PRIORITY SCHEDULING OF JOBS WITH HIDDEN TYPES

Zhankun Sun

A dissertation 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
2014

Approved by:

Nilay Tanık Argon

Serhan Ziya

Amarjit Budhiraja

Vidyadhar G. Kulkarni

Peng Sun

ABSTRACT

ZHANKUN SUN: PRIORITY SCHEDULING OF JOBS WITH HIDDEN TYPES.
(Under the direction of Nilay Tanık Argon and Serhan Ziya.)

In service systems, prioritization with respect to the relative "importance" of jobs helps allocate the limited resources efficiently. However, the information that is crucial to determine the importance level of a job may not be available immediately, but can be revealed through some preliminary investigation. While investigation provides useful information, it also delays the provision of services. Therefore, it is not clear if and when such an investigation should be carried out. To provide insights into this question, we consider a service system with a single server and the two possible types of jobs, where each type is characterized by its waiting cost and expected service time. Jobs' type identities are initially unknown, but the service provider has the option to spend time on investigation to determine the type of a job albeit with a possibility of making an incorrect determination. Our objective is to identify policies that balance the time spent on information extraction with the time spent on service. In this dissertation we consider two settings: one with finitely many jobs present at time zero and no external arrivals; the other with exogenous arrivals.

Under the assumption of linear waiting cost, our study on the first model reveals that investigation is less likely to be beneficial when one of the types is significantly dominated by the other in terms of numbers, or the two types of jobs are not significantly different from each other with respect to their importance. More interestingly, we find that if the server decides to do investigation for all jobs, it is possible that more accurate information might result in higher costs. We prove that the optimal dynamic policy can be characterized by a switching curve. One insight that comes out of this characterization is that the server should start with performing investigation when there are sufficiently many jobs at the beginning and never perform investigation when there are few jobs. Numerical study shows that the optimal policy could improve significantly upon some simple baseline policies. Heuristic policies developed based on the optimal policy perform well even with nonlinear holding cost.

When there are external arrivals to the system, we show that the optimal dynamic policy is of threshold type. The structure of the optimal policy implies that when there are few less-important jobs waiting for service, the server should perform investigation; otherwise, the server should stop investigation and serve jobs directly. Given that it is almost impossible to obtain an analytical expression of the threshold, we develop a heuristic policy based on the results for the clearing system. We carry out a simulation study and find that the heuristic policy performs significantly better than No-Triage Policy in most cases; for the rest, it performs at least as well as No-Triage Policy.

Finally, we study three extensions. The first extends the clearing system by considering multiple parallel servers. The second studies a queueing system in which investigation is instantaneous but incurs a fixed cost, and the last one extends the queueing system by assuming that investigation has to be done before service.

ACKNOWLEDGMENTS

I would like to thank my parents and my sister for their constant love and support. I thank my advisers, Professor Nilay Tanık Argon and Professor Serhan Ziya, for their guidance and encouragement along the course of my graduate study. This thesis would not have been possible without their help and support. My appreciation also goes to other members of my dissertation committee, Professor Vidyadhar G. Kulkarni, Professor Amarjit Budhiraja and Professor Peng Sun (Duke). I thank my friends from Department of Statistics and Operations Research, especially Chao Deng, Jianzhe Luo, Haojin Zhai, Xiaosun Lu, Liang Yin, Minghui Liu, Gen Li and Dong Wang. My life is more colorful because of you. A special thanks goes to my friend Yichen Zhang. Finally, I would like to thank my wife, Huiyin Ouyang, and my daughter Linxi Sun, who always stood by me through the good times and bad.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

In service systems, first-come-first-serve is a frequently used service discipline, however, in reality there are many situations of practical interest where customers are served not in the order of their arrival but according to the priorities that are assigned based on their relative "importance." In this dissertation, we refer this type of service as *priority scheduling*. Priority scheduling is prevalent in service systems. Especially when service capacity is limited, prioritization helps allocate this limited resource in a way that aligns with the overall objectives of the service provider. Priority scheduling has been practically applied in call centers, banks, machine maintenance, Emergency Departments (ED) of hospitals, military communications, etc. For example, in the EDs, the patients who need immediate medical attention will be seen first if the existing patients can be delayed.

Priority scheduling requires information about the jobs (we use *jobs* to denote customers, machines, parts, patients, etc.) to assign them priorities. This information is sometimes immediately available and can be used to determine priority levels. For example, a service provider who is interested in providing priority service to its "good" customers, might be able to use past data to determine its customers' priority classes instantly as they arrive. In some cases, however, the information that is crucial to determine the priority level of a job is not available immediately but can be obtained with some investigation. This investigation produces useful information but at the expense of delaying the service process. It is not clear if and when engaging in such investigation justifies the extra delay imposed. The goal of this research is to shed some light on this question.

Specific examples will help illustrate the practical relevance of the information/delay trade-off described above. When healthcare resources are severely restricted in comparison with the urgent demand as in the case of mass-casualty incidents (MCI)[1] or clinics in rural areas and underdeveloped countries, patients go through a process called "triage" before they are given treatment. The objective of triage is to determine the seriousness of the patients' conditions and prioritize them accordingly.

[1] An MCI is any incident in which emergency medical services resources, such as personnel and equipment, are overwhelmed by the number and severity of casualties. See Hafen et al. (1999)

When on-site medical personnel is not very limited in numbers, triage and treatment can proceed simultaneously and therefore unless triage takes unusually long it does not typically lead to delays in treatment or transportation. However, in austere mass-casualty conditions, battlefields, and clinics in economically deprived areas where in some cases healthcare services are delivered through mobile clinics, a single person or a team can be in charge, which necessitate a careful balancing of time spent on triage and time spent on treatment or a more thorough examination of the patients. The information/delay trade-off also appears in other contexts, such as prioritization of requests submitted daily to internal maintenance and repair departments (Taghipour et al., 2011); prioritization of sales leads in marketing particularly in business-to-business settings (Lichtenthal et al., 1989; Wilson, 2003; D'Haen and den Poel, 2013), where time is invested to assess the likelihood of existing leads to be successfully converted to actual sales; and intelligence (particularly human intelligence) collection management (Kaplan, 2010, 2012; Ni et al., 2013), where agents make some initial investigation of existing ambiguous cues, which might possibly be pointing to potential terrorist activities, and prioritize them prior to more in-depth investigation.

Despite the fact that these examples arise in very different contexts they share some key features: jobs are heterogeneous regarding their "importance" (or "urgency") and possibly their service requirement. The decision maker knows that the jobs are heterogenous but there is no information readily available, which can help in distinguishing one job from another. Investigating any given job reveals some information for that job, which then can be used to determine whether or not the job should get a priority in service but this information can be noisy and thus may lead to an incorrect classification. Furthermore, the investigation is "costly" in the sense that it takes time and resources. Spending time in investigation essentially eats away from the time that can be spent in actually serving the jobs. Thus, in all these examples, the fundamental goal is to carefully balance the time spent on information extraction with the time spent on serving the jobs.

In this thesis, we aim to contribute to the understanding of this trade-off and provide insights on how the decision on investigation should be made in order to achieve such a balance. The goal is to develop a generic formulation whose analysis leads to insights into how that can be done, rather than to model any single one of the application contexts mentioned above with its unique features.

We approach this problem by considering a single server model with two types of jobs. The server can serve a job without knowing its type. Alternatively, the server can triage a job, place the job in a particular class, which correlates with the type of the job, and then either proceed to serve that job right away or put that job aside for awhile in order to serve later. (We borrow the medical terminology "triage" to refer to the investigative process which results in the classification of jobs.) Triage is imperfect meaning that jobs can be classified incorrectly. The type of a job determines the expected service time for that job and the "cost" of keeping the job waiting. The objective is to minimize the total expected cost or the long-run average cost, depending on the specific model settings.

The remainder of this thesis is organized as follows. Chapter 2 reviews the literature on job scheduling problems and discuss how this work will contribute to the literature. Chapter 3 presents the description and analysis of our clearing model[2] . We first compare four simple policies and the analysis leads to some seemingly counter-intuitive findings. Then we provide a complete characterization of the optimal dynamic policy. In particular, we find that there is a switching curve that separates the states in which triage should be performed from the others. One interesting insight that comes out of this characterization is that spending time on triage helps if there are sufficiently many jobs but not when there are relatively few. Our analytical results assume that waiting cost is a linear function of time. A numerical study reveals that even though the structure of the optimal policy can be different when the waiting cost function is not linear, the heuristic policies developed based on the results in our model with linearity assumption perform well.

In Chapter 4, we study the information/delay trade-off in a setting where there are external arrivals to the system. With the assumption of a Poisson arrival stream and independent and identical (i.i.d) exponential service times, we show that the optimal policy on whether to triage or not in order to minimize the long-run average cost is characterized by a switching curve. To prove the structure of the optimal policy, we show various properties of the optimal value functions of a corresponding model with discounting, then extend these results to the optimal bias functions in our original model. With a simulation study, we observe that a heuristic policy of threshold type can improve significantly over the policy of skipping triage all the time.

In Chapter 5, we study three extensions to the models in Chapters 3 and 4. In Section 5.1, we study a clearing model with multiple identical servers instead of a single server. In Section 5.2, we

[2] A clearing model means there are finite number of jobs present at time zero and no outside arrivals to the system.

consider a model where there are external arrivals and triage is instantaneous but incurs a fixed cost. In Section 5.3, we analyze the case where triage is required for service, in which case the decision is to determine the class to be prioritized after triage: an untriaged job, or a job that is classified as class-1 or class-2. In each section, we describe the model assumptions and provide partial or complete characterization of the optimal policy.

Finally, we conclude our study in Chapter 6 and point to some future research directions.

CHAPTER 2: LITERATURE REVIEW

There are two streams of papers that are relevant to our work: (i) traditional job scheduling and (ii) priority scheduling under imperfect information on job identities.

2.1 Priority scheduling with perfect information

Within the context of this dissertation, job scheduling is the process of determining the order according to which jobs of different types will be processed. There are many different versions of the job scheduling problem. For example, a clearing system versus a system with exogenous arrivals, a single-server system versus a multi-server system, deterministic settings versus stochastic environments, preemption versus non-preemption, linear cost versus nonlinear cost, etc. There can be also different objectives, depending on the settings of the specific job scheduling problem, such as minimizing the total (or average) waiting time (or cost), minimizing the total tardiness or the number of tardy jobs, etc. There exists substantive work on job scheduling problems. Pinedo (2008) provides an extensive review of the scheduling problems that have been studied. We here review only the most related work that establish the optimality of the $c\mu$ rule under various conditions.

There are several papers that prove the optimality of the $c\mu$ rule in settings where there is no external arrival. Smith (1956) studies a single-stage production system with deterministic processing times and identical release times for all jobs. The $c\mu$ rule is shown to be optimal to minimize the weighted sum of job completion times. Since then, the $c\mu$ rule and its various generalizations are shown to be optimal in models with different settings. Pinedo (1983) considers the stochastic counterpart of the above model where the processing time of job j is exponentially distributed with rate λ_j and the release time is a random variable with arbitrary distribution. Preemption is allowed. The author showed that it is optimal to process the job with the highest value of $c_j\lambda_j$ among those available.

Cox and Smith (1961) appears to be the first to show that the $c\mu$ rule is the optimal static policy for a multiclass $M/G/1$ queue, where the objective is to minimize the long-run average waiting cost and service is non-preemptive. Kakalik and Little (1971) shows that the optimality of the $c\mu$ rule holds in the larger class of state-dependent dynamic policies as well, regardless of the option of idling the server. Klimov (1974) extends the optimality of the $c\mu$ rule to a multiclass $M/G/1$ queue with feedback. Harrison (1975) considers a multiclass $M/G/1$ system with discounted holding costs and shows that a static priority rule is optimal. Tcha and Pliska (1977) studies a model that combines discounting and feedback, and show that a static priority rule is optimal. Hirayama et al. (1989) studies a discrete-time $G/G/1$ queue with two classes under non-preemptive service discipline. The $c\mu$ rule is shown to be optimal to minimize the total holding cost in a finite-horizon scheduling period if the service times have a decreasing failure rate (DFR). Nain (1989) extends the optimality of the $c\mu$ rule to a multiclass $G/M/1$ queue with or without feedback. The paper also considers two $G/M/1$ queues in tandem and shows that the $c\mu$ rule is optimal for the second queue in that it minimizes the discounted holding cost. In the single-machine scheduling with arbitrary arrivals and machine breakdowns under a preemptive-resume discipline, Righter (1994) shows that processing jobs according to the non-increasing order of $\omega\mu$ value maximizes the number of correctly completed jobs by any time t when processing times have a DFR and ω_i is the probability that job i will be correctly completed. Recently Budhiraja et al. (2012) studied a multiclass $M(\nu)/M(\nu)/1$ model where the arrival and service rates fluctuate with a changing environment, described by the environment variable ν. The authors proved that the $c\mu$ rule is asymptotically optimal for minimizing an infinite-horizon discounted cost function.

The papers we mentioned so far all assume linear waiting costs. Van Mieghem (1995) is the first to prove the asymptotic optimality of the $c\mu$ rule in models with nonlinear costs. Specially, the model studied is a $G/G/1$ queue with multiclass jobs and the cost incurred by a job is a convex function of the job's sojourn time in the system. A generalized version of the $c\mu$ rule, or the so called generalized-$c\mu$ rule ($Gc\mu$-rule), is shown to be asymptotically optimal in heavy traffic in that it minimizes the total cumulative delay cost for a finite time horizon. The optimality of the $Gc\mu$-rule is robust in that it holds for a countable number of classes of jobs and several homogeneous servers. Mandelbaum and Stolyar (2004) has extended the optimality of the $Gc\mu$-rule to multiple flexible servers in parallel.

All the papers we mentioned so far assume that the information about the jobs, such as which type the job belongs to, is perfectly known. In practice, however, the information is usually partially known or unknown and can be collected (imperfectly) through diagnosis (*triage*). Our work is fundamentally different from the above in that we consider triage together with the service process.

2.2 Priority scheduling with imperfect information

Compared with the traditional job scheduling literature, there is limited work that deal with scheduling under imperfect information on job identities. Van Der Zee and Theil (1961) appears to be the first work that considered the misclassification problem in priority queues. The authors study a single-server queue with two priority classes having expected service times $E(s_1)$ and $E(s_2)$, respectively. Without loss of generality, assume $E(s_1) < E(s_2)$. Class 1 jobs arrive to the system according to a Poisson process with rate λ_1 but are misclassified into class 2 with rate δ_1, class 2 jobs arrive to the system according to a Poisson process with rate λ_2 but are mistakenly assigned into class 1 with rate δ_2. Under the assumption of misclassification, they find that prioritizing class 1 is better than FCFS in the sense of minimizing the expected waiting time if

$$\delta_1/\lambda_1 + \delta_2/\lambda_2 < 1. \tag{2.1}$$

They also develop a fixed-priority policy where there are three priority classes and find a condition under which this policy is no worse than FCFS by approximate analysis.

While van der Zee and Theil assume that the jobs are classified and priorities are assigned automatically, Argon and Ziya (2009) study the problem of how to assign priorities to the jobs based on partial information on the job identities to minimize the long-run average waiting cost. The authors consider an $M/G/1$ queue with two types of customers. The identity of each arrival is partially known in the sense that each customer brings a signal indicating the probability of being the important type. The authors show that increasing the number of priority classes decreases costs and it is optimal to give the highest priority to the customer with the highest signal. The authors also consider two-class priority policies and find the optimal cut-off level for the signal to obtain the two priority classes. The main difference of our model from theirs is that in their model the signal is free in that there is no need for triage to obtain the signal, while in our model the type identities have to be obtained through triage, perfectly or imperfectly.

The following two papers that are most related work to ours assume that the job identities are unknown as well. Alizamir et al. (2012) consider a queueing model with Poisson arrivals where each customer comes from one of two types but the server does not know which type the customer belongs to. The server diagnoses each customer through a series of independent tests and classifies it based on the server's belief. If the classification is correct, there is a reward; otherwise, there is a penalty. Each customer incurs a waiting cost during the customer's stay in the system. The authors find the optimal policy on how many tests to do to classify an arriving customer. Our model is different from theirs in that the jobs go through a service process after classification. On the contrary, their model focuses on the diagnostic process and the server does not perform any service after classification. Dobson and Sainathan (2011) does consider the classification and service in one model. The authors compare two models and both with Poisson arrivals. In one model jobs are first sorted by a pool of homogeneous sorters and then served by another pool of homogeneous processors (so called the prioritized model) while there are no sorting in the other model (so called the base model). The main goal is to find the optimal average waiting cost for the prioritized model by appropriately setting the number of sorters and processors under an exogenous budget constraint and compare the optimal waiting cost of the prioritized model and that of the base model. They find that sorting does not always benefit the system. Our model is different since we consider for a fixed number of servers that are capable of performing both triage and service tasks. More specifically, we concentrate on control decisions that are made dynamically based on the system state whereas Dobson and Sainathan (2011) focuses on a design problem.

Finally we would like to note that several authors have studied models on medical service with patient triage although with completely different research questions. Shumsky and Pinker (2003), motivated by a healthcare problem, consider a model where a firm hires a gatekeeper to make an initial diagnosis on each arriving customer then decides to solve the customer's problem or refer the customer to a specialist. Their focus is to design an incentive mechanism to lead the gatekeepers and specialists to make referral decisions that are optimal for the firm. Wang et al. (2010) study patient behaviors under a strategic queueing setting. Triage nurses provide advice on treatment after diagnosis, patients have autonomy to accept or decline the service, based on their expectation on the diagnostic accuracy and waiting time. They focus on the trade-off between diagnostic depth and congestion levels and the subsequent treatment is not modeled. Saghafian et al. (2012) consider a mechanism (streaming) to separate patients and medical resources into two streams based on the prediction of triage nurses in an Emergency Department (ED) of a hospital. They find conditions when streaming can improve the performance. They also examine the effects of misclassification by simulation and conclude that better triage information about patients can level up the performance of ED. In another paper, Saghafian et al. (2011) develop a new priority rule, namely "complexity-based triage," to classify patients in the Emergency Department. While misclassification in triage is considered in their model, the delay caused by triage is not. Their objective is to reduce the risk of adverse events for patients and improve operational efficiency (by shortening the average length of stay). Dobson et al. (2013) study a model in which an investigator collects information from a new customer to decide what work needs to be done in the second step by another server. Once the second step is finished the customer joins another queue to receive service from the investigator again and then leaves the system. The investigator needs to prioritize between the old and new customers. As we describe in the following chapters, these models are significantly different from the ones we analyze in this dissertation.

