision epochs.

3. Threshold-2 policy: First of all, from (A.0.9), we know that $\Lambda_{12}(1,1) \geq 0$. Furthermore, from Equation (A.0.6) and the conditions that $\alpha_2 > \alpha_1$ and $r_2 < r_1$, we have $\Lambda_{12}(q_1, q_2)$ is strictly increasing in q_1 and q_2 for $q_1, q_2 \geq 1$. Then, we conclude that $\Lambda_{12}(q_1, q_2) > 0$ for any $\mathbf{q}$ such that at least one of q_1 and q_2 is strictly greater than 1. Hence, any solution to $\Lambda_{12}(q_1, 1) = 0$ and $\Lambda_{12}(1, q_2) = 0$ must be at most one. Therefore, $\mathcal{T}_1 \leq 1$, which implies that type 2 jobs are preferred over type 1 jobs at all decision epochs. □

Proof of Proposition 4.1.1: We will again use a coupling argument to prove the result. Suppose policy π takes job j into service at t_0 while job i is in the queue. Without loss of generality, assume that $t_0 = 0$. We will construct a policy γ which serves job i at time 0, and for which $C_\pi(t) \leq C_\gamma(t)$ for all $t \geq 0$ along any given sample path.

Let Y_l^ρ denote the remaining lifetime of job l at time 0 under policy ρ, where $l \in \{i, j\}$ and $\rho \in \{\pi, \gamma\}$. Note that by the stochastic ordering relation among the remaining lifetimes of jobs, we can couple the random variables so that $Y_i^\pi = y_i \leq y_j = Y_j^\gamma$. Because policy $\pi(\gamma)$ serves job $j(i)$ at time zero and the job in service will not abandon, we do not need Y_j^π and Y_i^γ. Let $Y_l^\gamma = Y_l^\pi$ for all $l \neq i, j$. Let also S_l^ρ denote the service time of job l under policy $\rho \in \{\pi, \gamma\}$, and let $S_l^\gamma = S_l^\pi$ for all $l \neq i, j$. We can couple (S_i^π, S_j^π) with (S_i^γ, S_j^γ) so that $\underline{m} := \min\{S_i^\pi, S_j^\pi\} = \min\{S_i^\gamma, S_j^\gamma\} \leq \overline{m} := \max\{S_i^\pi, S_j^\pi\} = \max\{S_i^\gamma, S_j^\gamma\}$ and either $S_j^\pi = S_i^\gamma := a \in \{\underline{m}, \overline{m}\}$ and $S_i^\pi = S_j^\gamma := b \in \{\underline{m}, \overline{m}\} \backslash \{a\}$ (Case I) or $S_i^\pi = S_i^\gamma = \underline{m} \leq S_j^\pi = S_j^\gamma = \overline{m}$ (Case II). Note that such a coupling is possible from Lemma A.0.1 and the condition that $S_i \leq_{lr} S_j$. Finally, let Z_l^ρ denote the reward of serving job l under policy $\rho \in \{\pi, \gamma\}$ and let $Z_l^\gamma = Z_l^\pi$ for all $l \neq i, j$. Then, we can couple (Z_i^π, Z_j^π) with (Z_i^γ, Z_j^γ) so that $\min\{Z_i^\pi, Z_j^\pi\} = \min\{Z_i^\gamma, Z_j^\gamma\} \leq \max\{Z_i^\pi, Z_j^\pi\} = \max\{Z_i^\gamma, Z_j^\gamma\}$, and either $Z_j^\pi = Z_i^\gamma$ and $Z_i^\pi = Z_j^\gamma$ or $Z_j^\pi = Z_j^\gamma \leq Z_i^\pi = Z_i^\gamma$. Such a coupling is possible from Lemma A.0.1 and the condition that $Z_j \leq_{lr} Z_i$. Let τ be the time instance π takes job i into service ($\tau = \infty$ if job i is not taken into service). The following cases exhaust all possibilities:

Case I:

(a) We first consider the case where $\tau < \infty$. Note that, under Case I, the fist decision epoch after time zero is at time a for both π and γ. γ follows π during $[a, \tau)$, and at time τ, when π takes job i into service, γ takes job j. This is possible because $y_i \leq y_j$. At time $\tau + b$, the states will be the same under both policies and γ follows π from then on. Hence, we have $C_\gamma(t) - C_\pi(t) = Z_i^\gamma - Z_j^\pi \geq 0$ for all $0 \leq t < \tau$, and $C_\gamma(t) - C_\pi(t) = Z_i^\gamma + Z_j^\gamma - Z_i^\pi - Z_j^\pi = 0$ for all $t \geq \tau$.

(b) Now suppose that $\tau = \infty$. Then, γ follows π at all decision epochs after time zero except that it serves job j last (let the service start time be τ') if it is still available after all other jobs are cleared. Hence, we have $C_\gamma(t) - C_\pi(t) = Z_i^\gamma - Z_j^\pi \geq 0$ for all $0 \leq t < \tau'$, and if $\tau' < \infty$, $C_\gamma(t) - C_\pi(t) = Z_i^\gamma - Z_j^\pi + Z_j^\gamma \geq 0$ for all $t \geq \tau'$.

Case II:

(a) We again first consider the case where $\tau < \infty$. γ follows π at every decision epoch during $[\underline{m}, \tau - \overline{m} + \underline{m})$ and serves job j at time $\tau - \overline{m} + \underline{m}$ when π serves job i (at time τ). This is possible since $y_i \leq y_j$ and the service completion times under γ are $\overline{m} - \underline{m}$ units of time earlier than those under π between $\overline{m}$ and τ. The states under π and γ become the same at time $\underline{m} + \tau$, and γ follows π from then on. Hence, we have $C_\gamma(t) - C_\pi(t) = Z_i^\gamma - Z_j^\pi \geq 0$ for all $0 \leq t < \underline{m}$, $C_\gamma(t) - C_\pi(t) \geq Z_i^\gamma - Z_j^\pi \geq 0$ for all $\underline{m} \leq t < \tau - \overline{m} + \underline{m}$, $C_\gamma(t) - C_\pi(t) \geq Z_i^\gamma - Z_j^\pi + Z_j^\gamma \geq 0$ for all $\tau - \overline{m} + \underline{m} \leq t < \tau$, and $C_\gamma(t) - C_\pi(t) = Z_i^\gamma + Z_j^\gamma - Z_i^\pi - Z_j^\pi = 0$ for all $t \geq \tau$.

(b) We now consider the case where $\tau = \infty$. γ follows π starting at time m, except that it serves job j last (let the service start time be τ') if it is still available when all other jobs are cleared. As in Case II(a), this is possible since the service completion times under γ are $\overline{m} - \underline{m}$ units of time earlier than those under π after $\overline{m}$. Then, we have $C_\gamma(t) - C_\pi(t) = Z_i^\gamma - Z_j^\pi \geq 0$ for all $0 \leq t < \underline{m}$, $C_\gamma(t) - C_\pi(t) \geq Z_i^\gamma - Z_j^\pi \geq 0$ for all $\underline{m} \leq t < \tau'$, and if $\tau' < \infty$, $C_\gamma(t) - C_\pi(t) = Z_i^\gamma - Z_j^\pi + Z_j^\gamma \geq 0$ for all $t \geq \tau'$.

Thus, we have shown that $C_\gamma(t) \geq C_\pi(t)$ for all $t \geq 0$ along any sample path. □

In the proof of Proposition 4.2.1, we use the following lemma, which states that for a fixed number of jobs in queue, we prefer to have the job in service be a job with a smaller mean service time. This makes sense because the remaining lifetime of the job in service and the associated reward for that job are no longer relevant.

Lemma A.0.2. *If $\mu_j \leq (=)\ \mu_i$ for any pair (i, j), $i, j = 1, \ldots, K$, then $V(\mathbf{q}; P_j) \leq (=)\ V(\mathbf{q}; P_i)$.*

Proof: We first prove the inequality part. Couple the processing times of the jobs in service for the two states such that $S_i' \leq S_j'$ with probability one, where S_l' denotes the processing time of the type l job in service, $l \in \{i, j\}$. Let $V_0(\mathbf{q}; P_i)$ be the value function when the starting state is $(\mathbf{q}; P_i)$ and we idle from time S_i' to S_j' and then follow the optimal policy. Then, from Proposition 1, we have $V(\mathbf{q}; P_i) \geq V_0(\mathbf{q}; P_i) = V(\mathbf{q}; P_j)$.

We next prove the equality part. Couple the processing times of the jobs in service for the two states such that $S_i' = S_j'$ with probability one. Then, starting from states $(\mathbf{q}; P_i)$ and $(\mathbf{q}; P_j)$, the processes reach the same state after the service completion of the job in service, i.e., at $S_i' = S_j'$. Hence, we have $V(\mathbf{q}; P_i) = V(\mathbf{q}; P_j)$. □

Proof of Proposition 4.2.1: We will show that $\alpha_j + V(\mathbf{q}-\mathbf{e}_j;P_j) \geq \mathbb{I}_{\{q_i \geq 1\}}\alpha_i + V(\mathbf{q}-\mathbf{e}_i;P_i)$ for all $i = 1,\ldots,K$ under the given conditions. For $i \in \{1,\ldots,K\}$ such that $q_i = 0$, this holds trivially. Hence, we only consider the types of jobs for which $q_i \geq 1$.

For a fixed $i \in \{1,\ldots,K\} \setminus \{j\}$ such that $q_i \geq 1$, we have

$$\begin{aligned}
V(\mathbf{q}-\mathbf{e}_j;P_j) &= \frac{\mu_j V(\mathbf{q}-\mathbf{e}_j;R) + \sum_{k=1,k\neq j}^K q_k r_k V(\mathbf{q}-\mathbf{e}_k-\mathbf{e}_j;P_j) + (q_j-1)r_j V(\mathbf{q}-2\mathbf{e}_j;P_j)}{\mu_j - r_j + \sum_{k=1}^K q_k r_k} \\
&\geq \frac{1}{\mu_j - r_j + \sum_{k=1}^K q_k r_k}\Bigg\{\mu_j(\alpha_i + V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_i)) + \sum_{k=1,k\neq i,j}^K q_k r_k (V(\mathbf{q}-\mathbf{e}_k-\mathbf{e}_i;P_i) + \alpha_i - \alpha_j) \\
&+ q_i r_i V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_j) + (q_j-1)r_j(V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_i) + \alpha_i - \alpha_j)\Bigg\} \\
&= \frac{1}{\mu_j - r_j + \sum_{k=1}^K q_k r_k}\Bigg\{\alpha_j\mu_j + (\alpha_i - \alpha_j)(\mu_j - r_j + \sum_{k=1,k\neq i}^K q_k r_k) + q_i r_i V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_j) \\
&+ \sum_{k=1,k\neq i,j}^K q_k r_k V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_k;P_k) + (\mu_j + (q_j-1)r_j)V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_i)\Bigg\}, \qquad \text{(A.0.10)}
\end{aligned}$$

where the inequality follows because, for the first term, $V(\mathbf{q}-\mathbf{e}_j;R) \geq \alpha_i + V(\mathbf{q}-\mathbf{e}_j-\mathbf{e}_i;P_i)$; for the second term, $V(\mathbf{q}-\mathbf{e}_k;R) = \alpha_j + V(\mathbf{q}-\mathbf{e}_k-\mathbf{e}_j;P_j)$ for all $k \in \{1,\ldots,K\} \setminus \{j\}$ such that $q_k \geq 1$; and for the last term, either $q_j = 1$, so the inequality is trivial or by the condition that $\alpha_j + V(\mathbf{q}-2\mathbf{e}_j;P_j) \geq \alpha_i + V(\mathbf{q}-\mathbf{e}_j-\mathbf{e}_i;P_i)$ for $q_j \geq 2$. Furthermore, Equation (A.0.10) holds because $V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_k;P_i) = V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_k;P_k)$ for all $i,k \in \{1,\ldots,K\} \setminus \{j\}$ by Condition (4.2.6) and Lemma A.0.2.