CHAPTER 3: PRIORITY SCHEDULING OF JOBS WITH HIDDEN TYPES IN A CLEARING SYSTEM

3.1 Introduction

In this chapter, we investigate a problem that is similar to traditional scheduling problems with some important differences. We assume there are finitely many jobs at the beginning. The exact types, characterized by the service times and hold cost rates, are unknown to the server initially, however, the server has the option to spend some time to extract these information. Thus, our focus is on settings where an unexpected event triggers the sudden appearance of a large number of jobs to take care of (as in the case of mass-casualty events which necessitate patient triage and prioritization or search and rescue operations), where jobs accumulate at the beginning of a service period, say in the morning (as in the case of patients lining up to be seen in mobile clinics), or where a certain number of tasks are assigned to a single person (for example, a salesperson or an intelligence agent) to take care of over a certain period of time. The objective is to find the optimal policy that minimize the total expected cost by assigning priorities to the jobs and making decisions on whether or not to spend time to extract the type information.

The remainder of this chapter is organized as follows: Section 3.2 gives a detailed description of the formulation. In Section 3.3, we compare four simple policies, which are all easy-to-implement in practice and also serve as benchmark policies. Section 3.4 provides a complete characterization of the optimal dynamic policy, which allows making decisions based on the up-to-date system state. We carried out a numerical study in Section 3.5. We observe that the optimal policy improves the benchmark policies significant, and when the waiting cost is in fact not a linear function (which we assume it is), the heuristic policies developed based on our results perform well compared to the optimal policy.

3.2 Model description

We consider a single-server clearing system in which at time $t = 0$ there are $N \geq 2$ jobs waiting to be served. There will be no new job arrivals. Each job belongs to one of two types: type-1 or type-2. The probability of a randomly chosen job being of type 1 is $p \in (0, 1)$ and that of type 2 is $q = 1 - p$ independently of all the other jobs as well as the service process. (The cases that $p = 0$ or 1 are trivial and not of interest.) A job from type i incurs a holding cost of h_i for each unit of time the job stays in the system, and the expected service time for a type i job is $\tau_i < \infty,\ i = 1, 2$, with some general distribution. We assume that the service times of all jobs are independent conditional on their types. Once the service of a job is over, it leaves the system. The objective of the service provider is to minimize the expected total waiting cost of all jobs. If the type of each job were to be known, according to the well-known $c\mu$-rule, the optimal policy would be to give priority to type-1 jobs if $h_1/\tau_1 \geq h_2/\tau_2$ and to type-2 jobs otherwise. However, in our problem, while p is known, the types of jobs are unknown to the decision maker.

The server does not need to know the type of the job to serve it but s/he can choose to perform an investigative task first in an effort to learn more about the type of the job, which can be used to determine the service order. Following the medical terminology, we call this investigative task *triage* and the act of performing triage *to triage*. Triage time for each job is independent of everything else including the job's type and its expected value is denoted by $u < \infty$. As in the case of service times, we make no further assumptions on the distribution of triage times. As a result of triage, the job is classified as either class-1 or class-2. Note that the *type* of a job is an inherent characteristic unknown to the decision maker while its *class* is an attribute assigned after triage and is observed by the decision maker. Once a job is classified, the server can either proceed to serve the job immediately or simply puts it away making note of its class information, and moves on to another job. The service time of a job does not change depending on whether or not the job's class information is available. It only depends on the type of the job.

Under perfect triage, all type-1 jobs would be classified as class-1 and all type-2 jobs would be classified as class-2. However, triage is prone to errors and therefore jobs of either type can be classified as class-1 or class-2. If a type-1 job is classified as class-2 or a type-2 job is classified as class-1, we say that the job is *misclassified*. Let v_i denote the probability of classifying a type-i job as class-i where $v_i \in [0, 1]$ for $i = 1, 2$. Without loss of generality, we make the following two assumptions throughout this chapter unless otherwise stated:

Assumption 3.2.1. *(i)* $h_1/\tau_1 > h_2/\tau_2$; *(ii)* $v_1 + v_2 > 1$.

Part (i) of Assumption 3.2.1 implies that if the type information for all jobs were readily available, the optimal policy would simply give priority to type-1 jobs in accordance with the $c\mu$-rule. Part (ii) together with part (i) of Assumption 3.2.1 imply that if type information were not available but class information were immediately available, the optimal policy would give priority to class-1 jobs, which again follows from the $c\mu$-rule. (If part (ii) of Assumption 3.2.1 does not hold, i.e. $v_1+v_2 \leq 1$, but part (i) does, then if the class-information of jobs were immediately available, the optimal policy would be to give priority to class-2 jobs.) In the rest of this chapter, to distinguish between the two types and the two classes, we will refer to type-1 and class-1 as the *important* type and class, respectively. Note that due to the possibility of misclassification, some of the jobs that are classified as important may in fact not belong to the important type but from the perspective of the server, all jobs classified as important are treated as being important.

In the following section, we first investigate and compare the performances of four benchmark policies, which naturally arise as simple heuristics and thus are practically appealing.

3.3 Benchmark policies

We first define the four policies we analyze in this section:

No-Triage Policy (NT**)** Jobs are served in random order. No job goes through triage.

Triage-All-First Policy (TAF) First, all jobs go through triage in random order. Then, class-1 jobs are given priority in agreement with the $c\mu$-rule, i.e. all class-1 jobs are served before all class-2 jobs.

Triage-Prioritize-Class-1 Policy (TP_1**)** Each job, with the exception of the last one, goes through triage in random order. If a job is classified as class-1, it is served right away; otherwise, the job is put aside to be served later. When the triage of $N - 1$ jobs is completed, the remaining untriaged job is served followed by all class-2 jobs.

Triage-Prioritize-Class-2 Policy (TP_2**)** Each job, with the exception of the last one, goes through triage in random order. If a job is classified as class-2, it is served right away; otherwise, the job is put aside to be served later. When the triage of $N - 1$ jobs is completed, the remaining untriaged job is served followed by all class-1 jobs.

There are a few important points that are worth mentioning. First, in TP_1 and TP_2, the server does not triage the last untriaged job since one can show that there is no benefit to triaging the last remaining untriaged job. Second, both TP_1 and TAF prioritize class-1 jobs. However, while TP_1 serves class-1 jobs as soon as they are identified, TAF starts serving class-1 jobs only after all jobs go through triage. Once the triage of all jobs is complete, we know from the $c\mu$-rule that the optimal action is to prioritize class-1 jobs. However, we do not know whether carrying out triage for all jobs first or following some other policy works better. And third, it might seem that given that the $c\mu$-rule favors class-1 patients, considering TP_2, which prioritizes class-2 patients, is not necessary. We will see however that while this policy is never the best policy among the four considered here, it can actually be preferable to TP_1 under certain conditions.

Let C_π denote the total expected cost under policy π. Because of the relatively simple structure of the policies described above, we can come up with closed-form expressions for C_π for each $\pi \in \{NT, TP_1, TP_2, TAF\}$. We refer the reader to Appendix A for the expressions and their derivations.

3.3.1 Comparison of benchmark policies

In the following proposition, we first identify two policies, which can never be the single best policy among the four.

Proposition 3.3.1. *(i) Triage-Prioritize-Class-1 policy always performs at least as well as Triage-All-First policy, i.e., $C_{TP_1} \leq C_{TAF}$. (ii) No-Triage policy always performs better than Triage-Prioritize-Class-2 policy, i.e., $C_{NT} < C_{TP_2}$.*

Proposition 3.3.1 is not unexpected. The total expected cost for class 2 jobs are the same under both policies, however, class 1 jobs will wait less under Prioritize-Class-1 Policy than under Triage-All-First Policy. By Proposition 3.3.1, in the remainder of our analysis, Triage-All-First Policy is eliminated.

Proposition 3.3.1(i) simply says that delaying the start of service until every single job is classified does not work well. This is because, once a job which has a high priority is identified, there is no point in delaying the service of that job. We know for sure that no other job will get a higher priority. Based on this result, in the following discussion, we will ignore TAF because it is always outperformed by TP_1. Part (ii) of the proposition says that skipping triage altogether and serving jobs in a random order always works better than triaging jobs while serving class-2 jobs as soon as they are identified. Although just like TAF, TP_2 can also be ignored when determining the best policy among the four policies described above, in the following, we will keep this policy under consideration since its analysis leads to some interesting insights.

Next, we compare TP_1 and NT, and thereby provide a complete prescription for finding the best policy among the four policies. First define $\alpha = \frac{h_2/\tau_2}{h_1/\tau_1}$ to be a measure of the relative "importance" of type-1 jobs over type-2 jobs. Note that $\alpha \in (0,1)$ by Assumption 3.2.1. If α is close to 0, type-1 jobs are far more important than type-2 jobs; if α is close to 1, there is no significant difference between the importance levels of the two types of jobs.

Proposition 3.3.2. *For any $p \in (0,1)$, $C_{TP_1} \leq C_{NT}$ if and only if $0 < \alpha \leq \beta(p)$, where*

$$\beta(p) = \max\left\{0,\ \frac{p\tau_1\left[\frac{N-2}{N}v_1 - 2\right]u + pq(v_1+v_2-1)\tau_1\tau_2}{q\tau_2\left[2 - \frac{N-2}{N}(1-v_2)\right]u + pq(v_1+v_2-1)\tau_1\tau_2}\right\}. \tag{3.1}$$

In other words, Triage-Prioritize-Class-1 policy performs better than No-Triage policy, and thus is the best policy among the four simple policies if α is sufficiently small for a given value of p; otherwise, No-Triage policy is the best policy.

Proposition 3.3.2 confirms the intuition that when the two types are sufficiently similar to each other - with regards to their importance - serving jobs randomly with no triage is superior to triaging them all (except for the last one). More specifically, the proposition gives a precise description of what we mean by two types of jobs being *sufficiently similar.* The following corollary immediately follows from Propositions 3.3.1 and 3.3.2.

Corollary 3.3.1. *Among the four policies, NT, TP_1, TP_2, and TAF, the best policy, i.e., the policy that minimizes the total expected cost, is TP_1 if $\alpha \leq \beta(p)$; otherwise, the best policy is NT.*

We can also show that the function $\beta(\cdot)$ possesses certain properties, which provide further insights into how the fraction of type-1 jobs in the population affects whether the differences between the two types of jobs would be significant enough to make TP_1 more preferable than NT.

Proposition 3.3.3. *$\beta(\cdot)$ is a quasi-concave function of p and is first non-decreasing then non-increasing over $(0,1)$. Thus, for each fixed $0 < \alpha < 1$, there is an interval $I(\alpha) = [\underline{p}(\alpha),\ \bar{p}(\alpha)]$, which is possibly an empty set and satisfies the following conditions:*

(i) If $p \in I(\alpha)$, TP_1 is better than NT; otherwise, NT is better.

(ii) $0 < \underline{p}(\alpha) < \bar{p}(\alpha) < 1$, where $\underline{p}(\alpha)$ is a non-decreasing and $\bar{p}(\alpha)$ is a non-increasing function of α, i.e., $I(\alpha)$ gets smaller as α increases.

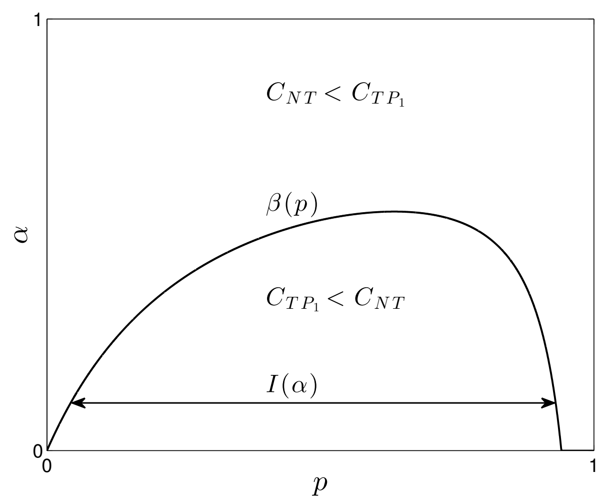

Figure 3.1: Comparison of Triage-Prioritize-Class-1 (TP_1) and No-Triage (NT) policies. ($u = 0.15,\ v_1 = 0.95,\ v_2 = 0.95,\ \tau_1 = 1,\ \tau_2 = 3,\ N = 100$).

Referring to Figure 3.1 might help the reader to better understand our results so far, particularly Propositions 3.3.2 and 3.3.3. As we can see from the figure, for TP_1 to be the best policy, the two job types should be sufficiently different. However, what is considered sufficiently different depends on p the true proportion of type-1 jobs in the population. For medium range values of p, i.e., when there is a good mixture of both types of jobs, neither type overwhelmingly dominating the other in numbers, it is relatively easier to meet the bar for being sufficiently different. But when p is small or large, either type-1 jobs are so rare that triage rarely ends up identifying a class-1 job or they are so dominant that triage rarely helps eliminate a class-2 job for immediate service. In any case, triage ends up being a waste of time, that is of course unless the two types are significantly different from each other as measured by α.

Propositions 3.3.3 shows that there is a unimodal curve that separates the Triage and No-Triage regions. We can strengthen this result further by showing that the benefit from triage (when there is) is smaller when one is close to the boundary described by $\beta(\cdot)$ and gets large as one moves away. More precisely, define the percentage improvement by triage as

$$\eta \equiv \frac{C_{NT} - C_{TP_1}}{C_{NT}} \times 100\%.$$

Proposition 3.3.4. *For any $\eta > 0$, there exists a unimodal curve $\beta(p, \eta)$ such that*

(i) $\frac{C_{NT}-C_{TP_1}}{C_{NT}} > \eta_0$ *if and only if* $\alpha < \beta(p, \eta_0)$ *for any given* η_0.

(ii) If $\eta_1 > \eta_2$, *then* $\beta(p, \eta_1) \leq \beta(p, \eta_2)$ *for all* $p \in [0, 1]$, *and*

(iii) $\beta(p,\eta) = \max\left\{0,\ \frac{p\tau_1\left[\frac{N-2}{N}v_1 - 2\right]u + pq(v_1+v_2-1)\tau_1\tau_2 - \eta p\tau_1[\frac{2}{N-1}\tau_1 + p\tau_1 + q\tau_2]}{q\tau_2\left[2 - \frac{N-2}{N}(1-v_2)\right]u + pq(v_1+v_2-1)\tau_1\tau_2 + \eta q\tau_2[\frac{2}{N-1}\tau_2 + p\tau_1 + q\tau_2]}\right\}.$

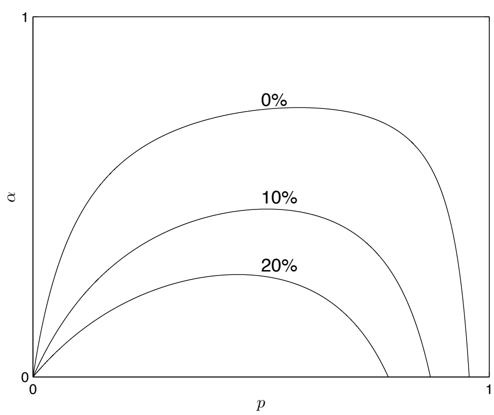

Figure 3.2: Comparison of Triage-Prioritize-Class-1 (TP_1) and No-Triage (NT) policies, different levels of η.

When $\eta = 0$, $\beta(p, 0) = \beta(p)$. Hence, Proposition 3.3.4 is consistent with Proposition 3.3.2. If we are interested in the cases that Policy T_1 can improve Policy NT by η, not simply which policy is better, then Proposition 3.3.4 says that there exists a unimodal curve $\beta(p, \eta)$ that divide the p-α plane into two parts and the area under $\beta(p, \eta)$ is the region where the percentage improvement by triage is at least η. Figure 3.2 illustrate the $\beta(p, \eta)$ when $\eta = 0\%, 10\%$ and 20%. The curve $\beta(p, 20\%)$

is below $\beta(p, 10\%)$, which implies that triage is more effective when there is significant difference between the two types of jobs, i.e. α is small.

3.3.2 Insights from comparison of policies TP_1 and TP_2

Our results provided a clear description of the conditions under which NT and TP_1 are the best policies among the four simple policies analyzed in this section. Clearly, there are many situations where skipping triage and serving jobs in random order is the best option. However, in many practical settings, because of unknown parameters such as p, it might be difficult to check whether or not the conditions are satisfied and as a result one might end up using a policy, which may or may not be optimal. Suppose for example that the service provider believes that all jobs should be triaged. (As many articles in the emergency response literature discuss, triage is performed in mass-casualty events despite the lack of any scientific evidence that it is actually beneficial.) The question then is whether priority should be given to class-1 jobs or class-2 jobs. More specifically, is it always true that $C_{TP_1} \leq C_{TP_2}$? One might be tempted to believe that based on the classical $c\mu$-rule result, the answer is yes and class-1 jobs should get a higher priority. After all, we know for a fact that if all jobs were already classified by time zero, the optimal action would have been to serve all class-1 jobs first. As we see in the following proposition, however, which only compares the policies TP_1 and TP_2, this intuitive argument is flawed.

Proposition 3.3.5. *(i) If $v_2 < 1/2 - \frac{Np\tau_1}{(N-2)u}(v_1 + v_2 - 1)$, then $C_{TP_1} < C_{TP_2}$ for all $\alpha \in (0, 1)$;*

(ii) if $v_1 < 1/2 - \frac{Nq\tau_2}{(N-2)u}(v_1 + v_2 - 1)$, then $C_{TP_1} > C_{TP_2}$ for all $\alpha \in (0, 1)$;

(iii) otherwise, $C_{TP_1} < C_{TP_2}$ if and only if $\alpha < \theta(p)$, where

$$\theta(p) = \min \left\{ 1,\ \frac{\frac{N-2}{N} p\tau_1(v_1 - 1/2)u + pq(v_1 + v_2 - 1)\tau_1\tau_2}{\frac{N-2}{N} q\tau_2(v_2 - 1/2)u + pq(v_1 + v_2 - 1)\tau_1\tau_2} \right\}. \tag{3.2}$$

Furthermore, $\theta(p) > \beta(p)$ for $p \in (0, 1)$.

Parts (i) and (ii) of Proposition 3.3.5 make it clear that v_1 and v_2, respective probabilities of correct classification for types 1 and 2, are important determinants of whether or not TP_1 is more preferable than TP_2. In particular, part (ii) states that if the correct classification of a type-1 job is sufficiently small, then TP_2 should be chosen over TP_1. However, misclassification is not the only reason why TP_2 can in fact be better than TP_1. Even if classification is perfect, i.e., $v_1 = v_2 = 1$, as we can see from part (iii) of Proposition 3.3.5, TP_2 is more preferable if $\alpha > \theta(p)$.

Before we explain why prioritizing less important jobs can in fact be better, first note the last statement of the proposition, which says that $\theta(p) > \beta(p)$ for $p \in (0, 1)$. This fact together with the rest of the proposition and Proposition 3.3.2 implies that TP_2 can be better than TP_1 only if NT is better than both TP_1 and TP_2. In other words, doing triage and prioritizing class-2 can be better than prioritizing class-1 only when triage is in fact a waste of time and it is better to serve jobs in random order without triage anyway. But in any case, the result shows that if the service provider makes the mistake of doing triage somehow believing that it *must surely* be beneficial, s/he can make things even worse if s/he further prioritizes class-1 jobs based on a flawed intuitive argument.

Now, why exactly is TP_2 better than TP_1 when the two types are not significantly different (α is large) and the proportion of type-1 jobs in the population, p, is small? Consider policy TP_1 when p is small. This policy will search for class-1 jobs to serve while leaving class-2 jobs till the end. The fact that p is small means the server will spend a lot of time triaging, very few jobs will be served right after triage, and many jobs will be left waiting to be served at the end once the triage process is completely over. Considering that α is large, i.e., the two types are not significantly different from each other, these long waiting times due to triage will not come with any tangible benefits. In contrast, consider policy TP_2. This policy will instead search for class-2 jobs to serve right away. Because fraction of type-2 jobs is large, it will take less time for the server to identify a class-2 job and serve right away. In many cases, triage will result in identification of a class-2 job, which will then be immediately served. This means that the number of jobs who will have to wait until all jobs are triaged and consequently the overall waiting time will be far less, which in the end makes the expected total cost less than it would be under TP_1. In short, TP_2 is better than TP_1 not because it is more beneficial to serve a class-2 job rather than a class-1 job at any given time, but because the relative importance of class-1 jobs over class-2 jobs does not justify lengthening waiting times as much as they would be under TP_1. It is simply more preferable to sacrifice relatively few important jobs so that a large number of jobs are served earlier.

3.3.3 Better triage better outcome?

Suppose that the service provider has the capability of improving triage accuracy possibly by more training, using an improved classification criteria, or using a more competent server. It is natural to expect that such an action should improve the outcome and in most cases it will. However, as we see in the following, this is not always true. The following proposition is an investigation into how the total expected cost under TP_1 changes with v_1 and v_2, the probabilities of correct classification for types 1 and 2, respectively.

Proposition 3.3.6. *(i) For all $\alpha \in (0,1)$ and $p \in (0,1)$, $\frac{\partial C_{TP_1}}{\partial v_1} < 0$, i.e., the expected cost under the Triage-Prioritize-Class-1 Policy always decreases with the probability of correctly classifying a type-1 job.*

(ii) Let $\gamma(p) = \frac{p\tau_1}{p\tau_1 + \frac{N-2}{N}u}$. For any fixed $p \in (0,1)$, if $\alpha > \gamma(p)$ then $\partial C_{TP_1}/\partial v_2 > 0$; i.e., if $\alpha > \gamma(p)$, then the expected cost under the Triage-Prioritize-Class-1 Policy increases with the probability of correctly classifying a type-2 job. Furthermore, $\gamma(p)$ is an increasing function of p and $\beta(p) < \gamma(p)$ for any $p \in (0,1)$.

Part (i) of Proposition 3.3.6 is intuitive. It simply says that an improvement in triage, which results in a higher probability of correct classification for type-1 jobs, also improves the performance of TP_1. On the other hand, part (ii), which says that, in some cases an improvement in the correct classification probability of type-2 jobs worsens the performance of the policy, is not as intuitive. It is true that TP_1 does not aim to prioritize type-2 jobs, but regardless, a higher probability of correct classification means a better way of sorting out the two types. Given this fact, why should a higher value of v_2 not always help?

The answer again lies in the somewhat indirect operational implications of correct (and incorrect) classification of jobs and is similar to the one that explains why TP_2 can sometimes be better than TP_1. First note that an increase in v_2 can only hurt when $\alpha > \gamma(p) > \beta(p)$, which is the case where performing triage is not really helpful in the first place and NT policy is the best, see Figure 3.1. Therefore, any change that will alleviate the negative effects of triage will be of help. Now, $\alpha > \gamma(p)$ means that the two types are not significantly different from each other and/or the proportion of type-1 jobs is small. Thus, under policy TP_1, significant time will be spent on identifying (possibly incorrectly) type-1 jobs, and a high percentage of jobs (all class-2 jobs) will have to wait the triage of all jobs to be over. It would actually be better if some of these jobs, if not all, were served before, right after they were triaged even if they belong to class-2. This is exactly what would happen if more of the type-2 jobs were misclassified as class-1 and therefore a decrease in correct classification probability for type-2 jobs helps.

3.4 State-dependent policies

In the previous section, we restricted ourselves to four policies, which are practically appealing because of their simplicity. In this section, we make no such restriction and concentrate on identifying the policy that is optimal within the whole class of state-dependent policies (which also includes state-independent policies). We first develop a Markov decision process (MDP) formulation for the problem described in Section 3.4.1 and then provide a complete description of the optimal policy.

3.4.1 Markov decision process formulation

The decision epochs are time zero, and triage and service completion times for the server (since we assume the service is non-preemptive). The state of the system can then be denoted by the triplet (i, k_1, k_2), where i represents the number of *untriaged* jobs, and k_1 and k_2 denote the number of jobs that have been classified as class-1 and class-2 but not yet served, respectively. Since we have N jobs in total, the state space can be described as $\mathcal{S} = \{(i, k_1, k_2) : i, k_1, k_2 \geq 0, i + k_1 + k_2 \leq N\}$.

Using a sample-path argument, it is straightforward to show that keeping the server idle is suboptimal. This allows us to ignore idling as an admissible action. Then, in a given state $s = (i, k_1, k_2)$, the available actions for the server are **SU:** serve an untriaged job without triage (only available if $i \geq 1$); **Tr:** triage an untriaged job (only available if $i \geq 1$); **SC1:** serve a class-1 job (only available if $k_1 \geq 1$); and **SC2:** serve a class-2 job (only available if $k_2 \geq 1$).

Our objective is to minimize the total expected cost. In general it is possible that there are more than one optimal action for any given state. If that is the case, we choose the action that is listed earlier in the action set $\{$SC1, SU, Tr, SC2$\}$. For instance, SC1 has precedence over all the other actions.

While this assumption is not crucial, it allows us to ensure that there is a unique optimal policy, which in turn simplifies the presentation of the results.

We define $a^*(s)$ for $s \in \mathcal{S}$ to be the optimal action in state s. We also let $V_\pi(i, k_1, k_2)$ denote the total expected cost under policy π and $V(i, k_1, k_2) = \min_\pi \{V_\pi(i, k_1, k_2)\}$ to be the total expected cost under an optimal policy starting from state (i, k_1, k_2) with no service or triage in progress.

Table 3.1: Notation used in writing optimality equations in a compact form.

	Untriaged job	Class-1 job	Class-2 job
Expected cost rate	$r = ph_1 + qh_2$	$r_1 = \frac{pv_1h_1+q(1-v_2)h_2}{PC_1}$	$r_2 = \frac{p(1-v_1)h_1+qv_2h_2}{PC_2}$
Expected service time	$T = p\tau_1 + q\tau_2$	$T_1 = \frac{pv_1\tau_1+q(1-v_2)\tau_2}{PC_1}$	$T_2 = \frac{p(1-v_1)\tau_1+qv_2\tau_2}{PC_2}$
Expected service cost	$c = ph_1\tau_1 + qh_2\tau_2$	$c_1 = \frac{pv_1h_1\tau_1+q(1-v_2)h_2\tau_2}{PC_1}$	$c_2 = \frac{p(1-v_1)h_1\tau_1+qv_2h_2\tau_2}{PC_2}$

Let PC_i denote the probability of classifying a random job as class-i for $i = 1, 2$ so that $PC_1 = pv_1 + q(1 - v_2)$ and $PC_2 = p(1 - v_1) + qv_2$. Then, using the notation in Table 3.1 we can write the optimality equations as follows:

$$
\begin{aligned}
V(i, k_1, k_2) = \min \Big\{ & PC_1 V(i-1, k_1+1, k_2) + PC_2 V(i-1, k_1, k_2+1) + (ir + k_1r_1 + k_2r_2)u \\
& V(i-1, k_1, k_2) + c + [(i-1)r + k_1r_1 + k_2r_2]\, T, \\
& V(i, k_1-1, k_2) + c_1 + [ir + (k_1-1)r_1 + k_2r_2]\, T_1, \\
& V(i, k_1, k_2-1) + c_2 + [ir + k_1r_1 + (k_2-1)r_2]\, T_2 \Big\}, \ \forall\, (i, k_1, k_2) \in S \setminus (0,0,0), \\
V(0,0,0) = 0, \text{ and } & V(s) = \infty, \ \forall s \notin S.
\end{aligned} \tag{3.3}
$$

3.4.2 Complete characterization of the optimal policy

We start by describing when SC1 and SC2 are optimal actions.

Theorem 3.4.1. *Consider state* $(i, k_1, k_2) \in \mathcal{S}$*:*

(i) If $k_1 \geq 1$*, then* $a^*(i, k_1, k_2) = \text{SC1}$*, i.e., as soon as the server identifies a class-1 job, that job should be served next.*

(ii) If $i + k_1 > 0$*, then* $a^*(i, k_1, k_2) \neq \text{SC2}$*, i.e., it is optimal to serve a class-2 job only when there are no untriaged or class-1 jobs.*

Theorem 3.4.1 clearly delineates the regions where serving jobs classified as class-1 and class-2 are optimal. Specifically, SC1 has precedence over all other actions no matter what the current state is. This means that as soon as a triage results in identification of a class-1 job, the next action is to serve that job. On the other hand, SC2 is at the bottom of the priority list meaning that the service of class-2 jobs starts at the end when there are no more class-1 or untriaged jobs waiting.

Given Theorem 3.4.1, to characterize the optimal policy completely, it now remains to study the states where there are no class-1 jobs, i.e., $k_1 = 0$, but there is at least one untriaged job, i.e., $i \geq 1$. Recall that in such a state, the server can choose to either triage or directly serve an untriaged job. We know that serving a class-2 job, if there is one, is suboptimal. It turns out that whether or not doing triage is optimal depends on the system state. More specifically, there is a line that separates the states in which doing triage is optimal from the states in which serving without triage is optimal. With the following theorem, we not only prove this structural property of the optimal policy but also provide a complete expression for this line.

Theorem 3.4.2. *There exists a linear function $L(\cdot)$ such that for any state $(i, 0, k_2) \in \mathcal{S}$ where $i \geq 1$ and $k_2 \geq 0$, if $k_2 \geq L(i)$, then $a^*(i, 0, k_2) =$ SU, i.e., the optimal action is to serve without triage; otherwise, $a^*(i, 0, k_2) =$ Tr, i.e., the optimal action is to perform triage. Furthermore,*

$$L(i) = \left(\frac{r(\tilde{u} - u)}{r_2 u}\right) i - \frac{r\tilde{u}}{r_2 u}, \tag{3.4}$$

where $\tilde{u} = PC_2(rT_2 - r_2T)/r$.

Figure 3.3 is a visual demonstration of Theorem 3.4.2 for a specific example. To better understand the intuition behind Theorem 3.4.2, first note that the condition $k_2 \geq L(i)$ can equivalently be written as $(ir + k_2r_2)u \geq (i - 1)PC_2(rT_2 - r_2T)$, where the left-hand side is the total expected additional cost that would be incurred by performing $triage$ in state $(i, 0, k_2)$ and one can show that the right-hand side is the total expected cost reduction that would be achieved as a result of having performed triage in state $(i, 0, k_2)$ and then skipping triage for all the remaining jobs. More specifically, we can show that the optimal policy for deciding whether or not to do triage is a *one-stage look-ahead policy*, i.e., it is optimal to stop doing triage in a given state if skipping triage for all the remaining jobs is at least as good as performing triage one last time and stopping immediately after.

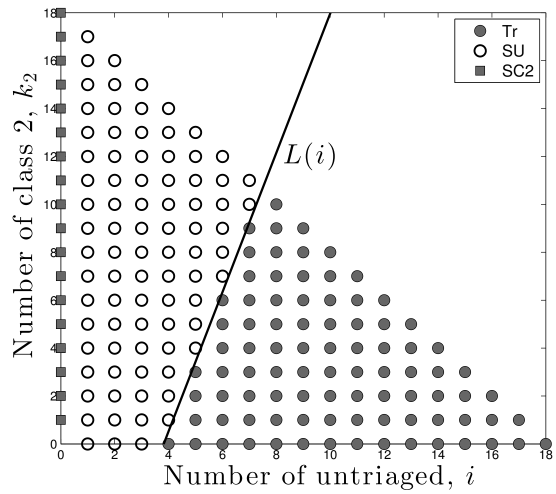

Figure 3.3: Visual description of the optimal policy when $k_1 = 0$ and $h_1 = 10, h_2 = 1, \tau_1 = 1, \tau_2 = 2, v_1 = 1, v_2 = 0.95, u = 0.26, p = 0.8, N = 18$.

Theorem 3.4.2 provides interesting insights into the decision of when to do triage and when to skip it. If N is large, meaning that there are too many jobs waiting to be served and we have no information regarding which ones are more important, one might be tempted to skip triage since performing triage will further lengthen the waiting times, which are already likely to be too long. With too many jobs to serve, spending time on triage might seem like an unwise use of time. In contrast, when N is small, triage might not seem all that harmful since waiting times are not going to be too long even with triage. As we explain in the following, however, this reasoning is flawed.

Theorem 3.4.2 states that - as one can also easily verify referring to Figure 3.3 - when the number of untriaged jobs is sufficiently large (initially more than or equal to 4 for the example whose solution is depicted in the figure) it is optimal to start with triage and continue to do so as long as the number of untriaged jobs and the number of class-2 jobs keep the state space under the line. (Note that if a class-1 job is identified, that job is served right away.) Once the threshold line is passed, the optimal policy starts serving jobs without triage until there are no more untriaged jobs waiting. Class-2 jobs, which were identified earlier, are served at the end. If the number of untriaged jobs is small (initially less than 4 in the example), then the optimal policy is simply to serve all the jobs without triage. Thus, contrary to the argument above, precisely because there are too many jobs, one cannot afford to skip triage. Even if triage is skipped, service will take quite a long time anyway. Therefore, it makes sense to spend some time at the beginning (specifically as long as the system state is to the right of the threshold line) to perform triage in an effort to at least prevent the waiting times for important jobs getting too long. On the other hand, when there are few jobs, service of all jobs including those of type-1, will not take too much time. Therefore, the value of class information that will be obtained through triage does not justify the additional waiting that all jobs will have to endure.

Finally, in this section, we investigate conditions under which the optimal policy turns out to be one of the simple benchmark policies investigated in Section 3.3. It would be natural to expect that under the optimal policy, when the expected triage time is sufficiently short (it might help to think of the limiting case where it is zero) all jobs would go through triage and when the expected triage time is sufficiently long no job would go through triage. Indeed, we can prove that is the case. The following proposition clearly describes what would qualify as sufficiently short and what would qualify as sufficiently long.

Proposition 3.4.1. *Let* $u_1 = \frac{N-1}{N}\tilde{u}$ *and* $u_2 = \frac{r}{2r+(N-2)r_2}\tilde{u}$. *Then,*

(i) the optimal policy is NT*, i.e., No-Triage policy, if and only if* $u \geq u_1$*;*

(ii) the optimal policy is TP_1*, i.e., Triage-Prioritize-Class-1 policy, if and only if* $u \leq u_2$.

When the expected triage time is as long as described in Proposition 3.4.1(i), the information that one would get through triage is simply not worth it. Hence, the optimal policy is to serve all jobs directly without triage. When the expected triage time is as short as described in Proposition 3.4.1(ii), one can "afford" to triage all the jobs no matter what types of jobs are identified during the triage process; however, in line with Theorem 3.4.1, if a class-1 job is identified as a result of triage, that job should be served first before moving on to the triage of the remaining jobs.

3.5 Numerical study: performance comparison of the proposed policies under linear and non-linear waiting costs

This section mainly consists of two parts. In the first part, we compare the performance of the optimal policy with the performances of the benchmark policies, specifically NT and TP_1, so as to understand whether there is significant benefit to using the optimal policy as opposed to benchmark policies, which are in most cases simpler and easy-to-use. In this first part of the study, we assume that waiting costs are linear functions of time as are assumed in our mathematical model. In the second part, based on our analytical results given in Section 3.4, we first devise heuristic methods that can be used when waiting costs are not linear. Then, we compare the performances of these heuristic methods with those of the optimal policy as well as the simple benchmark policies.

3.5.1 Performance comparison when waiting costs are linear in time

For this study, we considered a system with $N = 50$ jobs, all untriaged at time zero. We chose p from the set $\{0.1, 0.3, 0.5, 0.7, 0.9\}$, and for each value of p, we generated 2,000 scenarios by randomly and uniformly choosing u between 0 and 1, τ_1 and τ_2 between 0 and 10, h_1 between 0 and 4, h_2 between 0 and 1, v_1 and v_2 between 0.5 and 1; discarding cases for which $h_1/\tau_1 < h_2/\tau_2$. For each scenario, we obtained the total expected cost under the optimal policy, No-Triage policy (NT), and Triage-Prioritize-Class-1 policy (TP_1), and computed the percentage improvement in the total expected cost that one would get by using the optimal policy as opposed to each one of the benchmark policies NT and TP_1. Then, we constructed 95% confidence intervals for the mean percentage improvement as well as the maximum percentage improvement. (The confidence interval for the maximum percentage improvement was obtained by putting the 2,000 scenarios in groups of size 10 and determining the maximum within each group, which results in a total of 200 observations.) The results are provided in Table 3.2.