Similarly, for a fixed $i \in \{1,\ldots,K\} \setminus \{j\}$ such that $q_i \geq 1$, we have

$$\begin{aligned}
V(\mathbf{q}-\mathbf{e}_i;P_i) &= \frac{\mu_i V(\mathbf{q}-\mathbf{e}_i;R) + (q_i-1)r_i V(\mathbf{q}-2\mathbf{e}_i;P_i) + \sum_{k=1,k\neq i}^K q_k r_k V(\mathbf{q}-\mathbf{e}_k-\mathbf{e}_i;P_i)}{\mu_i - r_i + \sum_{k=1}^K q_k r_k} \\
&\leq \frac{1}{\mu_i - r_i + \sum_{k=1}^K q_k r_k}\Bigg\{\mu_i(\alpha_j + V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_j)) + (q_i-1)r_i(V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_j) + \alpha_j - \alpha_i) \\
&+ \sum_{k=1,k\neq i,j}^K q_k r_k V(\mathbf{q}-\mathbf{e}_k-\mathbf{e}_i;P_i) + q_j r_j V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_i)\Bigg\} \\
&= \frac{1}{\mu_i - r_i + \sum_{k=1}^K q_k r_k}\Bigg\{(\alpha_j - \alpha_i)q_i r_i + \alpha_j\mu_i - (\alpha_j - \alpha_i)r_i + (\mu_i + (q_i-1)r_i)V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_j) \\
&+ \sum_{k=1,k\neq i,j}^K q_k r_k V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_k;P_k) + q_j r_j V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_i)\Bigg\}, \qquad \text{(A.0.11)}
\end{aligned}$$

where the inequality follows because, for the first and second terms, $V(\mathbf{q}-\mathbf{e}_i;R) = \alpha_j + V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_j)$; and Equation (A.0.11) holds because $V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_k;P_i) = V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_k;P_k)$ for all $i,k \in \{1,\ldots,K\}\setminus\{j\}$ by Condition (4.2.6) and Lemma A.0.2.

Now, from (A.0.10) and (A.0.11), we get

$$\begin{aligned}
&\alpha_j + V(\mathbf{q}-\mathbf{e}_j;P_j) - \alpha_i - V(\mathbf{q}-\mathbf{e}_i;P_i)\\
\geq\ & (\alpha_j-\alpha_i)\left(1 - \frac{\mu_j - r_j + \sum_{k=1,k\neq i}^K q_k r_k}{\mu_j - r_j + \sum_{k=1}^K q_k r_k} - \frac{q_i r_i}{\mu_i - r_i + \sum_{k=1}^K q_k r_k}\right)\\
+\ & \frac{\alpha_j\mu_j(\mu_i - r_i + \sum_{k=1}^K q_k r_k) + ((\alpha_j-\alpha_i)r_i - \alpha_j\mu_i)(\mu_j - r_j + \sum_{k=1}^K q_k r_k)}{(\mu_j - r_j + \sum_{k=1}^K q_k r_k)(\mu_i - r_i + \sum_{k=1}^K q_k r_k)}\\
+\ & \left(\frac{q_i r_i}{\mu_j - r_j + \sum_{k=1}^K q_k r_k} - \frac{\mu_i + (q_i-1)r_i}{\mu_i - r_i + \sum_{k=1}^K q_k r_k}\right) V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_j)\\
+\ & \left(\frac{1}{\mu_j - r_j + \sum_{k=1}^K q_k r_k} - \frac{1}{\mu_i - r_i + \sum_{k=1}^K q_k r_k}\right) \sum_{k=1,k\neq i,j}^K q_k r_k V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_k;P_k)\\
+\ & \left(\frac{\mu_j + (q_j-1)r_j}{\mu_j - r_j + \sum_{k=1}^K q_k r_k} - \frac{q_j r_j}{\mu_i - r_i + \sum_{k=1}^K q_k r_k}\right) V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_i)
\end{aligned}\tag{A.0.12}$$

$$\begin{aligned}
\geq\ & \left(\frac{1}{\mu_j - r_j + \sum_{k=1}^K q_k r_k} - \frac{1}{\mu_i - r_i + \sum_{k=1}^K q_k r_k}\right)(\alpha_j-\alpha_i)q_i r_i\\
+\ & \frac{((\alpha_j-\alpha_i)r_i + \alpha_j(\mu_j-\mu_i))\sum_{k=1}^K q_k r_k + \alpha_i r_i(r_j-\mu_j) - \alpha_j r_j(r_i-\mu_i)}{(\mu_j - r_j + \sum_{k=1}^K q_k r_k)(\mu_i - r_i + \sum_{k=1}^K q_k r_k)}\\
+\ & \left(\frac{q_i r_i}{\mu_j - r_j + \sum_{k=1}^K q_k r_k} - \frac{\mu_i + (q_i-1)r_i}{\mu_i - r_i + \sum_{k=1}^K q_k r_k}\right) V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_j)\\
+\ & \left(\frac{1}{\mu_j - r_j + \sum_{k=1}^K q_k r_k} - \frac{1}{\mu_i - r_i + \sum_{k=1}^K q_k r_k}\right) \sum_{k=1,k\neq i,j}^K q_k r_k \left(V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_j) + \alpha_j - \alpha_k\right)\\
+\ & \left(\frac{\mu_j + (q_j-1)r_j}{\mu_j - r_j + \sum_{k=1}^K q_k r_k} - \frac{q_j r_j}{\mu_i - r_i + \sum_{k=1}^K q_k r_k}\right) V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_i)\\
=\ & \frac{(r_i - \mu_i - r_j + \mu_j)\sum_{k=1,k\neq j}^K \alpha_k q_k r_k + (\alpha_j r_j - \alpha_i r_i)\sum_{k=1,k\neq j}^K q_k r_k}{(\mu_j - r_j + \sum_{k=1}^K q_k r_k)(\mu_i - r_i + \sum_{k=1}^K q_k r_k)}\\
+\ & \frac{q_j r_j((\alpha_j-\alpha_i)r_i + \alpha_j(\mu_j-\mu_i)) + \alpha_i r_i(r_j-\mu_j) - \alpha_j r_j(r_i-\mu_i)}{(\mu_j - r_j + \sum_{k=1}^K q_k r_k)(\mu_i - r_i + \sum_{k=1}^K q_k r_k)}\\
+\ & \left(\frac{\mu_j + (q_j-1)r_j}{\mu_j - r_j + \sum_{k=1}^K q_k r_k} - \frac{q_j r_j}{\mu_i - r_i + \sum_{k=1}^K q_k r_k}\right)(V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_i) - V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_j))\\
=\ & \frac{\sum_{k=1}^K q_k r_k[\alpha_j r_j - \alpha_i r_i + \alpha_k(r_i - \mu_i - r_j + \mu_j)] - \alpha_j r_j(r_i-\mu_i) + \alpha_i r_i(r_j-\mu_j)}{(\mu_j - r_j + \sum_{k=1}^K q_k r_k)(\mu_i - r_i + \sum_{k=1}^K q_k r_k)}
\end{aligned}$$

$$
\begin{aligned}
&+ \quad \frac{(r_i-\mu_i)q_jr_j+(r_j-\mu_j)(\mu_i-r_i+\sum_{k=1,k\neq j}^{K} q_kr_k)}{(\mu_j-r_j+\sum_{k=1}^{K} q_kr_k)(\mu_i-r_i+\sum_{k=1}^{K} q_kr_k)} \\
&\times \quad (V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_j)-V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_i)), \qquad \text{(A.0.13)}
\end{aligned}
$$

where the second inequality follows because for the fourth term of (A.0.12) either $K=2$ so the inequality holds trivially or it follows from Condition (4.2.5) and the condition that $V(\mathbf{q}-\mathbf{e}_i;R) = \alpha_j + V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_j)$ for all $i \in \{1,\ldots,K\}\setminus\{j\}$ such that $q_i \geq 1$. Finally, Equation (A.0.13) is greater than or equal to zero, because, for the first term, Condition (4.2.3) holds; and for the second term, Condition (4.2.4) holds and we have $V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_i) \leq (\geq) V(\mathbf{q}-\mathbf{e}_i-\mathbf{e}_j;P_j)$ if $\mu_i \leq (\geq)\mu_j$ by Lemma A.0.2.□

Proof of Proposition 4.2.2: First of all, when $r_j \geq r_i$ for $i \in \{1,\ldots,K\}$, Proposition 4.1.1 states that it is optimal to give priority to type j jobs over type i jobs at all decision epochs since $\alpha_j \geq \alpha_i$ and $\mu_j > \mu_i$ for all $i \in \{1,\ldots,K\}$, i.e., $\alpha_j + V(\mathbf{q}-\mathbf{e}_j;P_j) \geq \alpha_i + V(\mathbf{q}-\mathbf{e}_i;P_i)$ for all jobs with $r_j \geq r_i$. Therefore, in the rest of the proof, we only consider type $i \in \{1,\ldots,K\}$ jobs where $r_j < r_i$. Let $\mathcal{C} = \{i : i = 1,\ldots,K; r_j < r_i\}$. Note that $\mathcal{C} \subset \{1,\ldots,K\}$ and $j \notin \mathcal{C}$. As $r_i > r_j \geq \mu_j > \mu_i$, we get $r_i - \mu_i > r_j - \mu_j \geq 0$ for all $i \in \mathcal{C}$. Hence, Conditions (4.2.4) and (4.2.5) are satisfied for all $i \in \mathcal{C}$ and $q_i, q_j \geq 1$. Next, for $i \in \mathcal{C}$, we rewrite Condition (4.2.3) as

$$
\begin{aligned}
&\sum_{k=1}^{K} q_kr_k[\alpha_jr_j-\alpha_ir_i+\alpha_k(r_i-\mu_i-r_j+\mu_j)] - \alpha_jr_j(r_i-\mu_i)+\alpha_ir_i(r_j-\mu_j) \geq 0 \\
&\Leftrightarrow \sum_{k=1,k\neq i,j}^{K} q_kr_k[\alpha_jr_j-\alpha_ir_i+\alpha_k(r_i-\mu_i-r_j+\mu_j)] + (q_i-1)r_i[r_j(\alpha_j-\alpha_i)+\alpha_i(\mu_j-\mu_i)] \\
&+(q_j-1)r_j[r_i(\alpha_j-\alpha_i)+\alpha_j(\mu_j-\mu_i)] + r_ir_j(\alpha_j-\alpha_i)+\alpha_jr_j\mu_j-\alpha_ir_i\mu_i \geq 0. \qquad \text{(A.0.14)}
\end{aligned}
$$

Since we have $\alpha_i \leq \alpha_j$, $\mu_i < \mu_j$, $\alpha_ir_i \leq \alpha_jr_j$ (and hence $\alpha_ir_i\mu_i \leq \alpha_jr_j\mu_j$), $r_i-\mu_i > r_j-\mu_j$ for all $i \in \mathcal{C}$, and $q_i, q_j \geq 1$, Condition (A.0.14) is satisfied for all $i \in \mathcal{C}$. Thus, we have shown that under the given conditions in Proposition 4.2.2, Conditions (4.2.3), (4.2.4), (4.2.5), and (4.2.6) are satisfied for all $i \in \mathcal{C}$ such that $q_i \geq 1$.