Table 3.2: 95% confidence intervals for the mean and maximum percentage improvement in the total expected cost by using the optimal policy as opposed to benchmark policies.

p	No-Triage Policy		Triage-Prioritize-Class-1 Policy	
	Mean	Maximum	Mean	Maximum
0.1	6.42 ± 0.46	27.96 ± 1.96	8.26 ± 0.52	31.32 ± 2.30
0.3	9.25 ± 0.51	30.83 ± 1.60	5.90 ± 0.43	25.38 ± 2.10
0.5	8.90 ± 0.51	31.78 ± 1.63	5.81 ± 0.42	24.50 ± 1.97
0.7	4.73 ± 0.35	20.51 ± 1.57	6.88 ± 0.45	27.10 ± 2.17
0.9	1.05 ± 0.15	7.33 ± 1.09	11.49 ± 0.56	36.38 ± 2.18

Table 3.2 clearly shows that there can be significant benefits to using the optimal policy as opposed to any one of the benchmark policies. Specifically, the mean percentage improvement can be more than 10% while the mean maximum improvement can be more than 35% depending on the value of p. As we have shown in Section 3.3, when p, the probability of a random job being type-1, is close to

zero or 1, NT is the best benchmark policy for a large range of values of α (it performs particularly well when p is close to 1) and thus it is no surprise that its performance is closer to that of the optimal policy for such values of p. The performance gap is more significant for mid-range values of p. When comparing the performances of the optimal policy and TP_1, we observe the opposite. TP_1 performs relatively better for mid-range values of p. This is not surprising. NT and TP_1 can be seen as at the two ends of the policy spectrum with the former skipping triage altogether and the latter performing triage for all the jobs. Thus when jobs are highly dominated by one type, NT tends to perform better since triage does not bring much benefit; when there is a good mixture of both types, TP_1 tends to perform better. The optimal dynamic policy hits the "right" balance between these two policies by choosing to triage or skip it depending on the system state.

3.5.2 Performance comparison when waiting costs are non-linear in time

One of the assumptions we made for our mathematical analysis was that the cost of keeping the jobs waiting is linear in time. There are, however, situations where this assumption would not be reasonable. Our goal in this section is to propose new heuristic methods based on our analysis under the linear waiting cost assumption and test how these methods perform in comparison with other benchmark heuristics when waiting costs are non-linear in time. In particular, we consider three different forms for the waiting cost function: increasing convex, increasing concave, and increasing convex-concave (S-shaped). In the following, we will use $f_1(\cdot)$ and $f_2(\cdot)$ to denote the waiting cost functions for types 1 and 2, respectively, i.e., $f_i(t)$ is the total cost that would be incurred by a type-i job that has waited for t time units in the system.

Proposed heuristic methods to be used when waiting costs are non-linear

We propose three heuristic methods:

(i) **Fixed Threshold-$c\mu$ policy (FT-$c\mu$):** We fit a least-squares line to each cost function and use the optimal dynamic policy as prescribed in Section 3.4 assuming that cost functions are these fitted functions. When fitting the least-squares line, we assume that the non-linear waiting cost function is defined over the interval $[0, N(\tau + u)]$, where the right end-point corresponds to the expected time the system would be cleared of all jobs if each job were to go through triage. We name the heuristic Fixed Threshold-$c\mu$ policy because (i) whether or not triage is carried out in a given state is determined by where the state lies with respect to the threshold line, which is fixed at time zero, and (ii) class-1 (high priority) and class-2 (low priority) are determined according to the expected version of the $c\mu$-rule. Note that the c term here is calculated using the slopes of the fitted lines for each type.

(ii) **Dynamic Threshold-$c\mu$ policy (DT-$c\mu$):** We fit a least-squares line to each cost function and, as in the FT-$c\mu$ policy, use the slopes of these lines to determine the high priority class and the low priority class with respect to the $c\mu$-rule. Unlike the FT-$c\mu$ policy, however, this policy updates the threshold line to be used for determining whether or not triage should be done by fitting new least-squares lines over the interval $[t_{\text{now}}, t_{\text{now}}+i(\tau+u)+k_1\tau_1+k_2\tau_2]$ where t_{now} is the current time, i.e., the time decision is to be made, and $t_{\text{now}} + i(\tau + u) + k_1\tau_1 + k_2\tau_2$ is the expected time the system would be cleared of all jobs if each remaining untriaged job were to go through triage. The time-dependent threshold line $L_t(\cdot)$ is obtained by using (3.4) but replacing h_1 and h_2 with the slopes of the two lines fitted to $f_1(\cdot)$ and $f_2(\cdot)$, respectively.

(iii) **Dynamic Threshold-$Gc\mu$ policy (DT-$Gc\mu$):** This policy updates the threshold line exactly the same way the DT-$c\mu$ policy does. However, when determining the priorities between the two classes, it uses the Generalized-$c\mu$ rule developed by Van Mieghem (1995). Specifically, this is how this heuristic works: Let $h_1(\cdot)$ and $h_2(\cdot)$ denote the derivative functions for $f_1(\cdot)$ and $f_2(\cdot)$, respectively. Consider a decision epoch where the system is in state (i, k_1, k_2). First, the high priority class is determined by comparing $h_1(t)/\tau_1$ with $h_2(t)/\tau_2$. If the former is larger, give class-1 higher priority; otherwise, give class-2 higher priority. If class-j is given higher priority, then the action to be taken is serving class-j whenever $k_j > 0$. If $k_j = 0$ and $i > 0$, then an untriaged job is triaged if $k_{3-j} \leq L_t(i)$, where $L_t(i)$ is obtained by using (3.4) but replacing h_1 and h_2 with $h_j(t)$ and $h_{3-j}(t)$, respectively. If $k_{3-j} > L_t(i)$, an untriaged job is directly served without triage. Finally, when $i = k_j = 0$, the action to be taken is serving class $3 - j$.

Numerical experiments and results

In our numerical study, we mainly considered three different scenarios each differing in the general structure for the waiting cost functions assumed. For each pair of cost function choices, we generated scenarios as follows: we assumed that initally there were 20 jobs all untriaged. The expected triage time was assumed to be 0.5 units. The expected service time τ was assumed to be the same for both types and was chosen from the set $\{1, 5, 10\}$. The probability of a random job being of type 1, p, was chosen from the set $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. For each pair of τ and p values chosen, 200 scenarios were randomly generated by choosing both v_1 and v_2 uniformly between 0.5 and 1. Triage times and service times were assumed to be deterministic so as to make it possible to obtain the optimal policy and compare its performance with those of the heuristic methods.

Convex increasing waiting cost functions: As discussed in detail in Van Mieghem (1995), for various reasons including customer expectations and the psychology of waiting, in certain settings a convex function that penalizes waits with an increasing rate might be a better fit. To investigate how the proposed methods might work in such settings, we assumed that $f_1(t) = \left(\frac{t}{210}\right)^2$ and $f_2(t) = \frac{1}{4}\left(\frac{t}{210}\right)^2$ (see the leftmost plot in Figure 3.4).

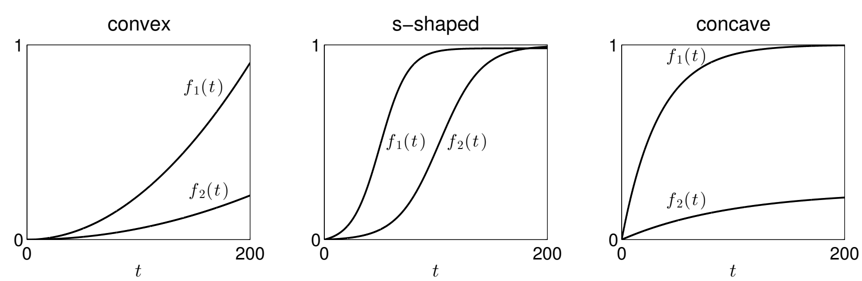

Figure 3.4: The waiting cost functions assumed in the numerical study.

Table 3.3 reports the 95% confidence intervals for the percentage increase in the total expected cost if one uses the heuristic methods as opposed to the optimal policy. Numbers close to zero indicate performances close to optimal whereas large numbers indicate poor performance. As we can observe from the table all of the three policies we propose, FT-$c\mu$, DT-$c\mu$, and DT-$Gc\mu$, perform well in all the settings with the percentage deviation from the optimal policy being very close to zero in almost all cases. When $\tau = 1$, i.e., the expected triage time is large relative to the expected service time, No-Triage policy, is optimal, but so are the three policies we propose, which reduce to No-Triage policy in that parameter region. However, No-Triage policy performs poorly when the expected service time is significantly larger than the expected triage time. We also observe that the benchmark policies TP_1 and TP_2 perform badly across almost all scenarios.

Table 3.3: Performance comparison for the convex cost case - 95% confidence interval for the mean percentage increase in total expected cost when compared with that under the optimal policy.

τ	p	FT-$c\mu$	DT-$c\mu$	DT-$Gc\mu$	NT	TP_1	TP_2
1	0.1	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	163.58 ± 1.59	198.47 ± 1.80
1	0.3	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	133.82 ± 2.85	229.68 ± 2.94
1	0.5	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	128.28 ± 3.20	231.57 ± 2.92
1	0.7	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	133.95 ± 3.04	222.21 ± 2.41
1	0.9	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	146.26 ± 2.57	207.96 ± 1.76
5	0.1	0.12 ± 0.03	0.12 ± 0.03	0.12 ± 0.03	0.65 ± 0.16	15.01 ± 0.60	46.91 ± 1.17
5	0.3	0.26 ± 0.03	0.26 ± 0.03	0.26 ± 0.03	6.20 ± 0.89	8.06 ± 0.66	68.00 ± 3.03
5	0.5	0.14 ± 0.02	0.14 ± 0.02	0.14 ± 0.02	5.48 ± 0.88	7.38 ± 0.74	63.39 ± 2.72
5	0.7	0.03 ± 0.01	0.03 ± 0.01	0.03 ± 0.01	1.31 ± 0.34	9.80 ± 0.89	48.04 ± 1.29
5	0.9	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	18.76 ± 0.61	35.86 ± 0.35
10	0.1	0.29 ± 0.03	0.29 ± 0.03	0.29 ± 0.03	4.20 ± 0.48	4.37 ± 0.25	36.41 ± 1.58
10	0.3	0.19 ± 0.02	0.19 ± 0.02	0.19 ± 0.02	14.64 ± 1.40	2.15 ± 0.19	62.14 ± 3.57
10	0.5	0.11 ± 0.01	0.11 ± 0.01	0.11 ± 0.01	13.83 ± 1.42	1.86 ± 0.21	56.86 ± 3.27
10	0.7	0.07 ± 0.01	0.07 ± 0.01	0.07 ± 0.01	6.68 ± 0.84	2.34 ± 0.31	37.98 ± 1.84
10	0.9	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.06 ± 0.04	6.19 ± 0.44	19.64 ± 0.28

S-shaped (convex-concave) increasing waiting cost functions: In the case of emergency response or search and rescue operations, jobs may correspond to injured individuals whose survivals are at stake. With passage of time, the survival probabilities of these individuals decline. In many cases, the way these survival probabilities decline with time has an inverse S-shape with the survival probabilities declining with a rate that is slow at the beginning but gradually getting faster but eventually getting slow again when the survival probabilities get closer to zero (Sacco et al., 2005). This corresponds to a waiting cost function, which has an S-shape with a convex increasing portion at the beginning followed by a concave increasing portion. To investigate how the proposed methods work when waiting cost functions have such a structure, we assumed that $f_1(t) = \frac{1}{1+e^{-6t/75+4}} - \frac{1}{1+e^4}$ and $f_2(t) = \frac{1}{1+e^{-4t/75+5.5}} - \frac{1}{1+e^{5.5}}$. These two functions are plotted in the middle graph in Figure 3.4. Both $f_1(\cdot)$ and $f_2(\cdot)$ are bounded by 1 so in the context of emergency response operations they can be interpreted as the decline in the survival probability of a particular patient with the passage of time. Type-1 jobs are in more serious condition than type-2 jobs since at any particular point in time, the survival probability for type-1 jobs is smaller than the survival probability for type-2 jobs, i.e., waiting cost function for type-1 jobs is above the waiting cost function for type-2 jobs at all times. However, note that the rate of increase in the waiting cost is not always higher for type-1 jobs. Their costs increase with a rate that is higher than that for type-2 jobs initially but once their cost gets sufficiently close to 1, the rate for type-2 jobs gets higher.

Table 3.4: Performance comparison for the S-shaped cost case - 95% confidence interval for the mean percentage increase in total expected cost when compared with that under the optimal policy.

τ	p	FT-$c\mu$	DT-$c\mu$	DT-$Gc\mu$	NT	TP_1	TP_2
1	0.1	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	100.99 ± 2.21	183.80 ± 2.68
1	0.3	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	85.97 ± 3.38	213.33 ± 3.61
1	0.5	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	91.53 ± 3.40	207.98 ± 3.08
1	0.7	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	104.40 ± 2.98	194.10 ± 2.28
1	0.9	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	121.57 ± 2.30	177.74 ± 1.52
5	0.1	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	24.13 ± 0.57	36.46 ± 0.53
5	0.3	1.23 ± 0.19	1.22 ± 0.20	1.09 ± 0.19	2.15 ± 0.43	11.11 ± 0.54	38.09 ± 1.21
5	0.5	1.71 ± 0.26	1.70 ± 0.27	1.22 ± 0.19	2.36 ± 0.43	9.31 ± 0.36	36.45 ± 1.41
5	0.7	0.65 ± 0.17	0.57 ± 0.14	0.36 ± 0.09	0.58 ± 0.15	9.51 ± 0.27	29.32 ± 0.90
5	0.9	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	10.93 ± 0.18	21.35 ± 0.38
10	0.1	0.11 ± 0.03	0.01 ± 0.00	0.04 ± 0.01	0.11 ± 0.03	9.19 ± 0.10	3.60 ± 0.18
10	0.3	2.99 ± 0.27	0.71 ± 0.08	1.30 ± 0.13	2.99 ± 0.27	13.48 ± 0.51	3.05 ± 0.16
10	0.5	4.72 ± 0.38	2.04 ± 0.25	2.02 ± 0.18	4.72 ± 0.38	15.62 ± 0.72	4.51 ± 0.30
10	0.7	4.05 ± 0.31	2.72 ± 0.35	1.53 ± 0.14	4.05 ± 0.31	12.66 ± 0.56	6.12 ± 0.45
10	0.9	1.33 ± 0.10	0.95 ± 0.12	0.26 ± 0.02	1.33 ± 0.10	5.79 ± 0.16	5.72 ± 0.32

The results are given in Table 3.4. When the expected service time is short meaning that triage times are relatively long, No-Triage policy turns out to be optimal along with all three policies we are proposing. As the expected service time gets longer, we start seeing some differences among the four policies. First of all, No-Triage policy is no longer optimal even though its performance is still very reasonable and very close to that of the FT-$c\mu$ policy. In fact, the FT-$c\mu$ policy simplifies to the No-Triage policy in most cases. In some cases, FT-$c\mu$ outperforms the No-Triage policy but nevertheless it is still difficult to make a strong argument that FT-$c\mu$ would be a better choice than No-Triage considering the simplicity of the latter policy. However, the performances of both DT-$c\mu$ and DT-$Gc\mu$ are superior to those of FT-$c\mu$ and the No-Triage policy in all the settings where the expected service time is large relative to the expected triage time and very close to that of the optimal policy in all the settings. In particular, the worst performance of DT-$Gc\mu$ is observed when $p = 0.5$ and $\tau = 10$ and even then the mean percentage difference from the optimal policy is only about 2%. On the other hand, the worst performance of DT-$c\mu$ is slightly worse than that. It is not possible to designate either DT-$c\mu$ or DT-$Gc\mu$ as the "best" since there are many settings in which their performances are not statistically different.

Concave increasing waiting cost functions: In some cases, concave functions can more appropriately capture the reality. For example, consider an emergency response situation which leads to S-shaped waiting cost functions as discussed above. Now, suppose that due to the delays in response, which might have various reasons, the service process starts long after the incident occurred and as a result we are faced with only the concave increasing portion of the waiting cost function. To investigate how our policies would perform in such settings, we assumed that $f_1(t) = 1 - e^{(-0.03)t}$ and $f_2(t) = (1 - e^{(-0.01)t})/4$, which are both concave increasing functions (see the rightmost graph in Figure 3.4 for plots of these two functions).

Table 3.5: Performance comparison for the concave cost case - 95% confidence interval for the mean percentage increase in total expected cost when compared with that under the optimal policy.

τ	p	$FT\text{-}c\mu$	$DT\text{-}c\mu$	$DT\text{-}Gc\mu$	NT	TP_1	TP_2
1	0.1	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	44.99 ± 0.98	72.43 ± 0.92
1	0.3	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	34.01 ± 1.29	77.40 ± 1.27
1	0.5	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	34.79 ± 1.18	75.39 ± 1.24
1	0.7	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	38.34 ± 0.99	71.42 ± 1.11
1	0.9	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	42.85 ± 0.79	66.50 ± 0.92
5	0.1	0.16 ± 0.02	0.28 ± 0.04	0.28 ± 0.04	1.82 ± 0.31	2.50 ± 0.22	15.32 ± 0.57
5	0.3	0.08 ± 0.01	0.22 ± 0.02	0.22 ± 0.02	4.43 ± 0.52	0.93 ± 0.13	20.49 ± 1.02
5	0.5	0.05 ± 0.01	0.10 ± 0.01	0.10 ± 0.01	2.62 ± 0.33	0.84 ± 0.14	17.77 ± 0.84
5	0.7	0.03 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.52 ± 0.11	1.35 ± 0.17	13.44 ± 0.53
5	0.9	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	3.38 ± 0.13	9.91 ± 0.30
10	0.1	0.30 ± 0.03	0.40 ± 0.05	0.42 ± 0.05	1.73 ± 0.23	0.90 ± 0.07	6.40 ± 0.29
10	0.3	0.24 ± 0.01	0.30 ± 0.03	0.29 ± 0.03	3.12 ± 0.30	0.42 ± 0.03	8.90 ± 0.45
10	0.5	0.25 ± 0.02	0.19 ± 0.01	0.16 ± 0.01	2.03 ± 0.18	0.40 ± 0.02	7.85 ± 0.39
10	0.7	0.21 ± 0.02	0.10 ± 0.01	0.06 ± 0.01	0.75 ± 0.08	0.42 ± 0.03	5.96 ± 0.31
10	0.9	0.01 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.76 ± 0.04	3.81 ± 0.18

The results are provided in Table 3.5. When the expected service time is small meaning that the expected triage time is relatively large, all three policies we propose reduce to No-Triage policy, which turns out to be the optimal policy. For mid to large values of the expected service time, the No-Triage policy is no longer optimal, it is outperformed by all three policies we propose, whose performances are very close to that of the optimal policy. It is however difficult to pick the "best" among the three since in some parameter settings none of the policies has a statistically superior performance while in others there is not a single policy which stands out. Regardless, the performances of all three policies are so close to each other that even if one picks a policy that turns out to be not optimal, the performance difference would in all likelihood be very small.

CHAPTER 4: PRIORITY SCHEDULING OF JOBS WITH HIDDEN TYPES IN A QUEUEING SYSTEM

4.1 Introduction

In this chapter, we study the information/delay trade-off in a setting where there are external arrivals to the system. Jobs, which arrive at the system, are of unknown type, and similar to the model in Chapter 3, the server has the option to spend time to extract the type information for each job. The queueing model in this chapter incorporates the feature that in an MCI, injuries may arrive or be transported to the triage/treatment field. Our objective is to study the optimal dynamic policy on whether to triage or skip triage in order to minimize the long-run average cost.