We will now apply induction on $\sum_{i=1}^{K} q_i$. First, consider the case where $\sum_{i=1}^{K} q_i = 1$ such that $q_j \geq 1$, i.e., $\mathbf{q} = \mathbf{e}_j$. In this case, the result holds trivially. Now, suppose that the result is true for all feasible $\mathbf{q}$ such that $\sum_{i=1}^{K} q_i = a$, for some integer $a \geq 1$ and $q_j \geq 1$. Then, for any $\tilde{\mathbf{q}} = (\tilde{q_1},\ldots,\tilde{q_K})$

such that $\sum_{i=1}^{K} \tilde{q}_i = a+1$ and $\tilde{q}_j \geq 1$, we have $V(\tilde{\mathbf{q}}; R) = \alpha_j + V(\tilde{\mathbf{q}} - \mathbf{e}_j; P_j)$ by Proposition 4.2.1 since Conditions (4.2.3), (4.2.4), (4.2.5), and (4.2.6) are satisfied for all $i \in \mathcal{C}$ such that $\tilde{q}_i \geq 1$. This shows that for all $i \in \mathcal{C}$ where $q_j \geq 1$, job j should receive the highest priority. □

Proof of Proposition 4.2.3: First, note that if the optimal policy is an index policy, then it is sufficient to show that a type i job will be served under the optimal policy at state $(\mathbf{e}_i + \mathbf{e}_j; R)$ if and only if the required condition holds.

(i) Given that $\alpha_i = \alpha_j = \alpha$ for all $i, j \in \{1, \ldots, K\}$, using Equations (4.2.1) and (4.2.2) multiple times, we obtain

$$V(\mathbf{e}_j; P_i) - V(\mathbf{e}_i; P_j) = \frac{\mu_i \alpha}{\mu_i + r_j} - \frac{\mu_j \alpha}{\mu_j + r_i} = \frac{\alpha(r_i\mu_i - r_j\mu_j)}{(\mu_i + r_j)(\mu_j + r_i)}, \quad \forall\, i, j = 1, \ldots, K,$$

Hence, $V(\mathbf{e}_j; P_i) \geq V(\mathbf{e}_i; P_j)$ if and only if $r_i\mu_i \geq r_j\mu_j$, for $i, j \in \{1, \ldots, K\}$, given that $\alpha_i = \alpha_j = \alpha$, which completes the proof for this case.

(ii) Given that $r_i = r_j = r$ for all $i, j \in \{1, \ldots, K\}$, using Equations (4.2.1) and (4.2.2) multiple times, we obtain

$$\alpha_i + V(\mathbf{e}_j; P_i) - \alpha_j - V(\mathbf{e}_i; P_j) = \alpha_i + \frac{\mu_i \alpha_j}{\mu_i + r} - \alpha_j - \frac{\mu_j \alpha_i}{\mu_j + r} = \frac{\alpha_i r_i(\mu_i - r) - \alpha_j(\mu_j + r)}{(\mu_i + r)(\mu_j + r)}.$$

Hence, $\alpha_i + V(\mathbf{e}_j; P_i) \geq \alpha_j + V(\mathbf{e}_i; P_j)$ if and only if $\alpha_i(\mu_i + r) \geq \alpha_j(\mu_j + r)$, for $i, j \in \{1, \ldots, K\}$, given that $r_i = r_j = r$, which completes the proof. □

Proof of Proposition 4.3.1: First, note that when $r_1 \leq r_2$, using Proposition 4.1.1 together with the conditions $\alpha_1 \leq \alpha_2$ and $\mu_1 < \mu_2$ we conclude that serving type 2 jobs is optimal at all decision epochs. This is consistent with Proposition 4.3.1, as $t(q_1) = 0$ for $r_1 \leq r_2$. Therefore, we focus on the case where $r_1 > r_2$ in the rest of the proof.

(i) Let $K = 2$, $i = 2$, and $j = 1$ in Proposition 4.2.1. Then, given $r_1 > r_2$, $\alpha_1 \leq \alpha_2$, and $\mu_1 < \mu_2$, Condition (4.2.3) diminishes to $q_2 \leq t(q_1)$. Moreover, given $\mu_1 < \mu_2$, we rewrite Condition (4.2.4) as

$$r_1(r_2 - \mu_2)q_1 + r_2(r_1 - \mu_1)q_2 \leq (r_1 - \mu_1)(r_2 - \mu_2). \tag{A.0.15}$$

Note that, for $r_2 < r_1 \leq \mu_1 < \mu_2$, Condition (A.0.15), and hence Condition (4.2.4), is satisfied for all

$q_1, q_2 \geq 0$.

We next use induction on q_1 to prove the result. For $q_1 = q_2 = 1$, the condition that $q_2 \leq t(q_1)$ diminishes to $\alpha_1 r_1(\mu_1 + r_2) \geq \alpha_2 r_2(\mu_2 + r_1)$, which is the necessary and sufficient condition for the optimality of serving a type 1 job at state $(1, 1; R)$; see Equations (4.2.1) and (4.2.2). Then, by Proposition 4.2.1, it is optimal to serve a type 1 job at states $(1, q_2; R)$ such that $q_2 \leq t(1)$. Now, suppose that for $q_1 = a$, where $a \geq 1$ is an integer, it is optimal to serve type 1 jobs at states $(a, q_2; R)$ such that $q_2 \leq t(a)$. Then, applying Proposition 4.2.1, we conclude that it is optimal to serve type 1 jobs at states $(a + 1, q_2; R)$ such that $q_2 \leq t(a + 1)$ since $t(q_1)$ is non-increasing in q_1.

(ii) Let $K = 2$, $i = 1$, and $j = 2$ in Proposition 4.2.1. Similar to part (i), given $r_1 > r_2$, $\alpha_1 \leq \alpha_2$, and $\mu_1 < \mu_2$, Condition (4.2.3) diminishes to $q_2 \geq t(q_1)$. Moreover, given $\mu_1 < \mu_2$, we rewrite Condition (4.2.4) as

$$r_1(r_2 - \mu_2)(q_1 - 1) + r_2(r_1 - \mu_1)(q_2 - 1) \geq \mu_1\mu_2 - r_1 r_2. \tag{A.0.16}$$

Note that for $r_1 > r_2 \geq \mu_2 > \mu_1$, Condition (A.0.16), and hence Condition (4.2.4), is satisfied for all $q_1, q_2 \geq 1$.

We next use induction on q_1 to prove the result. We start with the case where $q_1 = 1$. Proposition 4.2.1 implies that if there is a state $(1, b; R)$, where $b \geq t(1)$ and serving a type 2 job is optimal, then it is also optimal to serve type 2 jobs in all states $(1, q_2; R)$ such that $q_2 \geq b$. Next, suppose that serving a type 2 job is optimal in states $(a, q_2; R)$ for all $q_2 \geq \tilde{t}(a) \geq t(a)$, where $a \geq 1$ is an integer. If there exists a state $(a + 1, d)$ where $d \geq \tilde{t}(a) - 1$ and serving a type 2 job is optimal, then by Proposition 4.2.1, it is also optimal to serve type 2 jobs in all states $(a + 1, q_2; R)$ such that $q_2 \geq d$. (Note that $t(q_1)$ in non-increasing in q_1, hence Condition (4.2.3) is satisfied for $(a + 1, d + 1; R)$ as $d + 1 \geq \tilde{t}(a) \geq t(a) \geq t(a + 1)$.) This completes the proof. □

Proof of Proposition 4.3.2: Let $K = 2$, $i = 1$, and $j = 2$ in Proposition 4.2.1. Then, for $q_1 = q_2 = 1$, Condition (4.2.3) is satisfied as it diminishes to $\alpha_1 r_1(\mu_1 + r_2) \leq \alpha_2 r_2(\mu_2 + r_1)$. Next, we rewrite Condition (4.2.3) as

$$q_1 r_1[(\alpha_2 - \alpha_1)r_2 + \alpha_1(\mu_2 - \mu_1)] + q_2 r_2[(\alpha_2 - \alpha_1)r_1 + \alpha_2(\mu_2 - \mu_1)] \geq \alpha_2 r_2(r_1 - \mu_1) - \alpha_1 r_1(r_2 - \mu_2). \tag{A.0.17}$$

Note that, for $(\alpha_2 - \alpha_1)r_2 \geq \alpha_1(\mu_1 - \mu_2)$ and $(\alpha_2 - \alpha_1)r_1 \geq \alpha_2(\mu_1 - \mu_2)$, the left-hand side of Condition (A.0.17) is non-decreasing in q_1 and q_2, and hence Condition (4.2.3) is satisfied for all $q_1, q_2 \geq 1$. Moreover, given $\mu_1 < \mu_2$, we rewrite Condition (4.2.4) as Condition (A.0.16), which is satisfied for all $q_1, q_2 \geq 1$ because $r_1 \geq \mu_1$ and $r_2 \geq \mu_2$. Finally, the necessary and sufficient condition for the optimality of serving a type 2 job at state $(1, 1; R)$ is satisfied as it diminishes to $\alpha_1 r_1(\mu_1 + r_2) \leq \alpha_2 r_2(\mu_2 + r_1)$; see Equations (4.2.1) and (4.2.2). Then, by Proposition 4.2.1, it is optimal to serve a type 2 job at states $(q_1, 1; R)$ for all $q_1 \geq 1$. Furthermore, since it is also optimal to serve a type 2 job at states $(0, q_2; R)$ for all $q_2 \geq 1$, applying Proposition 4.2.1 multiple times completes the proof. □

Proof of Corollary 4.3.1: Note that when $r_1 \leq r_2$, using Proposition 4.1.1 together with the conditions $\alpha_1 \leq \alpha_2$ and $\mu_1 < \mu_2$ we conclude that serving type 2 jobs is optimal at all decision epochs. Hence, we focus on the case where $r_1 > r_2$. Now, since we have $r_1 > r_2 \geq \mu_2 > \mu_1$, $\alpha_1 \leq \alpha_2$, and $\alpha_1 r_1 \mu_1 \leq \alpha_2 r_2 \mu_2$, the conditions required in Proposition 4.3.2 are all satisfied, which completes the proof. □