The remainder of this chapter is organized as follows: Section 4.2 gives a detailed description of the formulation. Section 4.3 shows that there exists a threshold type policy that is optimal, and describes how we prove this results. A simulation study is carried out in Section 4.4. We observe that a heuristic policy we propose, which mimics the properties of the optimal policy, can bring significant improvement over simple and easy-to-implement policies.

4.2 Model description

Consider a service system with a single server and two types of jobs, type 1 and type 2. Jobs arrive at the system according to a Poisson process with rate λ, and they wait in a queue if they are not served upon arrival. The waiting space is unlimited. For convenience, we use "class 0" or "untriaged job" to refer to these new jobs that have not gone through triage. Each job belongs to type 1 with probability p and to type 2 with probability $q \equiv 1 - p$. Both p and q are exogenous parameters, and will not change over time. A type i job incurs a cost with rate h_i per unit time the job stays in the system where $i = 1,\ 2$. The service time of a job from type i is exponentially distributed with mean $\tau > 0$. Without loss of generality, we assume that type 1 jobs are important than type 2 jobs from the service provider's perspective in the sense of higher cost rate, i.e., $h_1 > h_2$.

We assume that the type information of each new job is hidden from the service system initially, i.e., the server does not know the exact type of a new arrival. The server can serve a job without knowing its type, but s/he also has the option to spend some time on investigation to obtain the type information of a job, and classify the job as class 1 or class 2. The investigation time of a job is exponentially distributed with mean $u > 0$, independent of the arrival process and the job's type. We denote class 1 as the important class, and class 2 as the less important class. The server tries her/his best to classify the type 1 jobs into class 1, and type 2 jobs into class 2. While the investigation on a job provides information on the job's type, the classification is error-prone. Define v_1 as the probability of classifying a type 1 job into class 1 and v_2 as the probability of classifying a type 2 job into class 2. Without loss of generality, we assume that $v_1 + v_2 > 1$. Denote PC_i as the probability of classifying a random job into class i, where $i = 1, 2$. Then, $PC_1 = pv_1 + q(1 - v_2),\ PC_2 = p(1 - v_1) + qv_2$.

We further assume that a preemptive discipline is used and there is no cost or changeover time for the server to switch actions. Let $x_j(t)$ denote the number of jobs in class j at time t where $j = 0, 1, 2$, then $X(t) = (x_0(t), x_1(t), x_2(t))$ is the current state of the system. Hence, the state space is $\mathcal{S} = \{(i, k_1, k_2) : i \geq 0, k_1 \geq 0, k_1 \geq 0\}$. At any time, the provider of the service system can take one of the following four actions: **SU** – serve an untriaged job without triage (only available if $i \geq 1$); **Tr** – triage an untriaged job (only available if $i \geq 1$); **SC1** – serve a class-1 job (only available if $k_1 \geq 1$); and **SC2** – serve a class-2 job (only available if $k_2 \geq 1$).

One can easily show that unforced idling is suboptimal due to the preemption assumption. Denote the action set by $\mathcal{A} = \{SC1, SU, Tr, SC2\}$ and the action taken at time t by $a(t)$. A control policy π specifies the action taken at time t given the current system state $X(t)$. Hence, we only consider control policies with Markovian properties. Our objective is to minimize the expected average cost per unit time over an infinite horizon which is formally defined as

$$g(\pi, s) = \limsup_{n \to \infty} \frac{V_n(\pi, s)}{n},\ \forall s \in \mathcal{S},$$

where $V_t(\pi, s)$ is total expected cost up to time t under policy π, starting from state s. In general it is possible that there are more than one optimal action for any given state. If that is the case, we choose the action that is listed earlier in the action set {SC1, SU, Tr, SC2}. For instance, SC1 has precedence over all the other actions. While this assumption is not crucial, it allows us to ensure that there is a unique optimal policy, which in turn simplifies the presentation of the results. Denote the optimal action at $X(t)$ by $a^*(X(t))$ and optimal expected average cost by $g^*(s)$:

$$g^*(s) = \inf_{\pi \in \Pi} g(\pi, s), \ \forall s \in \mathcal{S}, \tag{4.1}$$

4.3 State-dependent policies

4.3.1 Markov decision process formulation

Let r be the expected cost rate for an untriaged job, and r_i be the expected cost rate for a class-i job, $i = 1, 2$, then

$$r = ph_1 + qh_2, \ r_1 = \frac{pv_1h_1 + q(1-v_2)h_2}{PC_1}, \ r_2 = \frac{p(1-v_1)h_1 + qv_2h_2}{PC_2}. \tag{4.2}$$

Throughout the rest of this chapter, we assume that *uniformization* has been applied with the following uniformization constant

$$\phi = \lambda + \frac{1}{u} + \frac{1}{\tau}.$$

Without loss of generality we assume $\phi = 1$. Thus, instead of considering the above continuous-time problem, we study the discrete-time equivalent. Letting for $(i, k_1, k_2) \notin \mathcal{S}, \ v(i, k_1, k_2) = \infty$, the optimality equations can be written as follows. For $(i, k_1, k_2) \in \mathcal{S}$ and $i + k_1 + k_2 > 0$,

$$\begin{aligned} v(i,k_1,k_2) + g = \lambda v(i+1,k_1,k_2) + ir + k_1r_1 + k_2r_2 + \min\Big\{ & \\ & \frac{1}{u}\big[PC_1 v(i-1,k_1+1,k_2) + PC_2 v(i-1,k_1,k_2+1)\big] + \frac{1}{\tau}v(i,k_1,k_2), \\ & \frac{1}{\tau}v(i-1,k_1,k_2) + \frac{1}{u}v(i,k_1,k_2), \\ & \frac{1}{\tau}v(i,k_1-1,k_2) + \frac{1}{u}v(i,k_1,k_2), \\ & \frac{1}{\tau}v(i,k_1,k_2-1) + \frac{1}{u}v(i,k_1,k_2)\Big\}, \\ v(0,0,0) + g = \lambda v(1,0,0) + \frac{1}{\tau}v(0,0,0) + \frac{1}{u}v(0,0,0). & \end{aligned} \tag{4.3}$$

Proposition 4.3.1 (Existence of the optimal policy)**.** *Assume $\lambda\tau < 1$. There exist g, v that solve the* *above optimality equations. Moreover, there exists a stationary policy that is optimal for the above problem.*

4.3.2 Characterization of the optimal policy

The following theorem presents our main theoretical result, which describes the structure of the optimal policy as of threshold type.

Theorem 4.3.1. *There exists an optimal policy that can be described as follows:*

(i) If $k_1 \geq 1$, then $a^*(i, k_1, k_2) = SC1$, i.e., once the server identifies a class-1 job, the server should serve this job immediately.

(ii) The optimal action $a^*(i, k_1, k_2) = SC2$ only when $i = 0$ and $k_2 = 0$, i.e., the server serves a class-2 job only when there are no class-0 or class-1 jobs in the system.

(iii) If $k_1 = 0$ and $i > 0$, then for each $i \geq 1$, there exists a threshold $k_2^*(i)$ such that if $k_2 < k_2^*(i)$, the optimal action is triage; otherwise, serve without triage.

(iv) If $u \geq \tilde{u} := PC_2(r - r_2)\tau/r$, then $a^*(i, 0, k_2) = SU$ for all $(i, 0, k_2) \in S$.

This theorem establishes that for any given time and specified number of jobs of a given class (e.g., class-0, class-1, class-2), it is optimal to give class-1 jobs the highest priority and class-2 jobs the lowest priority. If there are no class-1 jobs, the server should triage a class-0 job if the number of the class-2 jobs is below a critical value, and serve a class-0 job directly without triage when the number of class-2 jobs is sufficiently large. The results agree with the well-known $c\mu$ rule, which gives priority to class-1 jobs over class-0 jobs and gives priority to class-0 jobs over class-2 jobs if triage is not an option. When there are many class-2 jobs (less-importance jobs) waiting for service, the value of job type information that will be obtained through triage could not compensate the additional delay that the remaining jobs will have to suffer. Hence, the optimal action is to skip triage. When triage takes a significant amount of time, the threshold, $k_2^*(i)$, becomes 0 implying that the optimal policy simplifies to the policy of not doing triage at all. The reason is that the value of triage diminishes when the jobs in the system have to endure long waiting. The expression of $\tilde{u}$ gives a precise description of what we mean by *a significant amount of time*.

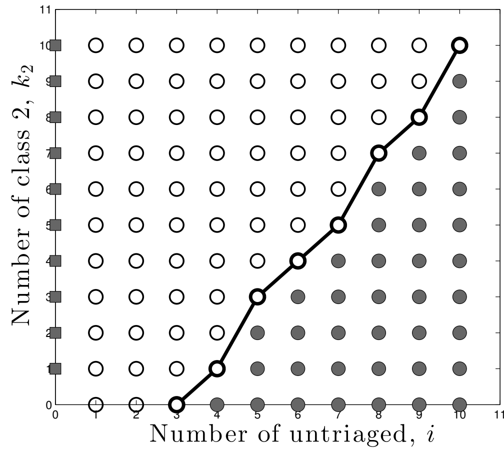

Figure 4.1: Visual description of the optimal policy when $k_1 = 0$ and $\lambda = 0.6, h_1 = 10, h_2 = 1, \tau_1 = \tau_2 = 1, v_1 = 0.9, v_2 = 0.95, u = 0.25, p = 0.6$.

An example of the threshold-type policy, determined by solving the Bellman's equation in (4.3) recursively, is presented in Figure 4.1. The x-axis represents the number of class-0/untriaged jobs and the y-axis represents the number of class-2 jobs in the system. The threshold-type policy, described in Theorem 4.3.1, reflects the real-time decision on whether to triage or not.

The proof follows after showing some properties of the optimal value functions $v(i, k_1, k_2)$. We first show that the desired properties for the value functions of the discounted version of the model are preserved under the value-iteration operator with small discount factor, then we show that such properties hold when the discount factor is approaching 0, which implies that the optimal value functions in the case of long-run average cost preserve such properties as well. The technical details are presented in Section 4.3.3.

Proposition 4.3.2. *If*

$$\lambda \leq \frac{\tau - u}{\tau^2}\left(1 - \frac{\tau}{\tau - u}\frac{r_2}{(\tilde{u}/u - 1)\, r + r_2}\right),$$

then $k_2^*(i)$ increases with i.

Proposition 4.3.2 states that $k_2^*(i)$ is increasing with i when the arrival rate is bounded above by certain value. The critical value increases as the number of untriaged jobs increases, since when there are many untriaged jobs waiting in queue, the cost reduced by discovering and prioritizing an important job is greater, which in turn means greater tolerance to the number of class-2 jobs waiting in queue.

4.3.3 Proof of Theorem 4.3.1 and Proposition 4.3.2

In this section, we sketch the idea for showing the theoretical results in Section 4.3.2. Consider a model in which the objective is to minimize the total discounted cost over an infinite time horizon. The model assumptions remain the same as those in Section 4.3.1. We refer to this model as the *discounted cost model* and the model in Section 4.3.1 as the *average cost model*. Let α denote the continuos-time discount factor, then we write down the MDP formulation of the total discounted model with uniformization factor $\lambda + \frac{1}{u} + \frac{1}{\tau} + \alpha$. Without loss of generality, we assume $\lambda + \frac{1}{u} + \frac{1}{\tau} + \alpha = 1$. However, all the results in this section hold without this assumption. The main idea is to first establish the structural properties of the optimal value functions in the discounted cost model, then extend the results to the average cost model by letting α go to zero. The optimality equations for the discounted cost model can be written as $v = Tv$, in which the operator T is defined as below. Similar to what we did in (4.3), we assume that for $(i, k_1, k_2) \notin \mathcal{S}$, $v(i, k_1, k_2) = \infty$. For $(i, k_1, k_2) \in \mathcal{S}$ and $i + k_1 + k_2 > 0$,

$$\begin{aligned} Tv(i,k_1,k_2) = {} & \lambda v(i+1,k_1,k_2) + ir + k_1 r_1 + k_2 r_2 + \min\Big\{ \\ & \frac{1}{u}\big[PC_1 v(i-1,k_1+1,k_2) + PC_2 v(i-1,k_1,k_2+1)\big] + \frac{1}{\tau}v(i,k_1,k_2), \\ & \frac{1}{\tau}v(i-1,k_1,k_2) + \frac{1}{u}v(i,k_1,k_2), \\ & \frac{1}{\tau}v(i,k_1-1,k_2) + \frac{1}{u}v(i,k_1,k_2), \\ & \frac{1}{\tau}v(i,k_1,k_2-1) + \frac{1}{u}v(i,k_1,k_2)\Big\}, \\ Tv(0,0,0) = {} & \lambda v(1,0,0) + \frac{1}{\tau}v(0,0,0) + \frac{1}{u}v(0,0,0). \end{aligned} \tag{4.4}$$

Proposition 4.3.3. *The optimal value function v^* can be obtained by the value iteration algorithm starting from any arbitrary function v_0, i.e.,*

$$\lim_{n\to\infty} T^{(n)} v_0 = v^*.$$

By Proposition 4.3.3, the optimal value function of the total discounted model exists and is well defined, we next show that it possesses the following sets of properties. We first define function $G(i, k_2)$ as

$$\begin{aligned} G(i,k_2) =& \frac{1}{u}[PC_1 v(i-1,1,k_2) + PC_2 v(i-1,0,k_2+1)] + \frac{1}{\tau}v(i,0,k_2) \\ & - \frac{1}{u}v(i,0,k_2) - \frac{1}{\tau}v(i-1,0,k_2),\ i \geq 1,\ k_2 \geq 0. \end{aligned} \tag{4.5}$$

Let $\mathcal{E}$ be the set of functions defined on Z^3 such that if $v \in \mathcal{E}$, then:

(e.1) $\frac{1}{\tau}v(i,k_1-1,k_2) + \frac{1}{u}v(i,k_1,k_2) < \frac{1}{u}\big[PC_1 v(i-1,k_1+1,k_2) + PC_2 v(i-1,k_1,k_2+1)\big] + \frac{1}{\tau}v(i,k_1,k_2),\ i \geq 1,\ k_1 \geq 1,\ k_2 \geq 0.$

(e.2) $v(i,k_1-1,k_2) < v(i-1,k_1,k_2),\ i \geq 1,\ k_1 \geq 1,\ k_2 \geq 0.$

(e.3) $v(i,k_1-1,k_2) < v(i,k_1,k_2-1),\ i \geq 0,\ k_1 \geq 1,\ k_2 \geq 1.$

(e.4) $v(i-1,k_1,k_2) < v(i,k_1,k_2-1),\ i \geq 1,\ k_1 \geq 0,\ k_2 \geq 1.$

(e.5) $v(i,k_1,k_2+1) - v(i,k_1,k_2) > 0,\ i \geq 0,\ k_1 \geq 0,\ k_2 \geq 0.$

Let $\mathcal{F}$ be the set of functions defined on Z^3 such that if $v \in \mathcal{F}$, then

(f.1) $\frac{PC_1}{u}\big[v(i,1,k_2) - v(i+1,0,k_2)\big] \leq r,\ i \geq 0,\ k_2 \geq 0.$

(f.2) $G(i,k_2) \geq 0,\ i \geq 1,\ k_2 \geq 0.$

Lemma 4.3.1. *Suppose* $u \geq \tilde{u}(\alpha) := \frac{PC_2(r-r_2)\tau}{r(1+\alpha\tau)}$. *If* $v \in \mathcal{E} \cap \mathcal{F}$, *then* $Tv \in \mathcal{E} \cap \mathcal{F}$.

Let $\mathcal{G}$ be the set of functions defined on Z^3 such that if $v \in \mathcal{G}$, then:

(g.1) $G(i,k_2) \leq G(i,k_2+1),\ i \geq 1,\ k_2 \geq 0.$

(g.2) $v(i+1,0,k_2) - v(i,1,k_2) \leq v(i+1,0,k_2+1) - v(i,1,k_2+1),\ i \geq 0,\ k_2 \geq 0.$

(g.3) $v(i,k_1,k_2+1) - v(i,k_1,k_2) \leq v(i,k_1,k_2+2) - v(i,k_1,k_1+1),\ i \geq 0,\ k_1 \geq 0,\ k_2 \geq 0.$

(g.4) $v(i,k_1,k_2+1) - v(i,k_1,k_2) \geq v(i-1,k_1,k_2+2) - v(i-1,k_1,k_2+1),\ i \geq 1,\ k_1 \geq 0,\ k_2 \geq 0.$

Lemma 4.3.2. *Suppose* $u < \tilde{u}(\alpha)$. *If* $v \in \mathcal{E} \cap \mathcal{G}$, *then* $Tv \in \mathcal{E} \cap \mathcal{G}$.

Let $\mathcal{H}$ be the set of functions defined on Z^3 such that if $v \in \mathcal{G}$, then:

(h.1) $G(i, k_2) \geq G(i+1, k_2),\ i \geq 1,\ k_2 \geq 0.$

(h.2) $v(i+1, 0, k_2) - v(i, 1, k_2) \leq v(i+1, 0, k_2+1) - v(i, 1, k_2+1),\ i \geq 0,\ k_2 \geq 0.$

(h.3) For $i \geq 1,\ k_2 \geq 0,\ \frac{PC_1}{u}\big[v(i-1, 1, k_2) - v(i, 0, k_2)\big] \geq \frac{\tilde{u}(\alpha)}{u} r.$

(h.4) For $i \geq 1,\ k_2 \geq 1$, $G(i, k_2) - \frac{1}{\tau}\big[v(i-1, 0, k_2) - v(i-1, 0, k_2-1)\big] \leq \frac{\tilde{u}(\alpha)}{u} r.$

(h.5) For $i \geq 0,\ k_1 \geq 0,\ k_2 \geq 0,$

$$v(0,0,1) - v(0,0,0) \leq \left(\frac{PC_2}{u} - \frac{1}{\tau}\right)(r - r_2)\tau^2,$$

$$\big[v(i, k_1, k_2+2) - v(i, k_1, k_2+1)\big] - \big[v(i, k_1, k_2+1) - v(i, k_1, k_2)\big] \leq \left(\frac{PC_2}{u} - \frac{1}{\tau}\right)(r - r_2)\tau^2.$$

(h.6) $G(i, 0) \leq r,\ i \geq 1.$

(h.7) For $i \geq 0,\ k_1 \geq 0,\ k_2 \geq 0$, $v(i, k_1, k_2+1) - v(i, k_1, k_2) \leq \frac{r_2}{\alpha}\Big[1 - \beta_1^{k_1+k_2+1}\beta_2^i\Big]$, where

$$\beta_1 = \frac{\frac{\lambda\tau}{\tau-u} + 1/\tau + \alpha - \sqrt{(\frac{\lambda\tau}{\tau-u} + 1/\tau + \alpha)^2 - 4\frac{\lambda\tau}{\tau-u}/\tau}}{2\lambda\tau/(\tau-u)}, \tag{4.6}$$

$$\beta_2 = \frac{\beta_1 - u/\tau}{1 - u/\tau}. \tag{4.7}$$

Lemma 4.3.3. *Suppose*

$$\lambda \leq \frac{\tau-u}{\tau^2}\left(1 - \frac{\tau}{\tau-u}\frac{r_2}{(\tilde{u}(\alpha)/u - 1)\, r + r_2}\right). \tag{4.8}$$

If $v \in \mathcal{E} \cap \mathcal{H}$, then $Tv \in \mathcal{E} \cap \mathcal{H}$.

The proofs of Lemma 4.3.1, 4.3.3 and 4.3.2 are given in Appendix B. Lemma 4.3.1, 4.3.3 and 4.3.2 show that the properties in set $\mathcal{E}, \mathcal{F}$ and $\mathcal{G}$ are preserved under operator T under certain conditions. Next, we show using Theorem 11.5 of Porteus (2002) that with the same conditions, the optimal value function of the discounted cost model satisfies the properties in sets $\mathcal{E}, \mathcal{F}$ and $\mathcal{G}$.

Lemma 4.3.4. *There exists an optimal value function for the discounted cost model that possesses (i) the properties in set $\mathcal{E} \cap \mathcal{F}$ if $u \geq \tilde{u}(\alpha)$; (ii) the properties in set $\mathcal{E} \cap \mathcal{G}$ if $u < \tilde{u}(\alpha)$; (iii) the properties in set $\mathcal{E} \cap \mathcal{H}$ if Equation* (4.8) *holds.*

Next, by verifying the three SEN conditions from Sennott (1999), we extend the above results to the average cost model by letting α go to 0, meaning that the properties of the optimal value functions for the discounted cost model hold for the optimal bias functions of the average cost model as well. Before we proceed, some technical details have to be explained. The discount factor in the SEN conditions, denoted by α_1 below, is the discrete-time discount rate, meaning that the present value of \$1 earned t days later is the same as \$$\alpha_1^t$. However, the discount factor in our model, α, is the continuos-time discount rate, meaning that the present value of \$1 earned t days later is \$$e^{-\alpha t}$. By uniformization, we transform a continuous-time MDP to a discrete-time MDP, and the continuous-time discount factor will be converted into an equivalent discrete-time discount factor. Thus letting $\alpha \to 0$ is equivalent as letting $\alpha_1 \to 1$. (See e.g., Alagoz and Ayvaci (2010))

Let z be a distinguished state in $\mathcal{S}$. The SEN conditions are

SEN1 The quantity $(1-\alpha_1)V_{\alpha_1}(z)$ is bounded, for $\alpha_1 \in (0,1)$. (This implies that $V_{\alpha_1}(z) < \infty$ and hence we may define the function $h_{\alpha_1}(i) =: V_{\alpha_1}(i) - V_{\alpha_1}(z)$ without fear of introducing an indeterminate form.)

SEN2 There exists a nonnegative (finite) function M such that $h_{\alpha_1}(i) \leq M(i)$ for $i \in \mathcal{S}, \alpha_1 \in (0,1)$.

SEN3 There exists a nonnegative (finite) constant L such that $-L \leq h_{\alpha_1}(i)$ for $i \in \mathcal{S}, \alpha_1 \in (0,1)$.

Lemma 4.3.5. *The three SEN conditions are satisfied for the discounted cost model.*

The proof of Lemma 4.3.5 is given in Appendix B. By Theorem 7.2.3 in Sennott (1999), the optimal bias functions of the average cost model inherit all the structural properties of the optimal value functions of the discounted cost model. Now, we are ready to prove the theoretical results in Section 4.3.2.

Proof of Theorem 4.3.1 and Proposition 4.3.2

Property **(e.1)**$\sim$**(e.4)** imply that if $k_1 \geq 1$, it is optimal to give priority to class-1 jobs; class-2 jobs should be served only when $i = k_1 = 0$, i.e., there are no jobs of other classes. Property **(f.2)** implies that when $u \geq \tilde{u}$, it is optimal to skip triage for any system state. When $u < \tilde{u}$, the monotonicity of $G(i, k_2)$ in k_2, i.e., Property **(g.1)**, implies that if it is optimal to skip triage in $(i, 0, k_2)$, it is optimal to do so in $(i, 0, k_2 + 1)$. On the other hand, if it is optimal to do triage in $(i, 0, k_2 + 1)$, it is optimal to do so in $(i, 0, k_2)$. Hence, the optimal policy on whether to triage or not is determined by a threshold for any given i. Note that this result is regardless of the value of u because when $u \geq \tilde{u}$, the optimal policy is a special form of the threshold-type policy with the threshold being 0. When Equation (4.8) holds, the monotonicity of $G(i, k_2)$ in i, i.e., Property **(h.1)**, implies that if it is optimal to skip triage in $(i + 1, 0, k_2)$, it is optimal to do so in $(i, 0, k_2)$. On the other hand, if it is optimal to do triage in $(i, 0, k_2)$, it is optimal to do so in $(i + 1, 0, k_2)$. Hence, the threshold $k_2^*(i)$ is an increasing function of i, which completes the proof of Proposition 4.3.2.

4.4 Simulation study

Unlike the case for the clearing model of Chapter 3, we have no explicit expressions for the threshold of the optimal policy. Therefore, it is particularly important to explore the performance of heuristic policies. The problem we investigate in this chapter has infinite state space. To numerically compute the optimal cost, we would need to solve the Bellman's equation which necessitates truncation of the state space. Hence, we would only able to get an approximation for the optimal cost. Since estimation of the error due to truncation is not possible, we carried out a simulation study instead of numerical experiments. First, we describe the three policies of interest.

No-Triage Policy (NT) Jobs are served in random order. No job goes through triage.

Triage-Prioritize-Class-1 Policy (TP_1) Each job goes through triage in random order. If a job is classified as class-1, it is served right away; otherwise, the job is put aside to be served later, and triage the next untriaged job. When there are no untriaged jobs in the system, class-2 jobs are served. Service is preemptive.

Threshold-type Policy (Th) Serve a class-1 job once it is identified, and serve a class-2 job only when there are no other types of jobs, i.e., $i = k_1 = 0$. When $k_1 = 0$ and $i > 0$, skip triage only if $k_2 \geq L(i)$, where

$$L(i) = \left(\frac{r(\tilde{u} - u)}{r_2 u}\right) i - \frac{r\tilde{u}}{r_2 u}, \tag{4.9}$$

and r, r_2 are defined in (4.2); $\tilde{u}$ is defined in Theorem 4.3.1.

Policy NT is the first-come-first-serve policy, and Policy TP_1 is the counterpart of the Triage-Prioritize-Class-1 Policy in the clearing system. The only difference is that the service of less important jobs, i.e., class-2 jobs, may be preempted by new arrivals. Policy Th is the same as the optimal policy for the clearing system in Chapter 3 except the parameters are adapted to that of Chapter 4.

The parameters of the simulation is described below. The service times are generated from exponential distributions with mean service time $\tau = 1$. The probability of a new arrival job being type-1, p, is fixed at 0.3. The conditional probabilities of correct classification for type-1 and type 2 are $v_1 = 0.9,\ v_2 = 1$, respectively. We chose λ from the set $\{0.1, 0.3, 0.5, 0.7, 0.9\}$, and chose h_1 from $\{1, 3, 5, 7, 9\}$ while h_2 is fixed at 1. The triage time is exponentially distributed and u is chosen from $\{0.1, 0.2, 0.3\}$. For each combination of (λ, h_1, u), the long-run average cost under Policy NT is computed by

$$g_{NT} = \frac{\rho}{1-\rho} \cdot r, \tag{4.10}$$

where $\rho = \lambda\tau$. We simulate the system evolution under Policy Th. We pick 2×10^4 time units as the warmup period by Welch's method. Then, we run the simulation for $T_s = 10^7$ time units in addition to the warmup period. We divide T_s into 1000 equal batches, each batch with 10^4 time units, and record the total cost incurred during each batch. We use the average cost of each batch, which can be easily obtained, to compute 95% confidence interval for the long-run average cost of the system under Policy Th. The mean of the confidence interval is denoted by g_{Th}.

Table 4.1 presents our simulation results. The confidence intervals of the long-run average cost for each scenario are displayed in columns labeled g_{Th}. To the right of each of the g_{Th} column, we present the percentage improvement, η, by Policy Th over Policy NT. If g_{NT} falls into the confidence interval of g_{Th}, the improvement is insignificant and we simply set $\eta = 0.00$; otherwise, it is defined as

$$\eta = \frac{g_{NT} - g_{Th}}{g_{NT}} \times 100. \tag{4.11}$$

From Table 4.1, we observe that the improvement is insignificant when (i) the cost rates of the two types are close; and/or (ii) when the arrival rate is small, and/or (iii) when the expected triage time is large. These observations are consistent with our intuition that when the two types of jobs are

h_1	λ	g_{NT}	$Th,\ u=0.1$		$Th,\ u=0.2$		$Th,\ u=0.3$	
			g_{Th}	Imprv%	g_{Th}	Imprv%	g_{Th}	Imprv%
1	0.1	0.111	0.111 ± 0.000	0.00	0.111 ± 0.000	0.00	0.111 ± 0.000	0.00
1	0.3	0.429	0.428 ± 0.001	0.00	0.428 ± 0.001	0.00	0.428 ± 0.001	0.00
1	0.5	1.000	0.999 ± 0.003	0.00	0.999 ± 0.003	0.00	0.999 ± 0.003	0.00
1	0.7	2.333	2.334 ± 0.011	0.00	2.334 ± 0.011	0.00	2.334 ± 0.011	0.00
1	0.9	9.000	8.945 ± 0.125	0.00	8.945 ± 0.125	0.00	8.945 ± 0.125	0.00
3	0.1	0.178	0.177 ± 0.000	0.00	0.178 ± 0.001	0.00	0.178 ± 0.001	0.00
3	0.3	0.686	0.670 ± 0.002	2.36	0.685 ± 0.002	0.00	0.684 ± 0.002	0.00
3	0.5	1.600	1.517 ± 0.004	5.22	1.598 ± 0.004	0.00	1.602 ± 0.005	0.00
3	0.7	3.733	3.459 ± 0.018	7.36	3.723 ± 0.016	0.00	3.736 ± 0.018	0.00
3	0.9	14.400	13.607 ± 0.161	5.51	14.303 ± 0.173	0.00	14.407 ± 0.182	0.00
5	0.1	0.244	0.240 ± 0.001	1.77	0.244 ± 0.001	0.00	0.245 ± 0.001	0.00
5	0.3	0.943	0.886 ± 0.002	6.08	0.933 ± 0.002	1.02	0.942 ± 0.002	0.00
5	0.5	2.200	1.945 ± 0.006	11.60	2.121 ± 0.007	3.61	2.201 ± 0.006	0.00
5	0.7	5.133	4.285 ± 0.017	16.53	4.817 ± 0.020	6.16	5.122 ± 0.023	0.00
5	0.9	19.800	16.266 ± 0.218	17.85	18.247 ± 0.210	7.84	19.626 ± 0.249	0.00
7	0.1	0.311	0.305 ± 0.001	2.06	0.310 ± 0.001	0.00	0.311 ± 0.001	0.00
7	0.3	1.200	1.101 ± 0.003	8.23	1.163 ± 0.003	3.07	1.199 ± 0.003	0.00
7	0.5	2.800	2.365 ± 0.006	15.53	2.590 ± 0.008	7.49	2.766 ± 0.009	1.20
7	0.7	6.533	5.048 ± 0.020	22.74	5.763 ± 0.023	11.79	6.336 ± 0.031	3.02
7	0.9	25.200	19.220 ± 0.249	23.73	22.077 ± 0.264	12.39	24.189 ± 0.317	4.01
9	0.1	0.378	0.367 ± 0.001	2.80	0.374 ± 0.001	0.91	0.378 ± 0.001	0.00
9	0.3	1.457	1.316 ± 0.003	9.72	1.390 ± 0.003	4.60	1.449 ± 0.004	0.55
9	0.5	3.400	2.785 ± 0.007	18.09	3.065 ± 0.009	9.84	3.313 ± 0.009	2.57
9	0.7	7.933	5.789 ± 0.020	27.03	6.725 ± 0.030	15.23	7.483 ± 0.038	5.68
9	0.9	30.600	21.706 ± 0.251	29.06	25.769 ± 0.288	15.79	28.482 ± 0.343	6.92

Table 4.1: Comparison in the average cost by using the heuristic policy Th as opposed to No-Triage policy.

similar in the sense of the cost rates, then triage brings few benefits and No-Triage Policy performs similar to the optimal policy. When the arrival rate is small, service resources are sufficient and the congestion level in the system is low. Hence, triage is not needed. When triage takes too much time, the additional delay imposed on the jobs can not be justified by the benefits brought by triage. On the contrary, when the differences among the jobs are significant, and/or the traffic intensity is high, and/or triage is fast, Policy Th improves over No-Triage Policy as much as 29%. Not surprisingly, the most significant improvement happens when $h_1 = 9, \lambda = 0.9$ and $u = 0.1$, i.e., when the two types of jobs are most different, the system's congestion level is high and triage can be done rapidly.

CHAPTER 5: EXTENSIONS

In this chapter, we study three extensions. In Section 5.1, we study a clearing model as in Chapter 3 but this time we consider having multiple identical servers instead of a single server. In Section 5.2, we consider a model with arrivals as in Chapter 4, but this time triage is instantaneous and incurs a fixed cost. In Section 5.3, we discuss the case that triage is not optional and is required to be done for each new arrival before services. In each section, we describe the model assumptions and are able to characterize the optimal policy partially or completely.

5.1 Multiple identical servers

Consider a service system with M servers and N jobs. The servers are identical and work in a non-cooperative manner. Each job belongs to one of the two types, type 1 and type 2. A job is of type 1 with probability p, and is of type 2 with probability $q \equiv 1 - p$. Both p and q are exogenous parameters, and will not change over time. The service time of a job from type i is exponentially distributed with mean $\tau > 0$, and a type i job incurs a cost with rate h_i per unit time the job stays in the system, $i = 1,\ 2$. Without loss of generality, we assume that type 1 jobs are more important than type 2 jobs in the sense of higher cost rate, i.e., $h_1 > h_2$.

We assume that the type information of a job is hidden from the service system initially, i.e., servers do not know the exact type of a job, but servers could serve a job without knowing its type. The servers also have the option to spend some time on investigation, i.e., triage, to obtain the type information of a job, and classify the job as class 1 or class 2. The triage time of a job is exponentially distributed with mean $u > 0$, independent the job's type. We denote class 1 as the important class, and class 2 as the less important class. Each server tries her/his best to classify the type 1 jobs into class 1, and type 2 jobs into class 2. While triaging a job provides information on the job's type, the classification is error-prone. Define v_1 as the probability of classifying a type 1 job into class 1 and v_2 as the probability of classifying a type 2 job into class 2. Without loss of generality, assume that

$v_1 + v_2 > 1$. Denote PC_i as the probability of classifying a random job into class i, where $i = 1, 2$. Then, $PC_1 = pv_1 + q(1 - v_2),\ PC_2 = p(1 - v_1) + qv_2$.

We further assume that a preemptive discipline is used and there is no cost for switching actions. The decision epochs are time zero, and triage and service completion times for the server. The state of the system can then be denoted by the triplet (i, k_1, k_2), where i represents the number of *untriaged* jobs, and k_1 and k_2 denote the number of jobs that have been classified as class-1 and class-2, respectively. Since we have N jobs in total, the state space can be described as $\mathcal{S} = \{(i, k_1, k_2) : i, k_1, k_2 \geq 0, i + k_1 + k_2 \leq N\}$.

Using a sample-path argument, it is straightforward to show that keeping any of the servers idle is suboptimal. This allows us to ignore idling as an admissible action. Then, in a given state $s = (i, k_1, k_2)$, the available actions for each server are **SU:** serve an untriaged job without triage (only available if $i \geq 1$); **Tr:** triage an untriaged job (only available if $i \geq 1$); **SC1:** serve a class-1 job (only available if $k_1 \geq 1$); and **SC2:** serve a class-2 job (only available if $k_2 \geq 1$). Our objective is to minimize the total expected cost.

Let $V_\pi(i, k_1, k_2)$ denote the total expected cost under policy π and $V(i, k_1, k_2) = \min_\pi \{V_\pi(i, k_1, k_2)\}$ to be the total expected cost under an optimal policy starting from state (i, k_1, k_2) with no service or triage in progress.

Theorem 5.1.1. *In the optimal policy, servers will work on the same type of jobs if possible.*

5.2 Instantaneous triage

Consider a service system with a single server and two types of jobs, type 1 and type 2. Jobs arrive to the system according to a Poisson process with a total rate λ, and they wait in a queue if they are not served upon arrival. The waiting space is unlimited. For convenience, we use "class 0" or untriaged jobs to denote these new arrivals that have not gone through triage. Each job belongs to type 1 with probability p and to type 2 with probability $q \equiv 1 - p$. Both p and q are exogenous parameters, and will not change over time. The service time of a job from type i is exponentially distributed with mean $\tau > 0$, and a type i job incurs a cost with rate h_i per unit time the job stays in the system where $i = 1,\ 2$. Without loss of generality, we assume that type 1 jobs are more important than type 2 jobs from the service provider's perspective in the sense of higher cost rate, i.e., $h_1 > h_2$.

We assume that the type information of an arriving job is hidden from the service system initially, i.e., the server does not know the exact type of a new arrival, but s/he could serve a job without knowing its type. The server also has the option to spend some time on investigation, i.e., triage, to obtain the type information of a job, and classify the job as class 1 or class 2. Unlike in Chapter 4, we assume that triage takes no time but a fixed cost $C > 0$. We denote class 1 as the important class, and class 2 as the less important class. The server tries her/his best to classify the type 1 jobs into class 1, and type 2 jobs into class 2. While the investigation on a job provides information on the job's type, the classification is error-prone. Define v_1 as the probability of classifying a type 1 job into class 1 and v_2 as the probability of classifying a type 2 job into class 2. Denote PC_i as the probability of classifying a random job into class i, where $i = 1, 2$. Then, $PC_1 = pv_1 + q(1 - v_2),\ PC_2 = p(1 - v_1) + qv_2$.