Proof of Proposition 4.3.3: Similar to the proof of Proposition 4.2.3, as the optimal policy is an index policy, it is sufficient to show that a type i job will be served under the optimal policy at state $(1, 1; R)$ if and only if the required condition holds. Note that using Equations (4.2.1) and (4.2.2) multiple times, we obtain

$$\begin{aligned}\alpha_1 + V(0,1;P_1) - \alpha_2 + V(1,0;P_2) &= \alpha_1 + \frac{\mu_1\alpha_2}{\mu_1 + r_2} - \alpha_2 - \frac{\mu_2\alpha_1}{\mu_2 + r_1} \\ &= \frac{\alpha_1 r_1(\mu_1 + r_2) - \alpha_2 r_2(\mu_2 + r_1)}{(\mu_1 + r_2)(\mu_2 + r_1)}.\end{aligned}$$

Hence, $\alpha_1 + V(0,1;P_1) \geq \alpha_2 + V(1,0;P_2)$ if and only if $\alpha_1 r_1(\mu_1 + r_2) \geq \alpha_2 r_2(\mu_2 + r_1)$, which completes the proof. □

In order to prove Propositions 5.1.1 and 5.1.2, we consider the following finite horizon problem:

$$\max_{\pi\in\Pi} \Gamma_\pi(T) \quad \text{for fixed} \quad T < \infty, \tag{A.0.18}$$

Our discrete-time MDP is again defined by the state space $\mathcal{S}$, action space $\mathcal{A}$, a set of known transition probabilities, and a reward function. Then, the total reward for this finite horizon problem is

$$r_N(s) + \sum_{n=0}^{N-1} r_n(s, d_n(s)),$$

where $r_N(s)$ is a terminal reward incurred at the end of the planning horizon which we set to zero.

Let $J_{\pi,N}(s_0)$ denote the expected average reward over the N-period decision making horizon if policy π is used and the system is in state s_0 at time zero. Then, for $\pi \in \Pi$, we have

$$J_{\pi,N}(s_0) = \frac{1}{N}\mathbb{E}\left[r_N(Y_\pi(N)) + \sum_{n=0}^{N-1} r_n(Y_\pi(n), d_n(Y_\pi(n)))\right], \forall s_0 \in \mathcal{S}.$$

The optimal reward is given by

$$J_N(s_0) = \max_{\pi \in \Pi} J_{\pi,N}(s_0).$$

Let $v_n(s, a_i)$ represent the average reward over the periods ranging from n to N, where system is in state s and action a_i is chosen in period n, and the optimal action is chosen in periods $n+1$ to N. Let also $v_n(s)$ be the optimal average reward from period n to N, when the system is in state s in period n. Then, for all $s \in \mathcal{S}$ and $n = 0, \dots, N-1$, we have

$$v_n(s) = \max_{a_i \in \mathcal{A}(s)} \{v_n(s, a_i)\}.$$

In order to prove Proposition 5.1.1, we need the following lemma.

Lemma A.0.3. *For all $k = 1, \dots, K$, $v(s) \geq v(s - e_k)$, where $s \in \mathcal{S}$ and $s_k \geq 1$.*

Proof of Lemma A.0.3: Fix $k \in \{1, \dots, K\}$. We first consider the finite horizon problem for which we will prove that $v_n(s) \geq v_n(s - e_k)$ for all periods $n = 0, \dots, N$ and all states $s \in \mathcal{S}$, where $s_k \geq 1$. To prove this result, we will consider two sample paths.

In the first sample path, suppose that the state is $s - e_k$, where $s \in \mathcal{S}$ and $s_k \geq 1$ in period n, where $n = 0, \dots, N$. Suppose that this sample path is governed by an optimal policy, which we call policy π. In the second sample path, suppose that the state is s in period n, where $n = 0, \dots, N$. We will next construct a policy, which we call policy π_0, and apply this policy in the second sample path. Then, using induction on n, we will show that $v_n(s - e_k) \leq v_n^{(0)}(s)$, where $v_n^{(0)}(s)$ is the value function under policy π_0. Since π_0 is not necessarily an optimal policy, this will imply that $v_n(s - e_k) \leq v_n(s)$ for all

$n \in \{0, \ldots, N\}$ and $s \in \mathcal{S}$.

First consider period N. In this case, we have $v_N^{(0)}(s) = v_N(s) = 0$ for all $s \in \mathcal{S}$ and hence the result holds trivially. Next, suppose that $v_{n+1}(s - e_k) \leq v_{n+1}^{(0)}(s)$ for some period $n + 1$, where $n \in \{0, \ldots, N - 1\}$, and all $s \in \mathcal{S}$, where $s_k \geq 1$, if π_0 does the same action in period $n + 1$ (with state s) that π takes in period $n + 1$ (with state $s - e_k$). We will show that this also holds for n.

Case 1 ($s_k < C_k$): At period n, the probability of the next event being an arrival, a service completion, or an abandonment for a type i job, where $i \neq k$, is the same for both sample paths. On the other hand, the probability of next event being the abandonment of a type k job is larger in the second sample path, which means that the probability of staying in the same state (due to uniformization) is smaller in the second sample path. Hence, when we can couple both sample paths, either both sample paths reach the same state in period $n + 1$, i.e., $s - e_k$, or the state under the first sample path is $s' - e_k$ in period $n + 1$ whereas it is s' under the second sample path, where $s' \in \mathcal{S}$. In the first situation, π_0 follows π exactly. This means that $v_{n+1}(s) = v_{n+1}^{(0)}(s)$ for all $s \in \mathcal{S}$, which implies that $v_n(s - e_k) \leq v_n^{(0)}(s)$. In the second situation, π_0 takes the same action under the second sample path that π takes under the first sample path. Hence, by the inductive hypothesis, we have $v_n(s - e_k) \leq v_n^{(0)}(s)$ for all $s \in \mathcal{S}$ and $s_k \geq 1$.

Case 2 ($s_k = C_k$): At period n, the probabilities of next events are the same as in Case 1. Hence, the results for Case 1 also apply here. The only difference is that, if the next event is the arrival of a type k job, then that job is lost under the second sample path, whereas it will be admitted in the first sample path. Hence, both sample paths will reach the same state in period $n + 1$, i.e., s. In this case, π_0 follows π exactly, which implies that $v_{n+1}(s) = v_{n+1}^{(0)}(s)$ for all $s \in \mathcal{S}$, and hence $v_n(s - e_k) \leq v_n^{(0)}(s)$.

Above we have proved that $v_n(s - e_k) \leq v_n^{(0)}(s) \leq v_n(s)$ for all $n = 0, \ldots, N$, $s \in \mathcal{S}$, and $s_k \geq 1$. Letting $N \to \infty$, we get $v(s - e_k) \leq v(s)$, which completes the proof. □

Proof of Proposition 5.1.1: To prove the result, we show that $M(s, a_i) \geq M(s, a_j)$ for all $j \in \{1, \ldots, K\} \setminus \{i\}$. $j \in \{1, \ldots, K\} \setminus \{i\}$, we have

$$\begin{aligned} M(s, a_i) - M(s, a_j) &= R_i\mu_i - R_j\mu_j + (\gamma_i - \mu_i)\Big[v(s) - v(s - e_i)\Big] - (\gamma_j - \mu_j)\Big[v(s) - v(s - e_j)\Big] \\ &\geq R_i\mu_i - R_j\mu_j \\ &\geq 0, \end{aligned}$$

where the first inequality holds from Lemma A.0.3 and the conditions that $\gamma_i \geq \mu_i$ and $\gamma_j \leq \mu_j$ $j \in \{1, \ldots, K\} \setminus \{i\}$, and the second inequality holds because $R_i\mu_i \geq R_j\mu_j$. This completes the proof. □

In order to prove Proposition 5.1.2, we consider a capacity restriction on the total number of jobs, and we let $C < \infty$ denote that capacity. Then, the state space $\mathcal{S}$ is given by

$$\mathcal{S} = \left\{ s = (s_1, \cdots, s_K) : s_i \in \{0, 1, \ldots, C\} \text{ for } i = 1, \ldots, K, \ \sum_{j=1}^{K} s_j \leq C \right\},$$

where s_i is the number of type i jobs in the system. Furthermore, the uniformization constant is now given by

$$\psi = \sum_{i=1}^{K} \lambda_i + \max_{i=1,\ldots,K} \mu_i + (C-1) \max_{i=1,\ldots,K} \gamma_i.$$

Moreover, let $C(s)$ denote the total number of jobs in the system when it is in state $s = (s_1, \ldots, s_K) \in \mathcal{S}$, i.e., $C(s) = \sum_{i=1}^{K} s_i$, and $\mathbf{1}_{\{C(s)<C\}}$ be the indicator function of the set $\{C(s) < C\}$. Then, considering the finite horizon problem, for all $a_i \in \mathcal{A}(s)$, $s \in \mathcal{S}$, $i = 1, \ldots, K$ and $n = 0, \ldots, N-1$, we have:

$$\begin{aligned} v_n(s, a_i) = \; & R_i\mu_i + \mathbf{1}_{\{C(s)<C\}} \sum_{j=1}^{K} \lambda_j v_{n+1}(s + e_j) + \sum_{j=1}^{K} s_j\gamma_j v_{n+1}(s - e_j) \\ & + (\mu_i - \gamma_i) v_{n+1}(s - e_i) \\ & + \left[1 - \mathbf{1}_{\{C(s)<C\}} \sum_{j=1}^{K} \lambda_j - \sum_{j=1}^{K} s_j\gamma_j - (\mu_i - \gamma_i) \right] v_{n+1}(s). \end{aligned}$$

Then, for $s \in \mathcal{S}$, the Bellman's average reward optimality equations are given by

$$\begin{aligned} & v_N(s) = 0, \\ & v_n(s) = \; \mathbf{1}_{\{C(s)<C\}} \sum_{j=1}^{K} \lambda_j v_{n+1}(s + e_j) + \sum_{j=1}^{K} s_j\gamma_j v_{n+1}(s - e_j) \\ & \qquad + \left[1 - \mathbf{1}_{\{C(s)<C\}} \sum_{j=1}^{K} \lambda_j - \sum_{j=1}^{K} s_j\gamma_j \right] v_{n+1}(s) + \max_{a_i \in \mathcal{A}(s)} \{ M_{n+1}(s, a_i) \}, \end{aligned}$$

for $n = 0, \ldots, N-1$, where

$$M_{n+1}(s, a_i) = R_i\mu_i + (\gamma_i - \mu_i)\Bigg[v_{n+1}(s) - v_{n+1}(s - e_i)\Bigg], \quad \text{for } i = 1, \ldots, K.$$