We further assume that a preemptive discipline is used and there is no cost or changeover time for the server to switch actions. Let $x_j(t)$ denote the number of jobs in class j at time t where $j = 0, 1, 2$, then $X(t) = (x_0(t), x_1(t), x_2(t))$ is the current state of the system. Hence, the state space $\mathcal{S} = \{(i, k_1, k_2) : i \geq 0, k_1 \geq 0, k_1 \geq 0\}$. At any time, the provider of the service system can take one of the following four actions: **SU** – serve an untriaged job without triage (only available if $i \geq 1$); **Tr** – triage an untriaged job (only available if $i \geq 1$); **SC1** – serve a class-1 job (only available if $k_1 \geq 1$); and **SC2** – serve a class-2 job (only available if $k_2 \geq 1$).

One can easily show that unforced idling is suboptimal due to the preemption assumption. In general it is possible that there are more than one optimal action for any given state. If that is the case, we choose the action that is listed earlier in the action set {SC1, SU, Tr, SC2}. For instance, SC1 has precedence over all the other actions. While this assumption is not crucial, it allows us to ensure that there is a unique optimal policy, which in turn simplifies the presentation of the results. A control policy π specifies the action taken at time t given the current system state $X(t)$. Denote the action taken at time t by $a(t)$ and the optimal action at $X(t)$ as $a^*(X(t))$. The total expected discounted cost under any policy π, denoted by v_π, is defined by

$$v_\pi(s) = E_\pi \left[\int_0^\infty K(X(t), a(t)) e^{-\alpha t} dt | X(0) = s \right],\ \forall s \in \mathcal{S},$$

where $K(X(t), a(t))$ is the cost rate at time t when the system state is $X(t)$ and the action is $a(t)$, α is the discount factor. Our objective is to identify policies that minimize the total discounted cost.

Throughout the rest of this section, we assume that *uniformization* has been applied with the following uniformization constant

$$\phi = \lambda + \frac{1}{\tau} + \alpha.$$

Without loss of generality we assume $\phi = 1$. Thus, instead of considering the above continuous-time problem, we study a discrete-time version. With the same notation in Chapter 4, the optimality equations for the total discounted model can be written as $v = Tv$, where the operator T is defined as below. Similar to what we did in (4.3), we assume that for $(i, k_1, k_2) \notin \mathcal{S},\ v(i, k_1, k_2) = \infty$. For $(i, k_1, k_2) \in \mathcal{S}$ and $i + k_1 + k_2 > 0$,

$$\begin{aligned} Tv(i, k_1, k_2) = \min \Big\{ & PC_1 v(i-1, k_1+1, k_2) + PC_2 v(i-1, k_1, k_2+1) + C, \\ & \lambda v(i+1, k_1, k_2) + \frac{1}{\tau} v(i-1, k_1, k_2) + ir + k_1 r_1 + k_2 r_2, \\ & \lambda v(i+1, k_1, k_2) + \frac{1}{\tau} v(i, k_1 - 1, k_2) + ir + k_1 r_1 + k_2 r_2, \\ & \lambda v(i+1, k_1, k_2) + \frac{1}{\tau} v(i, k_1, k_2 - 1) + ir + k_1 r_1 + k_2 r_2 \Big\}, \\ Tv(0,0,0) = {} & \lambda v(1,0,0) + \frac{1}{\tau} v(0,0,0). \end{aligned} \tag{5.1}$$

The following theorem provides a partial characterization of the optimal policy.

Theorem 5.2.1. *Consider state $(i, k_1, k_2) \in \mathcal{S}$:*

(i) If $k_1 \geq 1$, then $a^*(i, k_1, k_2) = \text{SC1}$, i.e., as soon as the server identifies a class-1 job, that job should be served next.

(ii) If $i + k_1 > 0$, then $a^*(i, k_1, k_2) \neq \text{SC2}$, i.e., it is optimal to serve a class-2 job only when there are no untriaged or class-1 jobs.

Theorem 5.2.1 depicts the service order of the untriaged jobs, class-1 jobs and class-2 jobs. It is optimal to give class-1 jobs the highest priority and class-2 jobs the lowest priority. These results can be easily extended to a model with the same settings but to minimize the long-run average cost. The proof repeats the idea used in the proof of Theorem 4.3.2. The decision question remains on when to triage and when to serve the untriaged jobs directly without triage. From the intuition of Chapter 4, we conjecture that there exists a single threshold such that if the number of untriaged jobs exceeds the threshold, skip triage; otherwise, do triage. The benefit of triaging a job is to be able to prioritize an important job. The magnitude of the benefit becomes greater when there are many untriaged jobs waiting for service, which is consistent with the result in Chapter 4 that the server prefers to do triage when the number of untriaged jobs is large. In Chapter 4, the threshold depends on both the number of untriaged jobs and the number of class-2 jobs because triage takes time and causes delay to all the jobs in the system. However, in our current model, triage takes no time but a fixed cost, which means there will be no delay imposed on the class-2 jobs in the system. Hence, we conjecture that the optimal policy can be characterized by a single threshold that is independent of the number of class-2 jobs.

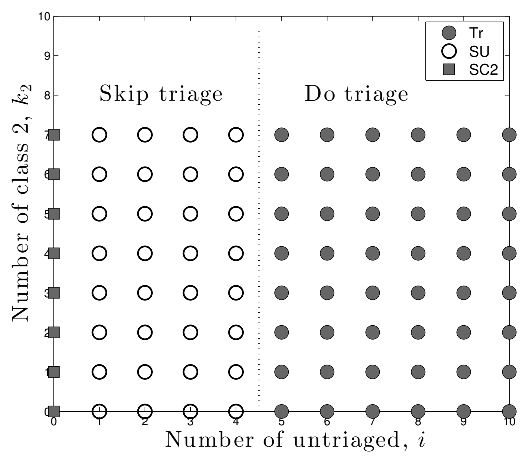

Figure 5.1: An example of the threshold-type policy when triage is instantaneous and incurs a fixed cost.

Figure 5.1 provides an example to illustrate the conjectured optimal policy when there are no class-1 jobs. The threshold is $i^* = 4$. When the number of untriaged jobs is more than 4, it is optimal for the server to do triage; otherwise, the server should skip triage. The threshold i^* does not depend on the number of class-2 jobs. We are not able to prove that the conjectured policy is optimal now. In the following section, we study the set of such policies.

5.2.1 Threshold-type policies

In this section, we study a set of threshold policies where if $i > i^*$, do triage; otherwise, skip triage. Under this type of policy, we calculated the long-run expected cost. Let $\mu = 1/\tau$. Given an i^*, we model the system dynamic as a CTMC (continuous-time Markov chain). We can write down the balance equations as follows.

$$\lambda \pi_{0,0,0} = \mu \pi_{0,0,1} + \mu \pi_{1,0,0}. \tag{5.2}$$

$$(\lambda + \mu)\pi_{0,0,k_2} = \mu \pi_{0,0,k_2+1} + \mu \pi_{1,0,k_2},\ k_2 \geq 1. \tag{5.3}$$

$$(\lambda + \mu)\pi_{i,0,k_2} = \lambda \pi_{i-1,0,k_2} + \mu \pi_{i+1,0,k_2},\ 1 \leq i \leq i^* - 1, k_2 \geq 0. \tag{5.4}$$

$$(\lambda + \mu)\pi_{i^*,0,0} = \lambda \pi_{i^*-1,0,0} + \mu \pi_{i^*,1,0}. \tag{5.5}$$

$$(\lambda + \mu)\pi_{i^*,0,k_2} = \lambda \pi_{i^*-1,0,k_2} + \lambda P C_2 \pi_{i^*,0,k_2-1} + \mu \pi_{i^*,1,k_2},\ k_2 \geq 1. \tag{5.6}$$

$$(\lambda + \mu)\pi_{i^*,k_1,0} = \lambda \pi_{i^*,k_1-1,0} + \mu \pi_{i^*,k_1+1,0},\ k_1 \geq 1. \tag{5.7}$$

$$(\lambda + \mu)\pi_{i^*,k_1,k_2} = \lambda P C_1 \pi_{i^*,k_1-1,k_2} + \lambda P C_2 \pi_{i^*,k_1,k_2-1} + \mu \pi_{i^*,k_1+1,k_2},\ k_1, k_2 \geq 1. \tag{5.8}$$

The transition diagram of the CTMC is

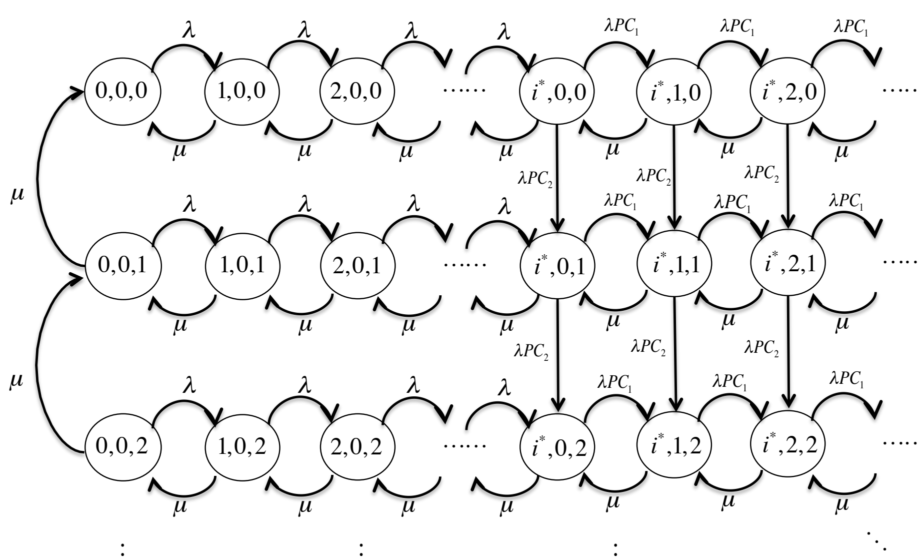

Figure 5.2: Transition diagram of the CMTC under a given threshold-type policy.

Next, we solve the stationary distribution of this CTMC and compute the long-run average cost.

Let $\rho \equiv \lambda\tau$.

STEP 1 Add up (5.2) **and** (5.3)

$$\lambda \sum_{k_2=0}^{\infty} \pi_{0,0,k_2} = \mu \sum_{k_2=0}^{\infty} \pi_{1,0,k_2} \;\Rightarrow\; \sum_{k_2=0}^{\infty} \pi_{1,0,k_2} = \rho \sum_{k_2=0}^{\infty} \pi_{0,0,k_2}.$$

STEP 2 For each $1 \leq i \leq i^* - 1$**, add up** (5.4)

$$\begin{aligned}
&\lambda \sum_{k_2=0}^{\infty} \pi_{1,0,k_2} = \mu \sum_{k_2=0}^{\infty} \pi_{2,0,k_2} \;\Rightarrow\; \sum_{k_2=0}^{\infty} \pi_{2,0,k_2} = \rho \sum_{k_2=0}^{\infty} \pi_{1,0,k_2} = \rho^2 \sum_{k_2=0}^{\infty} \pi_{0,0,k_2},\\
&\quad\vdots\\
&\lambda \sum_{k_2=0}^{\infty} \pi_{i^*-1,0,k_2} = \mu \sum_{k_2=0}^{\infty} \pi_{i^*,0,k_2} \;\Rightarrow\; \sum_{k_2=0}^{\infty} \pi_{i^*,0,k_2} = \rho \sum_{k_2=0}^{\infty} \pi_{i^*-1,0,k_2} = \rho^{i^*} \sum_{k_2=0}^{\infty} \pi_{0,0,k_2},
\end{aligned}$$

STEP 3 For each $k_1 \geq 0$**, add up** (5.5)**,** (5.6)**,** (5.7) **and** (5.8)

$$\begin{aligned}
&\lambda PC_1 \sum_{k_2=0}^{\infty} \pi_{i^*,0,k_2} = \mu \sum_{k_2=0}^{\infty} \pi_{i^*,1,k_2} \;\Rightarrow\; \sum_{k_2=0}^{\infty} \pi_{i^*,1,k_2} = \rho PC_1 \sum_{k_2=0}^{\infty} \pi_{i^*,0,k_2} = (\rho PC_1)\rho^{i^*} \sum_{k_2=0}^{\infty} \pi_{0,0,k_2},\\
&\lambda PC_1 \sum_{k_2=0}^{\infty} \pi_{i^*,1,k_2} = \mu \sum_{k_2=0}^{\infty} \pi_{i^*,2,k_2} \;\Rightarrow\; \sum_{k_2=0}^{\infty} \pi_{i^*,2,k_2} = \rho PC_1 \sum_{k_2=0}^{\infty} \pi_{i^*,1,k_2} = (\rho PC_1)^2\rho^{i^*} \sum_{k_2=0}^{\infty} \pi_{0,0,k_2},\\
&\quad\vdots
\end{aligned}$$

Hence,

$$\begin{aligned}
&\sum_{k_2=0}^{\infty} \pi_{i,0,k_2} = \rho^i \sum_{k_2=0}^{\infty} \pi_{0,0,k_2},\; 0 \leq i \leq i^* - 1.\\
&\sum_{k_2=0}^{\infty} \pi_{i,k_1,k_2} = (\rho PC_1)^{k_1}\rho^{i^*} \sum_{k_2=0}^{\infty} \pi_{0,0,k_2},\; k_1 \geq 0.
\end{aligned}$$

Since

$$\sum_{i=0}^{i^*-1} \sum_{k_2=0}^{\infty} \pi_{i,0,k_2} + \sum_{k_1=0}^{\infty} \sum_{k_2=0}^{\infty} \pi_{i^*,k_1,k_2} = 1,$$

we can get

$$\sum_{k_2=0}^{\infty} \pi_{0,0,k_2} = \left(\frac{1-\rho^{i^*}}{1-\rho} + \frac{\rho^{i^*}}{1-\rho PC_1} \right)^{-1}.$$

Define Q_i $(0 \le i \le i^* - 1)$ and QC_{k_1} $(k_1 \ge 0)$ as follows:

$$Q_i \equiv \sum_{k_2=0}^{\infty} \pi_{i,0,k_2},\ 0 \le i \le i^* - 1,$$
$$QC_{k_1} \equiv \sum_{k_2=0}^{\infty} \pi_{i^*,k_1,k_2},\ k_1 \ge 0.$$

Then, Q_i $(0 \le i \le i^* - 1)$ and QC_{k_1} $(k_1 \ge 0)$ are

$$Q_0 = \left(\frac{1-\rho^{i^*}}{1-\rho} + \frac{\rho^{i^*}}{1-\rho PC_1}\right)^{-1} = \frac{(1-\rho)(1-\rho PC_1)}{1 - PC_1\rho - PC_2\rho^{i^*+1}}, \tag{5.9}$$
$$Q_i = \rho^i Q_0,\ 0 \le i \le i^* - 1, \tag{5.10}$$
$$QC_{k_1} = (\rho PC_1)^{k_1}\rho^{i^*} Q_0,\ k_1 \ge 0. \tag{5.11}$$

Denote the expected number of untriaged jobs, class-1 jobs, class-2 jobs by $L_0(\cdot)$, $L_1(\cdot)$ and $L_2(\cdot)$, respectively. Then,

$$\begin{aligned}
L_0(i^*) &= \sum_{i=0}^{i^*-1} i \cdot Q_i + \sum_{k_1=0}^{\infty} i^* \cdot QC_{k_1} = \sum_{i=0}^{i^*-1} i \cdot \rho^i Q_0 + \sum_{k_1=0}^{\infty} i^* \cdot (\rho PC_1)^{k_1}\rho^{i^*} Q_0 \\
&= \left[\frac{\rho - \rho^{i^*}}{(1-\rho)^2} - \frac{(i^*-1)\rho^{i^*}}{1-\rho} + \frac{i^*\rho^{i^*}}{1-\rho PC_1}\right] Q_0, && (5.12)\\
L_1(i^*) &= \sum_{k_1=0}^{\infty} k_1 \cdot QC_{k_1} = \sum_{k_1=0}^{\infty} k_1 \cdot (\rho PC_1)^{k_1}\rho^{i^*} Q_0 = \frac{PC_1\rho^{i^*+1}Q_0}{(1-PC_1\rho)^2}, && (5.13)\\
L_2(i^*) &= \frac{\rho}{1-\rho} - L_0(i^*) - L_1(i^*). && (5.14)
\end{aligned}$$

Based on (5.9)$\sim$(5.14), we get a closed-form expression for the long-run average cost of the system given the threshold i^*.

$$\begin{aligned}
\phi(i^*) &= rL_0(i^*) + r_1L_1(i^*) + r_2L_2(i^*) + \sum_{k_1=0}^{\infty} C \cdot \lambda QC_{k_1} \\
&= rL_0(i^*) + r_1L_1(i^*) + r_2L_2(i^*) + \frac{\lambda\rho^i Q_0}{1-PC_1\rho}C. && (5.15)
\end{aligned}$$

Based on (5.15), we can compute the optimal threshold in order to minimize the long-run average cost within the set of threshold type policies. Consider an example. Let $\lambda = 0.9,\ p = 0.6,\ v_1 =$

$1,\ v_2 = 0.9,\ \tau = 1,\ h_1 = 10,\ h_2 = 1,\ C = 50$. For each given i^*, we compute the long-run average cost $\phi(i^*)$ by (5.15). The threshold i^* varies from 0 to 30. We plot the 31 points $(i^*,\ \phi(i^*))$ and connect them to their adjacent neighbors with line segments in Figure 5.3. In this example, it is quite obvious that the long-run average cost is minimized when the threshold is 4. It means that in order to achieve the minimum cost, the server should perform triage when there are no class-1 jobs and more than 4 untriaged jobs; otherwise, skip triage. This is a simple example. With the closed-form expression for $\phi(i^*)$, the optimal threshold can be computed for examples in which $\phi(i^*)$ is more complicated.

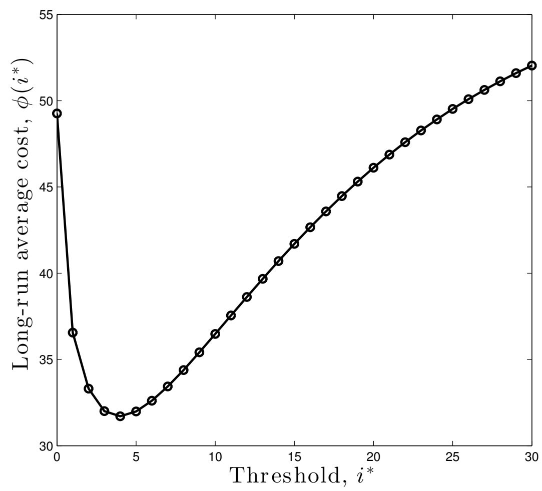

Figure 5.3: The long-run average cost as a function of the threshold i^*.