Next, for the $K = 2$ case, before we introduce new notation, we redefine the period n as the number of periods left to reach N for simplicity of notation. For $n = 1, \ldots, N$ and $s_1, s_2 \geq 1$ where $(s_1, s_2) \in \mathcal{S}$, we let $M_n(s_1, s_2) = \max_{i=1,2}\{M_n(s_1, s_2; a_i)\}$. Then, we have

$$\begin{aligned} M_n(s_1, s_2) = \max\Bigg\{ \quad & R_1\mu_1 \quad +(\gamma_1 - \mu_1)\left[v_n(s_1, s_2) - v_n(s_1 - 1, s_2)\right], \\ & R_2\mu_2 \quad +(\gamma_2 - \mu_2)\left[v_n(s_1, s_2) - v_n(s_1, s_2 - 1)\right]\Bigg\}. \end{aligned}$$

Next, we let $\Delta_n^{(1)}(s_1, s_2) = v_n(s_1, s_2) - v_n(s_1 - 1, s_2)$ and $\Delta_n^{(2)}(s_1, s_2) = v_n(s_1, s_2) - v_n(s_1, s_2 - 1)$. Then, $\forall (s_1, s_2) \in \mathcal{S}$ and $n = 1, \ldots, N$, we get

$$M_n(s_1, s_2) = \max\Bigg\{R_1\mu_1 + (\gamma_1 - \mu_1)\Delta_n^{(1)}(s_1, s_2), R_2\mu_2 + (\gamma_2 - \mu_2)\Delta_n^{(2)}(s_1, s_2)\Bigg\}.$$

Next, for $s_1 \geq 1$, we have

$$\begin{aligned} \Delta_n^{(1)}(s_1, s_2) \quad &= \quad \mathbf{1}_{\{s_1+s_2<C\}}\left[\lambda_1\Delta_{n-1}^{(1)}(s_1 + 1, s_2) + \lambda_2\Delta_{n-1}^{(1)}(s_1, s_2 + 1)\right] \\ &+ \quad (s_1 - 1)\gamma_1\Delta_{n-1}^{(1)}(s_1 - 1, s_2) + s_2\gamma_2\Delta_{n-1}^{(1)}(s_1, s_2 - 1) \\ &+ \quad \left[1 - \mathbf{1}_{\{s_1+s_2-1<C\}}(\lambda_1 + \lambda_2) - (s_1\gamma_1 + s_2\gamma_2)\right]\Delta_{n-1}^{(1)}(s_1, s_2) \\ &+ \quad \mathbf{1}_{\{s_2>0\}}\Bigg(M_{n-1}(s_1, s_2) - \mathbf{1}_{\{s_1>1\}}M_{n-1}(s_1 - 1, s_2) \\ &- \quad \mathbf{1}_{\{s_1=1\}}\left[R_2\mu_2 + (\gamma_2 - \mu_2)\Delta_{n-1}^{(2)}(s_1 - 1, s_2)\right]\Bigg) \\ &+ \quad \mathbf{1}_{\{s_2=0\}}\Bigg(\left[R_1\mu_1 + (\gamma_1 - \mu_1)\Delta_{n-1}^{(1)}(s_1, s_2)\right] \\ &- \quad \mathbf{1}_{\{s_1>1\}}\left[R_1\mu_1 + (\gamma_1 - \mu_1)\Delta_{n-1}^{(1)}(s_1 - 1, s_2)\right]\Bigg) \\ &+ \quad \mathbf{1}_{\{s_1+s_2=C\}}\lambda_2\left[v_{n-1}(s_1, s_2) - v_{n-1}(s_1 - 1, s_2 + 1)\right]. \end{aligned}$$

Similarly, for $s_2 \geq 1$, we have

$$\begin{aligned}
\Delta_n^{(2)}(s_1,s_2) &= \mathbf{1}_{\{s_1+s_2<C\}}\left[\lambda_1\Delta_{n-1}^{(2)}(s_1+1,s_2)+\lambda_2\Delta_{n-1}^{(2)}(s_1,s_2+1)\right] \\
&+ s_1\gamma_1\Delta_{n-1}^{(2)}(s_1-1,s_2)+(s_2-1)\gamma_2\Delta_{n-1}^{(2)}(s_1,s_2-1) \\
&+ \left[1-\mathbf{1}_{\{s_1+s_2-1<C\}}(\lambda_1+\lambda_2)-(s_1\gamma_1+s_2\gamma_2)\right]\Delta_{n-1}^{(2)}(s_1,s_2) \\
&+ \mathbf{1}_{\{s_1>0\}}\Bigg(M_{n-1}(s_1,s_2)-\mathbf{1}_{\{s_2>1\}}M_{n-1}(s_1,s_2-1) \\
&- \mathbf{1}_{\{s_2=1\}}\left[R_1\mu_1+(\gamma_1-\mu_1)\Delta_{n-1}^{(1)}(s_1,s_2-1)\right]\Bigg) \\
&+ \mathbf{1}_{\{s_1=0\}}\Bigg(\left[R_2\mu_2+(\gamma_2-\mu_2)\Delta_{n-1}^{(2)}(s_1,s_2)\right] \\
&- \mathbf{1}_{\{s_2>1\}}\left[R_2\mu_2+(\gamma_2-\mu_2)\Delta_{n-1}^{(2)}(s_1,s_2-1)\right]\Bigg) \\
&+ \mathbf{1}_{\{s_1+s_2=C\}}\lambda_1\left[v_{n-1}(s_1,s_2)-v_{n-1}(s_1+1,s_2-1)\right].
\end{aligned}$$

Let $\Delta_n(s_1,s_2)=\Delta_n^{(1)}(s_1,s_2)-\Delta_n^{(2)}(s_1,s_2)=v_n(s_1,s_2-1)-v_n(s_1-1,s_2)$ for $s_1\geq 1$ and $s_2\geq 1$. Then, for $s_1,s_2\geq 1$, we have

$$\begin{aligned}
\Delta_n(s_1,s_2) &= \mathbf{1}_{\{s_1+s_2-1<C\}}\left[\lambda_1\Delta_{n-1}(s_1+1,s_2)+\lambda_2\Delta_{n-1}(s_1,s_2+1)\right] \\
&+ (s_1-1)\gamma_1\Delta_{n-1}(s_1-1,s_2)+(s_2-1)\gamma_2\Delta_{n-1}(s_1,s_2-1) \\
&+ \left[1-\mathbf{1}_{\{s_1+s_2-1<C\}}(\lambda_1+\lambda_2)-(s_1\gamma_1+s_2\gamma_2)\right]\Delta_{n-1}(s_1,s_2) \\
&- \gamma_1\Delta_{n-1}^{(2)}(s_1-1,s_2)+\gamma_2\Delta_{n-1}^{(1)}(s_1,s_2-1) \\
&+ \mathbf{1}_{\{s_2>1\}}M_{n-1}(s_1,s_2-1)+\mathbf{1}_{\{s_2=1\}}\left[R_1\mu_1+(\gamma_1-\mu_1)\Delta_{n-1}^{(1)}(s_1,s_2-1)\right] \\
&- \mathbf{1}_{\{s_1>1\}}M_{n-1}(s_1-1,s_2)-\mathbf{1}_{\{s_1=1\}}\left[R_2\mu_2+(\gamma_2-\mu_2)\Delta_{n-1}^{(2)}(s_1-1,s_2)\right].
\end{aligned} \tag{A.0.19}$$

Finally, we need the following lemma to prove Proposition 5.1.2.

Lemma A.0.4. *If $R_1\mu_1=R_2\mu_2$, $\gamma_1\leq\mu_1$, and $\gamma_1-\gamma_2\geq\mu_1-\mu_2\geq 0$, then, for all $n\geq 0$, $\Delta_n^{(2)}(s_1,s_2)\geq 0$ for $s_2\geq 1$, and $M_n(s_1,s_2)=R_1\mu_1+(\gamma_1-\mu_1)\Delta_n^{(1)}(s_1,s_2)$ and $\Delta_n(s_1,s_2)\leq 0$ for $s_1\geq 1$ and $s_2\geq 1$.*

Proof of Lemma A.0.4: The proof is by induction on n. By definition $v_0(s)=0$, $\forall s\in S$. Thus,

$\Delta_0^{(1)}(s_1,s_2) = \Delta_0^{(2)}(s_1,s_2) = 0$ and $M_0(s) = R_1\mu_1 = R_2\mu_2$. Then, by the definition of $\Delta_n^{(2)}(s_1,s_2)$, $\Delta_n(s_1,s_2)$ and $M_n(s_1,s_2)$, and as $\mathbf{1}_{\{s_1>0\}} + \mathbf{1}_{\{s_1=0\}} = 1$ for $s_1 \geq 0$, $\mathbf{1}_{\{s_1>1\}} + \mathbf{1}_{\{s_1=1\}} = 1$ for $s_1 \geq 1$, $\mathbf{1}_{\{s_2>1\}} + \mathbf{1}_{\{s_2=1\}} = 1$ for $s_2 \geq 1$, we have $\Delta_1^{(1)}(s_1,s_2) = \Delta_1^{(2)}(s_1,s_2) = 0$, $M_n(s_1,s_2) = R_1\mu_1 = R_2\mu_2$, and $\Delta_1(s_1,s_2) = 0$ for $s_1 \geq 1$ and $s_2 \geq 1$ and $\Delta_1^{(2)}(s_1,s_2) = \mathbf{1}_{\{s_1=0,s_2=1\}}R_2\mu_2 \geq 0$ for $s_2 \geq 1$. Thus, the result holds for $n = 1$. Next suppose that the result holds for $n-1$. We will show that it also holds for n. Note that we have $M_{n-1}(s_1,s_2) = R_1\mu_1 + (\gamma_1 - \mu_1)\Delta_{n-1}^{(1)}(s_1,s_2)$ from the induction hypothesis. We first prove that for $\Delta_n^{(2)}(s_1,s_2) \geq 0$ for $s_2 \geq 1$. The following two cases exhaust all possibilities:

<u>Case 1: $s_1 \geq 1$</u>

$$\begin{aligned}
\Delta_n^{(2)}(s_1,s_2) &= \mathbf{1}_{\{s_1+s_2<C\}}\left[\lambda_1\Delta_{n-1}^{(2)}(s_1+1,s_2) + \lambda_2\Delta_{n-1}^{(2)}(s_1,s_2+1)\right] \\
&+ (s_1-1)\gamma_1\Delta_{n-1}^{(2)}(s_1-1,s_2) + \gamma_1\Delta_{n-1}^{(2)}(s_1-1,s_2) + (s_2-1)\gamma_2\Delta_{n-1}^{(2)}(s_1,s_2-1) \\
&+ \left[1 - \mathbf{1}_{\{s_1+s_2-1<C\}}(\lambda_1+\lambda_2) - (s_1\gamma_1 + s_2\gamma_2)\right]\Delta_{n-1}^{(2)}(s_1,s_2) \\
&+ \left[R_1\mu_1 + (\gamma_1-\mu_1)\Delta_{n-1}^{(1)}(s_1,s_2)\right] - \left[R_1\mu_1 + (\gamma_1-\mu_1)\Delta_{n-1}^{(1)}(s_1,s_2-1)\right] \\
&+ \mathbf{1}_{\{s_1+s_2=C\}}\lambda_1\left[v_{n-1}(s_1,s_2) - v_{n-1}(s_1+1,s_2-1)\right] \\
&= \mathbf{1}_{\{s_1+s_2<C\}}\left[\lambda_1\Delta_{n-1}^{(2)}(s_1+1,s_2) + \lambda_2\Delta_{n-1}^{(2)}(s_1,s_2+1)\right] \\
&+ (s_1-1)\gamma_1\Delta_{n-1}^{(2)}(s_1-1,s_2) + (s_2-1)\gamma_2\Delta_{n-1}^{(2)}(s_1,s_2-1) \\
&+ \left[1 - \mathbf{1}_{\{s_1+s_2-1<C\}}(\lambda_1+\lambda_2) - (s_1-1)\gamma_1 - s_2\gamma_2 - \mu_1\right]\Delta_{n-1}^{(2)}(s_1,s_2) \\
&+ \gamma_1\left[\Delta_{n-1}^{(2)}(s_1-1,s_2) + \Delta_{n-1}^{(1)}(s_1,s_2) - \Delta_{n-1}^{(1)}(s_1,s_2-1) - \Delta_{n-1}^{(2)}(s_1,s_2)\right] \\
&+ \mu_1\left[\Delta_{n-1}^{(1)}(s_1,s_2-1) - \Delta_{n-1}^{(1)}(s_1,s_2) + \Delta_{n-1}^{(2)}(s_1,s_2)\right] \\
&- \mathbf{1}_{\{s_1+s_2=C\}}\lambda_1\Delta_{n-1}(s_1+1,s_2) \\
&= \mathbf{1}_{\{s_1+s_2<C\}}\left[\lambda_1\Delta_{n-1}^{(2)}(s_1+1,s_2) + \lambda_2\Delta_{n-1}^{(2)}(s_1,s_2+1)\right] \\
&+ (s_1-1)\gamma_1\Delta_{n-1}^{(2)}(s_1-1,s_2) + (s_2-1)\gamma_2\Delta_{n-1}^{(2)}(s_1,s_2-1) \\
&+ \left[1 - \mathbf{1}_{\{s_1+s_2-1<C\}}(\lambda_1+\lambda_2) - (s_1-1)\gamma_1 - s_2\gamma_2 - \mu_1\right]\Delta_{n-1}^{(2)}(s_1,s_2) \\
&+ \mu_1\Delta_{n-1}^{(2)}(s_1-1,s_2) - \mathbf{1}_{\{s_1+s_2=C\}}\lambda_1\Delta_{n-1}(s_1+1,s_2) \geq 0,
\end{aligned}$$

where the inequality holds because $\Delta_{n-1}^{(2)}(s_1,s_2) \geq 0$ and $\Delta_{n-1}(s_1+1,s_2) \leq 0$ for $s_1 \geq 0$ and $s_2 \geq 1$ and all coefficients of $\Delta_{n-1}^{(2)}(s_1,s_2)$ are non-negative.

Case 2: $s_1 = 0$

$$\begin{aligned}
\Delta_n^{(2)}(0,s_2) &= \mathbf{1}_{\{s_2<C\}}\left[\lambda_1\Delta_{n-1}^{(2)}(1,s_2)+\lambda_2\Delta_{n-1}^{(2)}(0,s_2+1)\right]+(s_2-1)\gamma_2\Delta_{n-1}^{(2)}(0,s_2-1)\\
&+ \left[1-\mathbf{1}_{\{s_2-1<C\}}(\lambda_1+\lambda_2)-s_2\gamma_2\right]\Delta_{n-1}^{(2)}(0,s_2)\\
&+ R_2\mu_2+(\gamma_2-\mu_2)\Delta_{n-1}^{(2)}(0,s_2)-\mathbf{1}_{\{s_2>1\}}\left[R_2\mu_2+(\gamma_2-\mu_2)\Delta_{n-1}^{(2)}(0,s_2-1)\right]\\
&+ \mathbf{1}_{\{s_2=C\}}\lambda_1\left[v_{n-1}(0,s_2)-v_{n-1}(1,s_2-1)\right]\\
&= \mathbf{1}_{\{s_2<C\}}\left[\lambda_1\Delta_{n-1}^{(2)}(1,s_2)+\lambda_2\Delta_{n-1}^{(2)}(0,s_2+1)\right]\\
&+ \left[1-\mathbf{1}_{\{s_2-1<C\}}(\lambda_1+\lambda_2)-(s_2-1)\gamma_2-\mu_2\right]\Delta_{n-1}^{(2)}(0,s_2)\\
&+ \mathbf{1}_{\{s_2>1\}}\left[(s_2-2)\gamma_2\Delta_{n-1}^{(2)}(0,s_2-1)+\mu_2\Delta_{n-1}^{(2)}(0,s_2-1)\right]\\
&+ \mathbf{1}_{\{s_2=1\}}R_2\mu_2-\mathbf{1}_{\{s_2=C\}}\lambda_1\Delta_{n-1}(1,s_2)\geq 0,
\end{aligned}$$

where the inequality holds because $\Delta_{n-1}^{(2)}(0,s_2)\geq 0$ and $\Delta_{n-1}(1,s_2)\leq 0$ for $s_2\geq 1$ and all coefficients of $\Delta_{n-1}^{(2)}(0,s_2)$ are non-negative.

Now, we show that $\Delta_n(s_1,s_2)\leq 0$ for $s_1\geq 1$ and $s_2\geq 1$ for the following two cases:

Case 1: $s_1 > 1$ From Equation (A.0.19), we have

$$\begin{aligned}
\Delta_n(s_1,s_2) &= \mathbf{1}_{\{s_1+s_2-1<C\}}\left[\lambda_1\Delta_{n-1}(s_1+1,s_2)+\lambda_2\Delta_{n-1}(s_1,s_2+1)\right]\\
&+ (s_1-1)\gamma_1\Delta_{n-1}(s_1-1,s_2)+(s_2-1)\gamma_2\Delta_{n-1}(s_1,s_2-1)\\
&- \gamma_1\Delta_{n-1}^{(2)}(s_1-1,s_2)+\gamma_2\Delta_{n-1}^{(1)}(s_1,s_2-1)\\
&+ \left[1-\mathbf{1}_{\{s_1+s_2-1<C\}}(\lambda_1+\lambda_2)-(s_1\gamma_1+s_2\gamma_2)\right]\Delta_{n-1}(s_1,s_2)\\
&+ R_1\mu_1+(\gamma_1-\mu_1)\Delta_{n-1}^{(1)}(s_1,s_2-1)-\left[R_1\mu_1+(\gamma_1-\mu_1)\Delta_{n-1}^{(1)}(s_1-1,s_2)\right]\\
&= \mathbf{1}_{\{s_1+s_2-1<C\}}\left[\lambda_1\Delta_{n-1}(s_1+1,s_2)+\lambda_2\Delta_{n-1}(s_1,s_2+1)\right]+(s_1-2)\gamma_1\Delta_{n-1}(s_1-1,s_2)\\
&+ (s_2-1)\gamma_2\Delta_{n-1}(s_1,s_2-1)-\gamma_1\Delta_{n-1}^{(2)}(s_1-1,s_2)+\gamma_2\Delta_{n-1}^{(1)}(s_1,s_2-1)\\
&+ \left[1-\mathbf{1}_{\{s_1+s_2-1<C\}}(\lambda_1+\lambda_2)-(s_1\gamma_1+s_2\gamma_2)-\mu_1\right]\Delta_{n-1}(s_1,s_2)\\
&+ \gamma_1\left[\Delta_{n-1}^{(1)}(s_1,s_2-1)+\Delta_{n-1}(s_1-1,s_2)-\Delta_{n-1}^{(1)}(s_1-1,s_2)\right]\\
&+ \mu_1\left[\Delta_{n-1}(s_1,s_2)-\Delta_{n-1}^{(1)}(s_1,s_2-1)+\Delta_{n-1}^{(1)}(s_1-1,s_2)\right]\\
&= \mathbf{1}_{\{s_1+s_2-1<C\}}\left[\lambda_1\Delta_{n-1}(s_1+1,s_2)+\lambda_2\Delta_{n-1}(s_1,s_2+1)\right]\\
&+ \left[(s_1-2)\gamma_1+\mu_1\right]\Delta_{n-1}(s_1-1,s_2)+\gamma_1\Delta_{n-1}(s_1,s_2)
\end{aligned}$$

$$\begin{aligned}
&+ \quad (s_2-1)\gamma_2\Delta_{n-1}(s_1,s_2-1) + (\gamma_2-\gamma_1)\Delta^{(2)}_{n-1}(s_1-1,s_2) + \gamma_2\Delta_{n-1}(s_1,s_2) \\
&+ \quad \left[1-\mathbf{1}_{\{s_1+s_2-1<C\}}(\lambda_1+\lambda_2) - (s_1\gamma_1+s_2\gamma_2) - \mu_1\right]\Delta_{n-1}(s_1,s_2) \le 0,
\end{aligned}$$

where the inequality holds because $\Delta_{n-1}(s_1,s_2) \le 0$ and $\Delta^{(2)}_{n-1}(s_1-1,s_2) \ge 0$ for $s_1 \ge 1$ and $s_2 \ge 1$, $\gamma_2-\gamma_1 \le 0$, and all coefficients of $\Delta_{n-1}(s_1,s_2)$ are non-negative.

Case 2: $s_1 = 1$ From Equation (A.0.19), we have

$$\begin{aligned}
\Delta_n(1,s_2) \quad = \quad & \mathbf{1}_{\{s_2<C\}}\left[\lambda_1\Delta_{n-1}(2,s_2) + \lambda_2\Delta_{n-1}(1,s_2+1)\right] + (s_2-1)\gamma_2\Delta_{n-1}(1,s_2-1) \\
+ \quad & \left[1-\mathbf{1}_{\{s_2<C\}}(\lambda_1+\lambda_2) - (\gamma_1+s_2\gamma_2)\right]\Delta_{n-1}(1,s_2) - \gamma_1\Delta^{(2)}_{n-1}(0,s_2) + \gamma_2\Delta^{(1)}_{n-1}(1,s_2-1) \\
+ \quad & R_1\mu_1 + (\gamma_1-\mu_1)\Delta^{(1)}_{n-1}(1,s_2-1) - \left[R_2\mu_2 + (\gamma_2-\mu_2)\Delta^{(2)}_{n-1}(0,s_2)\right] \\
= \quad & \mathbf{1}_{\{s_2<C\}}\left[\lambda_1\Delta_{n-1}(2,s_2) + \lambda_2\Delta_{n-1}(1,s_2+1)\right] \\
+ \quad & (s_2-1)\gamma_2\Delta_{n-1}(1,s_2-1) + R_1\mu_1 - R_2\mu_2 + (\mu_2-\mu_1)\Delta^{(2)}_{n-1}(0,s_2) \\
+ \quad & \left[1-\mathbf{1}_{\{s_2<C\}}(\lambda_1+\lambda_2) - (s_2-1)\gamma_2 - \mu_1\right]\Delta_{n-1}(1,s_2) \le 0,
\end{aligned}$$

where the inequality holds because $\Delta_{n-1}(1,s_2) \le 0$ and $\Delta^{(2)}_{n-1}(0,s_2) \ge 0$ for $s_2 \ge 1$, $R_1\mu_1 = R_2\mu_2$, $\mu_2-\mu_1 \le 0$, and all coefficients of $\Delta_{n-1}(1,s_2)$ are non-negative.

Finally, using the definition of $M_n(s_1,s_2)$, we have

$$\begin{aligned}
M_n(s_1,s_2) \quad = \quad & R_2\mu_2 + (\gamma_2-\mu_2)\Delta^{(2)}_n(s_1,s_2) \\
+ \quad & \max\{0, R_1\mu_1 - R_2\mu_2 + (\gamma_1-\mu_1)\Delta^{(1)}_n(s_1,s_2) - (\gamma_2-\mu_2)\Delta^{(2)}_n(s_1,s_2)\}.
\end{aligned}$$

Note that

$$\begin{aligned}
& R_1\mu_1 - R_2\mu_2 + (\gamma_1-\mu_1)\Delta^{(1)}_n(s_1,s_2) - (\gamma_2-\mu_2)\Delta^{(2)}_n(s_1,s_2) \\
\ge \quad & (\gamma_1-\mu_1)\Delta^{(1)}_n(s_1,s_2) - (\gamma_1-\mu_1)\Delta^{(2)}_n(s_1,s_2) \\
= \quad & (\gamma_1-\mu_1)\Delta_n(s_1,s_2) \ge 0.
\end{aligned}$$

where the first inequality holds because $R_1\mu_1 = R_2\mu_2$, $0 \ge \gamma_1-\mu_1 \ge \gamma_2-\mu_2$, and $\Delta^{(2)}_n(s_1,s_2) \ge 0$ for $s_2 \ge 1$, and the second inequality holds because $\gamma_1 \le \mu_1$ and $\Delta_n(s_1,s_2) \le 0$ for $s_1 \ge 1$ and $s_2 \ge 1$.

Therefore,

$$M_n(s_1, s_2) = \quad R_1\mu_1 + (\gamma_1 - \mu_1)\Delta_n^{(1)}(s_1, s_2).$$

This completes the proof. □

Proof of Proposition 5.1.2: Proof of Proposition 5.1.2 follows immediately from Lemma A.0.4 since the result in Lemma A.0.4 implies that type 1 customers are prioritized for the N-period problem. Letting $N \to \infty$ completes the proof. □

Similar to the proof of Proposition 5.1.1, we need the following lemma to prove Proposition 5.2.1.

Lemma A.0.5. *For all $k = 1, \ldots, K$, $v(s) \geq v(s - e_k)$, where $s \in \mathcal{S}$ and $s_k \geq 1$.*

Proof of Lemma A.0.5: For all $k \in \{1, \ldots, K\}$, we follow the following argument. We first consider the finite horizon problem for which we will prove that $v_n(s) \geq v_n(s - e_k)$ for all periods $n = 0, \ldots, N$ and all states $s \in \mathcal{S}$, where $s_k \geq 1$. To prove this result, we will consider two sample paths.

In the first sample path, suppose that the state is $s - e_k$, where $s \in \mathcal{S}$ and $s_k \geq 1$ in period n, where $n = 0, \ldots, N$. Suppose that this sample path is governed by an optimal policy, which we call policy π. In the second sample path, suppose that the state is s in period n, where $n = 0, \ldots, N$. We will next construct a policy, which we call policy π_0, and apply this policy in the second sample path. Then, using induction on n, we will show that $v_n(s - e_k) \leq v_n^{(0)}(s)$, where $v_n^{(0)}(s)$ is the value function under policy π_0. Since π_0 is not necessarily an optimal policy, this will imply that $v_n(s - e_k) \leq v_n(s)$ for all $n \in \{0, \ldots, N\}$ and $s \in \mathcal{S}$.

First consider period N. In this case, we have $v_N^{(0)}(s) = v_N(s) = 0$ for all $s \in \mathcal{S}$ and hence the result holds trivially. Next, suppose that $v_{n+1}(s - e_k) \leq v_{n+1}^{(0)}(s)$ for some period $n + 1$, where $n \in \{0, \ldots, N - 1\}$, and all $s \in \mathcal{S}$, where $s_k \geq 1$, if π_0 does the same action in period $n + 1$ (with state s) that π takes in period $n + 1$ (with state $s - e_k$). We will show that this also holds for n.

Case 1 ($s_k < C_k$): At period n, the probability of the next event being an arrival, a service completion, or a departure of a job from stage i, where $i \neq k$, is the same for both sample paths. On the other hand, the probability of next event being the departure from stage k is larger in the second sample path, which means that the probability of staying in the same state (due to uniformization) is smaller in the second sample path. Hence, when we couple both sample paths, there can be two possible situations. Firstly, consider the situation where the next event is staying in the same state for the first sample path, and the next event is the departure of a job from stage k for the second sample path. If $k = K$, both sample paths reach the same state in period $n+1$, i.e., $s - e_K$, and then π_0 follows π exactly. Hence, $v_{n+1}(s) = v_{n+1}^{(0)}(s)$ for all $s \in \mathcal{S}$, which implies that $v_n(s-e_k) \leq v_n^{(0)}(s)$. Otherwise, the state under the first sample path is $s - e_k$ in period $n+1$ whereas it is $s - e_k + e_{k+1}$ under the second sample path. In other words, letting $s' := s - e_k + e_{k+1}$, the state under the first sample path is $s' - e_{k+1}$ in period $n+1$ whereas it is s' under the second sample path. As the inductive hypothesis holds for all $k \in \{1, \ldots, K\}$ and $s \in \mathcal{S}$, we have $v_{n+1}(s' - e_{k+1}) \leq v_{n+1}^{(0)}(s')$, and hence $v_n(s - e_k) \leq v_n^{(0)}(s)$ for all $s \in \mathcal{S}$ and $s_k \geq 1$. For the events not covered in first situation, the state under the first sample path is $s' - e_k$ in period $n+1$ whereas it is s'' under the second sample path, where $s'' \in \mathcal{S}$. π_0 takes the same action under the second sample path that π takes under the first sample path. Hence, by the inductive hypothesis, we have $v_n(s - e_k) \leq v_n^{(0)}(s)$ for all $s \in \mathcal{S}$ and $s_k \geq 1$.

Case 2 ($s_k = C_k$): At period n, the probabilities of next events are the same as in Case 1. Hence, the results for Case 1 also apply here. The only difference is that, if the next event is the arrival of a job to stage k, then that job is lost under the second sample path, whereas it will be admitted in the first sample path. Hence, both sample paths will reach the same state in period $n+1$, i.e., s. In this case, π_0 follows π exactly, which implies that $v_{n+1}(s) = v_{n+1}^{(0)}(s)$ for all $s \in \mathcal{S}$, and hence $v_n(s - e_k) \leq v_n^{(0)}(s)$.

Above we have proved that $v_n(s - e_k) \leq v_n^{(0)}(s) \leq v_n(s)$ for all $n = 0, \ldots, N$, $s \in \mathcal{S}$, and $s_k \geq 1$. Letting $N \to \infty$, we get $v(s - e_k) \leq v(s)$, which completes the proof. □

Proof of Proposition 5.2.1: To prove the result, we show that $M(s, a_K) \geq M(s, a_j)$ for all $j = 1, \ldots, K-1$. For all $j = 1, \ldots, K-1$, we have

$$\begin{aligned}
M(s,a_K) - M(s,a_j) &= R_K\mu_K - R_j\mu_j + (\gamma_K - \mu_K)\Big[v(s) - v(s-e_K)\Big] \\
&- \gamma_j\Big[v(s) - v(s-e_j+e_{j+1})\Big] + \mu_j\Big[v(s) - v(s-e_j)\Big] \\
&\geq R_K\mu_K - R_j\mu_j + (\gamma_K - \mu_K)\Big[v(s) - v(s-e_K)\Big] - (\gamma_j - \mu_j)\Big[v(s) - v(s-e_j)\Big] \\
&\geq R_K\mu_K - R_j\mu_j \\
&\geq 0,
\end{aligned}$$

where the first and second inequalities hold from Lemma A.0.5 and the fact that $\gamma_K \geq \mu_K$ and $\gamma_j \leq \mu_j$

for all $j = 1, \ldots, K-1$, and the third inequality holds because $R_K\mu_K \geq R_j\mu_j$ for all $j = 1, \ldots, K-1$. This completes the proof. $\square$

Bibliography

[1] Argon, N., J. E. Winslow, and S. Ziya, "Triage in the aftermath of mass-casualty incidents," to appear in the *Wiley Encyclopedia of Operations Research and Management Science* (2010).

[2] Argon, N., S. Ziya, and, R. Righter, "Scheduling impatient jobs in a clearing system with insights on patient triage in mass casualty incidents," *Probability in the Engineering and Informational Sciences* **22** (2008), No. 3, 301–332.

[3] Armony, M., and C. Maglaras, "Contact centers with a call-back option and real-time delay information," *Operations Research* **52** (2004), No. 4, 527-545.

[4] Armony, M., and C. Maglaras, "On customer contact centers with a call-back option: Customer decisions, routing rules and system design," *Operations Research* **52** (2004), No. 2, 271-292.

[5] Armony, M., N. Shimkin, and W. Whitt, "The impact of delay announcements in manyserver queues with abandonment," *Operations Research* **57** (2009), No. 2, 66-81.

[6] Atar, R., C. Giat, and N. Shimkin, "The $c\mu/\theta$ rule for many servers with abandonment," Unpublished manuscipt, Department of Electrical Engineering, Techion-Israel Institute of Technology, Haifa, Israel, 2009.

[7] Bae, J., S. Kim, and E. Y. Lee, "The virtual waiting time of M/G/1 queue with impatient customers," *Queueing Systems* **38** (2001), 485–494.

[8] Barrer, D. Y., "Queueing with impatient customers and indifferent clerks," *Operations Research* **5** (1957), 644–649.

[9] Barrer, D. Y., "Queueing with impatient customers and ordered service," *Operations Research* **5** (1957), 650–656.

[10] Bernard, M., and P. R. Mathews, "Evacuation of a maternal–newborn area during hurricane Katrina," *MCN: The American Journal of Maternal/Child Nursing* **33** (2008), No. 4, 213–223.

[11] Bhattacharya, P. P., and A. Ephremides, "Two competing queues with time-critical customers: Optimal server assignment," *Proceedings of the 28th IEEE Conference on Decision and Control* (1989), 1069–1070.

[12] Bhattacharya, P. P., and A. Ephremides, "Optimal scheduling with strict deadlines," *IEEE Transactions on Automatic Control* **34** (1989), 721–728.

[13] Bhattacharya, P. P., and A. Ephremides, "Optimal allocation of a server between two queues with due times," *IEEE Transactions on Automatic Control* **36** (1991), 1417–1423.

[14] Boots, N. K., and H. Tijms "A multi-server queueing system with impatience customers," *Management Science* **45** (1999), 444–448.

[15] Boxma, O. J., and F. G. Forst, "Minimizing the expected weighted number of tardy jobs in stochastic flow shops," *Operations Research Letters* **5** (1986), No. 3, 119–126.

[16] Brandt, A., and M. Brandt, "On the M(n)/M(n)/s queue with impatient calls," *Performance Evaluation* **35** (1999), 1–18.

[17] Brandt, A., and M. Brandt, "On a two queue priority system with impatience and its applications to a call center," *Methodology and Computing in Applied Probabability* **1** (1999), 191–210.

[18] Brandt, A., and M. Brandt, "Asymptotic results and a Markovian appproximation for the M(n)/M(n)/s + GI system," *Queueing Systems* **41** (2002), 73–94.

[19] Brandt, A., and M. Brandt, "On the two-class M/M/1 system under preemptive resume and impatience of the prioritized customers," *Queueing Systems* **47** (2004), 147–168.

[20] Brouns, G. A. J. F., and J. van der Wal, "Optimal threshold policies in a workload model with a variable number of service phases per job," *Operations Research* **58** (2003), 483–501.

[21] Childers, A. K., G. Visagamurthy, and K. Taaffe, "Prioritizing patients for evacuation from a healthcare facility," to appear in *Transportation Research Record: Journal of the Transportation Research Board*, 2010.

[22] Choi, B. D., B. Kim, and J. Chung, "M/M/1 queue with impatient customers of higher priority," *Queueing Systems* **38** (2001), 49–66.

[23] Coffman, E. G., L. Flatto, M. R. Garey, and R. R. Weber, "Minimizing expected makespan on uniform processor systems," *Advances in Applied Probability* **19** (1987), 177–201.

[24] Cone, D. C., and D. S. MacMillan, "Mass-casualty triage systems: A hint of science," *Academic Emergency Medicine* **12** (2005), No. 8, 739–741.

[25] Darr, K., "Katrina: lessons from the aftermath," *Hospital Topics* **84** (2006), No. 2, 30–33.

[26] Down, D. D., G. Koole, and M. E. Lewis, "Dynamic control of a single-server system with abandonments," under review (2010).

[27] Emmons, H., and M. Pinedo, "Scheduling stochastic jobs with due dates on parallel machines," *European Journal of Operational Research* **47** (1990), 49–55.

[28] Fink, S., "The deadly choices at Memorial," *The New York Times Magazine*, August 30, 2009, 28–46.

[29] Frykberg, E. R., "Medical management of disasters and mass casualties from terrorist bombings: How can we cope?" *The Journal of Trauma* **53** (2002), No. 2, 201–212.

[30] Ghamami, S., and A. Ward, " Dynamic scheduling of an N-system with reneging," preprint (2009).

[31] Glazebrook, K. D., P. S. Ansell, R. T. Dunn, and R. R. Lumley "On the optimal allocation of impatient tasks," *Journal of Applied Probability* **41** (2004), 51–72.

[32] Green, L., "A multiple dispatch queueing model of police patrol operations," *Management Science* **30** (1984), No. 6, 653–664.

[33] Green, L., and P. Kolesar, "A comparison of the multiple dispatch and M/M/c priority queueing models of police patrol," *Management Science* **30** (1984), No. 6, 665–670.

[34] Green, L., and P. Kolesar, "The feasibility of one-officer patrol in New York City," *Management Science* **30** (1984), No. 8, 964–981.

[35] Green, L., and P. Kolesar, "Testing the validity of a queueing model of police patrol," *Management Science* **35** (1989), No. 2, 127–148.

[36] Green, L., and P. Kolesar, “Improving emergency responsiveness with Management Science,” *Management Science* **50** (2004), No. 8, 1001–1014.

[37] Haugen, R. B., and E. Skogan, “Queueing systems with stochastic time-out,” *IEEE Transactions on Communications* **COM-28** (1980), 1984–1989.

[38] Holt, G. R., “Making difficult ethical decisions in patient care during natural disasters and other mass casualty events,” *Otolaryngology* **-** *Head and Neck Surgery* **139** (2008), 181–186.

[39] Hougaard, P., *Analysis of Multivariate Survival Data*, Springer-Verlag, New York, NY, 2000.

[40] Iravani, F., and B. Balcıoğlu, “ On priority queues with impatient customers,” *Queueing Systems*, **58** (2008), 239-260.

[41] Jenkins, J. L., M. L. McCarthy, L. M. Sauer, G. B. Green, S. Stuart, T. L. Thomas, and E. B. Hsu, “Mass-casualty triage: Time for an evidence-based approach,” *Prehospital and Disaster Medicine* **23** (2008), No. 1, 3–8.

[42] Johansen, S. G., and C. Larsen, “Computation of a near-optimal service policy for a single-server queue with homogeneous jobs,” *European Journal of Operational Research* **134** (2001), 648–663.

[43] Kaplan, E. H., D. L. Craft, and L. M. Wein,“Emergency response to a smallpox attack: The case for mass vaccination,” *Proceedings of the National Academy of Sciences* **99** (2002), 10935–10940.

[44] Kim, S., J. Bae, and E. Y. Lee, “Busy periods of Poisson arrival queues with loss,” *Queueing Systems* **39** (2001), 201–212.

[45] Koçağa, Y., and A. Ward, “ Admission control for a multi-server queue with abandonment,” preprint (2009).

[46] Kolesar, P., and A. Swersey, “The deployment of urban emergency units: A survey,” *Studies in the Management Sciences* **22** (1986), 87–120.

[47] Larkin, G. L., and J. Arnold, “Ethical considerations in emergency planning, preparedness, and response to acts of terrorism,” *Prehospital Disaster Medicine* **18** (2003), No. 3, 170–178.

[48] Larson, R. C., and T. Rich, “Travel time analysis of New York City Police patrol cars,” *Interfaces* **17** (1987), No. 2, 15–20.

[49] Larson, R. C., “Public sector operations reseach: A personal journey,” *Operations Research* **50** (2002), No. 1, 135–145.

[50] Lerner, E. B., et al. “Mass casualty triage: An evaluation of the data and development of a proposed national guideline,” *Disaster Medicine and Public Health Preparedness* **2** (2008), Supplement 1, S25–S34.

[51] Li, D., and K. D. Glazebrook, “An approximate dynamic programming approach to the development of heurtistics for the scheduling of impatient jobs in a clearing system,” to appear in *Naval Research Logistics*, 2010.

[52] Lippman, D. P., “Applying a new device in the optimization of exponential queueing systems,” *Operations Research* **23** (1975), 687–710.

[53] Liu, C. L., and J. W. Layland, “Scheduling algorithms for multiprogramming in a hard real-time environment,” *Journal of the Association for Computing Machinery* **20** (1973), 46–61.

[54] Nocera, A., and A. Garner, “An Australian mass-casualty incident triage system for the future based upon triage mistakes of the past: The homebush triage standard,” *Australian and New Zealand Journal of Surgery* **69** (1999), 603–608.

[55] Okie, S., “Dr. Pou and the hurricane – implications for patient care during disasters,” *The New England Journal of Medicine* **358** (2008), No. 1, 1–5.

[56] Pandelis, D. G., and D. Teneketzis “Optimal multiserver stochastic scheduling of two interconnected priority queues,” *Advances in Applied Probability* **26** (1994), 258–279.

[57] Panwar, S. S., D. Towsley, and J. K. Wolf, “Optimal scheduling for a class of queues with customer deadlines to the beginning of service,” *Journal of the Association for Computing Machinery* **35** (1988), 832–844.

[58] Perry, O., and W. Whitt, “Responding to unexpected overloads in large-scale service systems,” submitted to *Management Science*, manuscipt.

[59] Pinedo, M., “Stochastic scheduling with release dates and due dates,” *Operations Research* **31** (1983), 559–572.

[60] Pinedo, M., *Scheduling: Theory, Algorithms, and Systems*, Prentice Hall, Englewood Cliffs, NJ, 2002.

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

[62] Righter, R., “Scheduling,” *Stochastic Orders and Their Applications*, Chapter 13, edited by M. Shaked and J. G. Shanthikumar. New York: Academic Press, 381–432, 1994.

[63] Righter, R., “Job scheduling to minimize weighted flowtime on uniform processors,” *Systems and Control Letters* **10** (1988), 211–216.

[64] Righter, R., “Expulsion and scheduling control for multiclass queues with heterogeneous servers,” *Queueing Systems* **34** (2000), 289–300.

[65] Rund, D. A., and Rausch, T. S., *Triage*, The C. V. Mosby Company, St. Louis, MO, 1981.

[66] Sacco, W. J., M. Navin, K. E. Fiedler, R. K. Waddell II, W. B. Long, R. F. Buckman Jr., “Precise formulation and evidence-based application of resource-constrained triage,” *Academic Emergency Medicine* **12** (2005), 759–770.

[67] Sacco, W. J., M. Navin, R. K. Waddell II, K. E. Fiedler, , W. B. Long, R. F. Buckman Jr., “A new resource-constrained triage method applied to victims of penetrating injury,” *The Journal of Trauma* **63** (2007), 316–325.

[68] Schaack, C., and R. C. Larson, “An N-server cut-off priority queue,” *Operations Research* **34** (1986), No. 2, 257–266.

[69] Schaack, C., and R. C. Larson, “An N-server cut-off priority queue where arriving customers request a random number of servers,” *Management Science* **35** (1989), No. 5, 614–634.

[70] Shaked, M., and J. G. Shanthikumar, *Stochastic Orders and their Applications*, Academic Press, Boston, MA, 1994.

[71] Mueller, A., and D. Stoyan, *Comparison Methods for Stochastic Models and Risks*, John Wiley and Sons, Chichester, UK, 2002.

[72] Swersey, A., "The deployment of police, fire and emergency medical units," *Handbooks in Operations Reseach and Management Science* **6** (1994), 151–190.

[73] Ward, A. R., and P. W. Glynn, "A Diffusion approximation for a Markovian queue with reneging," *Queueing Systems* **43** (2003), 103–128.

[74] Weber, R. R., P. Varaiya, and J. Walrand, "Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flow time," *Journal of Applied Probability* **23** (1986), 841–847.

[75] Wein, L. M., D. L. Craft, and E. H. Kaplan, "Emergency response to an anthrax attack," *Proceedings of the National Academy of Sciences* **100** (2003), 4346–4351.

[76] Weiss, G., and M. Pinedo, "Scheduling tasks with exponential service times on non-identical processors to minimize various cost functions," *Journal of Applied Probability* **17** (1980), 187–202.

[77] Whitt, W., "Improving service by informing customers about anticipated delays," *Management Science* **45** (1999), 192–207.

[78] Xu, S. H., "A duality approach to admission and scheduling control of queues," *Queueing Systems* **18** (1994), 273–300.

[79] Xu, S. H., and J. G. Shanthikumar, "Optimal expulsion control—a dual approach to admission control of an ordered-entry system," *Operations Research* **41** (1993), 1137–1152.

[80] Zhao, Z., S. S. Panwar, and D. Towsley, "Queueing performance with impatient customers," *Proceedings of the 10th IEEE Conference on Computer Communications* (1991), 400–409.