5.3 When triage is not optional

In practice, there are situations that triage has to be done, due to ethical issues or protocol standards, or obtaining the type information of a job is a necessary step for subsequent service. Hence, every job will go through triage, and skipping triage is not an option to the server.

The basic model setup is that we consider a service system with a single server and two types of jobs, type 1 and type 2. Jobs arrive to the system according to a Poisson process with a total rate λ, and they wait in a queue if they are not served upon arrival. The waiting space is unlimited. For convenience, we use class 0 to denote these new arrival jobs that have not receive any service. Each job belongs to type 1 with probability p and to type 2 with probability $q \equiv 1 - p$. Both p and q are exogenous parameters, and will not change over time. A type i job incurs a cost with rate h_i per unit time the job stays in the system where $i = 1,\ 2$. The service time of a job from type i has a general distribution with mean $\tau_i > 0$. Without loss of generality, we assume that type 1 jobs are important than type 2 jobs from the service provider's perspective in the sense of higher cost rate, i.e., $h_1/\tau_1 > h_2/\tau_2$.

We assume that the type information of an arriving job is hidden from the service system initially. The server will triage each new arrival, and classify the job as class 1 or 2. The mean triage time of any job is $u > 0$, independent of the arrival process and the job's type. Triaging a job provides information on the job's type, however, the classification is error-prone. Define v_1 as the probability of classifying a type 1 job into class 1 and v_2 as the probability of classifying a type 2 job into class 2. Without loss of generality, we assume that $v_1 + v_2 > 1$. Denote PC_i as the probability of classifying a random job into class i, where $i = 1, 2$. Then, $PC_1 = pv_1 + q(1 - v_2),\ PC_2 = p(1 - v_1) + qv_2$. Both triage and service are non-preemptive. Our objective is to minimize the long-run average cost.

Under this setup, the available actions at any decision epoch will be **Tr** – triage an untriaged job (if there is one); **SC1** – serve a class-1 job (if there is one); and **SC2** – serve a class-2 job (if there is one). We find that this model is a special case of the model in Klimov (1974). Applying the main result in Klimov (1974), we get the following theorem.

Theorem 5.3.1. *The optimal policy can be described as follows:*

(i) As soon as the server identifies a class-1 job, that job should be served next.

(ii) If the mean triage time $u < (rT_2 - r_2T)/r_2$, triaging an untriaged job is more preferable than serving a class-2 job; otherwise, serving a class-2 job has higher priority than triage.

The notations $r, r_i, T, T_i,\ i = 1, 2$, are defined in Table 3.1. Theorem 5.3.1 implies that (i) If the expected triage time is greater than $(rT_2 - r_2T)/r_2$, the server should triage a job and serve this job immediately regardless of the job's type; (ii) If the expected triage time is smaller than $(rT_2 - r_2T)/r_2$, Triage-Prioritize-Class-1 Policy defined in Section 3.3 is optimal. Note that this result is independent of the arrival process and the distributions of the triage times and service times.

The inequality on the triage time, $u < (rT_2 - r_2T)/r_2$, can be rewritten as $r_2/T_2 < r/(u + T)$. The left-hand side is the expected $c\mu$ value for class-2 jobs. Similarly, the expected $c\mu$ value for class-

1 jobs is r_1/T_1, and we can regard $r/(u+T)$ as the expected $c\mu$ for triaging an untriaged job. It is straightforward to show that

$$\frac{r_1}{T_1} > \frac{r}{T} > \frac{r}{u+T} \text{ and } \frac{r_1}{T_1} > \frac{r_2}{T_2}.$$

However, the order between $r/(u+T)$ and r_2/T_2 is indefinite. If $r/(u+T) > r_2/T_2$, give priority to triage over serve class-2; otherwise, prioritize class-2 jobs over untriaged jobs. This implies that the optimal policy is essentially an index policy. By comparing the index for each class of jobs, the server can make a decision on which action to choose among triage an untriaged job, serve a class-1 job and serve a class-2 job. At last, we point out that Theorem 5.3.1 holds when there are no external arrivals at the system, i.e., a clearing system. This can be shown by interchange argument and induction on the number of untriaged jobs in the system.

CHAPTER 6: CONCLUSIONS AND DIRECTIONS FOR FUTURE RESEARCH

In this dissertation, we introduce a generic model to investigate the information/delay trade-off when scheduling jobs of hidden types. We consider two different models, one assuming all jobs being present at time zero and the other assuming new arrivals. In both models, the server has the option to extract the type information at the cost of delaying service (*triage*). For both models, we provide characterizations of the optimal policy.

In the first model, namely the clearing system, it is shown to be optimal to prioritize the class of jobs with the larger expected cost over service time ratio. When it comes to make a decision on whether to triage or not, the optimal policy is shown to be described by a switching curve, a line, to be precise. One implication of this property is that when the number of untriaged jobs waiting for service exceeds a threshold, performing triage brings benefits offsetting the delay imposed on other jobs. We provide a complete expression for this line. Although our result is proven only under the assumption of linear holding cost, with numerical study we find that several heuristic policies developed from the optimal policy of the clearing system perform well when the holding cost is nonlinear. Thus, our numerical results provide some justification for the robustness of the optimal policy. We also compare four simple baseline policies and provide some insights. For example, we find that it is possible that the policy that prioritizes less important jobs performs better, and improvements in the classification process that results in lower misclassification probabilities do not necessarily lead to better outcomes.

In the second model, we study the same information/delay trade-off but consider new job arrivals. Unlike in the clearing system where we do not assume any specific distributions for the service/triage times, we assume service and triage times are exponentially distributed and assume that service times for all jobs are i.i.d. for analytical tractability. Under these assumptions, we prove that the optimal policy is again of threshold type, hence the insights from the clearing system continue to hold. By means of a simulation study, we find that a threshold type policy, which may not necessarily be the optimal threshold policy, could bring significant improvements over the first-come-first-serve policy.

We also study three extensions to these two models mentioned above. Even though the basic settings are similar, some important assumptions are modified or relaxed; namely, we consider the cases where (i) there are multiple servers in a clearing model, or (ii) triage is instantaneous and incurs fixed cost in a queueing model, or (iii) triage can not be avoided in both clearing and queueing systems. We use Markov decision process formulation to study each of them and partially characterize the optimal control policy. This study is more of an exploratory work that could potentially lead to more detailed analysis.

The formulation and results in this dissertation will contribute to the understanding of the information/delay trade-off shared by many services in practice. Even in our daily lives, we constantly prioritize our tasks while assessing the relative value of prioritizing one task over the other given highly imperfect information. The insights from the mathematical analysis also provide guidelines to decision makers, especially, to the emergency response community, in their efforts to devise practical and efficient policies. Of course, more work remains to be done. There are several aspects of the control problem that merit additional analysis. In the following, we discuss two of them.

Multiple servers with different skills

In many settings, especially in the aftermath of many mass-casualty incidents, there will be more than one emergency responder at the scene, some performing triage while others treating patients. These servers may have different skills. Some servers (e.g., triage nurses) are only trained to triage, and we refer to them as *dedicated servers*. There are other servers (e.g., paramedic and physicians) that have multiple skills and could both triage and treat the patients. We refer to them as *flexible servers*. The question of interest is how to allocate the flexible servers. The goal is to find the optimal dynamic policy that decides when to send the flexible servers to perform triage and when they should focus on treating patients.

Two levels of triage

In this dissertation, the decision on triage is binary: the server either performs triage or skips triage. In addition, the probability of correct classification is not a function of the triage time. In reality, triage nurses could spend more time on triage, for instance, by performing more tests, to get a better idea about the patient's condition and help classify the patient into the right class. Therefore, we can consider multiple levels of triage as options to server. For example, there can be two levels: *simple triage* and *advanced triage*. While advanced triage takes more time, it provides a better classification of the jobs in the sense of higher accuracy. The goal is to find the optimal decision on triage: whether the server should skip triage, perform simple triage, or perform advanced triage. Based on our analysis in Chapter 3, we conjecture that there again exists a switching curve that separates the states in which skipping triage is optimal from the states in which performing triage is optimal. However, how the decision is made between using simple triage and using advanced triage is unclear and is of interest.

APPENDIX A: PROOF OF RESULTS IN CHAPTER 3

Expressions and derivations for C_{NT}, C_{TP_1}, C_{TP_2}, and C_{TAF}

Expression for C_{NT}

Consider the i^{th} job to be served, $1 \leq i \leq N$. The sojourn time in the system is the total service time of the first $i-1$ jobs plus its own service time. This leads to

$$\begin{aligned} C_{NT} &= \sum_{i=1}^{N} [ph_1\tau_1 + qh_2\tau_2 + (i-1)(ph_1 + qh_2)(p\tau_1 + q\tau_2)] \\ &= N(ph_1\tau_1 + qh_2\tau_2) + \frac{N(N-1)}{2}(ph_1 + qh_2)(p\tau_1 + q\tau_2). \end{aligned} \tag{A.1}$$

Expressions for C_{TP_1} and C_{TP_2}

We will derive an expression for C_{TP_1}. An expression for C_{TP_2} can be obtained similarly. Let N_1 denote the number of class-1 jobs among the $N-1$ jobs that go through triage. We know $N_1 \sim B(N-1, PC_1)$ where $B(n,p)$ indicates a binomial random variable with parameters n and p. Let $N_1 = k$ and $s = \{s_1, s_2, \cdots, s_k\}$ where s_j indicates the order number of the jth class-1 job when the server is picking among the untriaged jobs randomly, i.e. the jth job classified as Class-1 is the s_jth job to have been picked by the server among the untriaged jobs. Assume $s_1 = i_1, s_2 = i_2, \cdots, s_k = i_k, 1 \leq i_j < N, j = 1, 2, \cdots, k$. We can show that $P\big(s_1 = i_1, s_2 = i_2, \cdots, s_k = i_k \big| N_1 = k\big) = \frac{1}{\binom{N-1}{k}}$.

Using the notation defined in Table 3.1, conditional on $N_1 = k$ and $s = \{s_1, s_2, \cdots, s_k\}$, the expected cost incurred by k class-1 jobs due to the triage of all $N-1$ jobs is given by $\Gamma_1\Big|_{s,N_1=k}(triage) = r_1 \cdot (i_1u + i_2u + \cdots + i_ku) = r_1u\sum_{m=1}^{k} i_m$. Conditional on $N_1 = k$ and $s = \{s_1, s_2, \cdots, s_k\}$, the expected cost incurred by k class-1 jobs due to the service of all k class-1 jobs is given by $\Gamma_1\Big|_{s,N_1=k}(service) = c_1 + (c_1 + r_1T_1) + \cdots + (c_1 + (k-1)r_1T_1) = kc_1 + \frac{k(k-1)}{2}r_1T_1$. Then, the total expected cost incurred by the k class-1 jobs is $\Gamma_1\Big|_{s,N_1=k} = kc_1 + r_1\left(u\sum_{m=1}^{k} i_m + \frac{k(k-1)}{2}T_1\right)$.

Define $\mathcal{A}_k = \{\text{all possible combinations of } (i_1, \cdots, i_k), 1 \leq i_1 < i_2 < \cdots < i_k < N\}$. The total expected cost for the k class-1 jobs is

$$\begin{aligned}
\Gamma_1\Big|_{N_1=k} &= \sum_{\mathcal{A}_k} \Gamma_1\Big|_{s,N_1=k} \cdot P\big(s_1 = i_1, s_2 = i_2, \cdots, s_k = i_k \big| N_1 = k\big) \\
&= \sum_{\mathcal{A}_k} \left[kc_1 + r_1 \left(u \sum_{m=1}^{k} i_m + \frac{k(k-1)}{2} T_1 \right) \right] \cdot P\big(s_1 = i_1, s_2 = i_2, \cdots, s_k = i_k \big| N_1 = k\big) \\
&= kc_1 + \frac{k(k-1)}{2} r_1 T_1 + r_1 u \sum_{\mathcal{A}_k} \sum_{m=1}^{k} i_m \cdot P\big(s_1 = i_1, s_2 = i_2, \cdots, s_k = i_k \big| N_1 = k\big) \\
&= kc_1 + \frac{k(k-1)}{2} r_1 T_1 + r_1 u \sum_{\mathcal{A}_k} \sum_{m=1}^{k} i_m \cdot \frac{1}{\binom{N-1}{k}}.
\end{aligned}$$

In the term $\sum_{\mathcal{A}_k} \sum_{m=1}^{k} i_m$ each number $i \in \{1, 2, \cdots, N-1\}$ appears exactly $k \cdot \binom{N-1}{k}/(N-1)$ times. Hence the total expected cost for class-1 jobs is

$$\begin{aligned}
\Gamma_1\Big|_{N_1=k} &= \sum_{\mathcal{A}_k} \Gamma_1\Big|_{s,N_1=k} \cdot P\big(s_1 = i_1, s_2 = i_2, \cdots, s_k = i_k \big| N_1 = k\big) \\
&= kc_1 + \frac{k(k-1)}{2} r_1 T_1 + r_1 u \cdot \frac{k \cdot \binom{N-1}{k}}{N-1} \cdot \frac{N(N-1)}{2} \cdot \frac{1}{\binom{N-1}{k}} \\
&= kc_1 + \frac{k(k-1)}{2} r_1 T_1 + \frac{Nk}{2} r_1 u. \qquad \text{(A.2)}
\end{aligned}$$

The expected cost for the last job, the only job that does not go through triage, conditional on $N_1 = k$ is

$$\psi(k) = r\left[(N-1)u + kT_1\right] + c. \qquad \text{(A.3)}$$

The total expected cost for class-2 jobs when there are k class-1 jobs is

$$\begin{aligned}
\Gamma_2\Big|_{N_1=k} &= \Big(r_2[(N-1)u + T + kT_1] + c_2\Big) + \Big(r_2\left[(N-1)u + T + kT_1 + T_2\right] + c_2\Big) + \cdots \\
&+ \Big(r_2[(N-1)u + T + kT_1 + (N-k-2)T_2] + c_2\Big) \\
&= \sum_{m=k+2}^{N} \Big(c_2 + r_2[(N-1)u + T + kT_1] + r_2(m-k-2)T_2\Big). \qquad \text{(A.4)}
\end{aligned}$$

Then, the total expected cost under Policy TP_1 is

$$C_{TP_1} = \sum_{k=0}^{N-1} \left[\left(\sum_{\mathcal{A}_k} \Gamma_1 \Big|_{N_1=k} \cdot P\big(s_1 = i_1, \cdots, s_k = i_k \big| N_1 = k\big) \right) + \psi(k) + \Gamma_2 \Big|_{N_1=k} \right] P(N_1 = k). \tag{A.5}$$

Plugging in (A.2), (A.3) and (A.4) into (A.5) with some algebraic manipulation we get

$$\begin{aligned} C_{TP_1} = & \sum_{k=0}^{N-1} \Bigg[\Big(kc_1 + \frac{k(k-1)}{2} r_1 T_1 + \frac{Nk}{2} r_1 u \Big) + \Big(r\left[(N-1)u + kT_1\right] + c \Big) \\ & + \sum_{m=k+2}^{N} \Big(c_2 + r_2[(N-1)u + T + kT_1] + r_2(m-k-2)T_2 \Big) \Bigg] P(N_1 = k) \\ = & \, N(N-1)(ph_1 + qh_2)u - \frac{(N-1)(N-2)}{2} \big[pv_1h_1 + q(1-v_2)h_2\big]u + N(ph_1\tau_1 + qh_2\tau_2) \\ & + \frac{N(N-1)}{2}(ph_1 + qh_2)(p\tau_1 + q\tau_2) - \frac{N(N-1)}{2} pq(v_1 + v_2 - 1)\tau_1\tau_2\left(\frac{h_1}{\tau_1} - \frac{h_2}{\tau_2}\right). \end{aligned}$$

Using (A.1), we get

$$\begin{aligned} C_{TP_1} = & \, N(N-1)(ph_1 + qh_2)u - \frac{(N-1)(N-2)}{2} \big[pv_1h_1 + q(1-v_2)h_2\big]u \\ & + C_{NT} - \frac{N(N-1)}{2} pq(v_1 + v_2 - 1)\tau_1\tau_2\left(\frac{h_1}{\tau_1} - \frac{h_2}{\tau_2}\right). \end{aligned} \tag{A.6}$$

We can similarly obtain

$$\begin{aligned} C_{TP_2} = & \, N(N-1)(ph_1 + qh_2)u - \frac{(N-1)(N-2)}{2} \big[p(1-v_1)h_1 + qv_2h_2\big]u \\ & + C_{NT} - \frac{N(N-1)}{2} pq(v_1 + v_2 - 1)\tau_1\tau_2 \left(\frac{h_2}{\tau_2} - \frac{h_1}{\tau_1} \right). \end{aligned} \tag{A.7}$$

Expression for C_{TAF}

The derivation of C_{TAF} is similar to that of C_{TP_1}. The total expected triage cost for all jobs is $N \cdot (ph_1 + qh_2) \cdot Nu$. Denote N_1 as the number of jobs classified as class-1, $N_1 \sim B(N, PC_1)$. Then, given $N_1 = k$, the total expected cost incurred by class-1 jobs during the service of all class-1 jobs is $\Gamma_1 \Big|_{N_1=k} = \Big(c_1 + (k-1)r_1T_1\Big) + \Big(c_1 + (k-2)r_1T_1\Big) + \cdots + \Big(c_1 + (k-k)r_1T_1\Big) = kc_1 + \frac{k(k-1)}{2} r_1 T_1$. Given $N_1 = k$, the total expected cost incurred by class-2 jobs during the service of all jobs is $\Gamma_2 \Big|_{N_1=k} = \Big((N-k)r_2 \cdot kT_1\Big) + \Big(c_2 + (N-k-1)r_2T_2\Big) + \Big(c_2 + (N-k-2)r_2T_2\Big) + \cdots +$

$\Big(c_2 + 0 \cdot r_2 T_2\Big) = (N-k) r_2 \cdot k T_1 + (N-k) c_2 + \frac{(N-k)(N-k-1)}{2} r_2 T_2$. Hence, the total expected cost under Policy TAF is

$$
\begin{aligned}
C_{TAF} &= N^2(ph_1 + qh_2)u + \sum_{k=0}^{N} \Bigg[\Big(kc_1 + \frac{k(k-1)}{2} r_1 T_1 \Big) + \Big((N-k) r_2 \cdot k T_1 \\
&+ (N-k)c_2 + \frac{(N-k)(N-k-1)}{2} r_2 T_2 \Big) \Bigg] P(N_1 = k) \\
&= N^2(ph_1 + qh_2)u + N(ph_1\tau_1 + qh_2\tau_2) + \frac{N(N-1)}{2}(ph_1 + qh_2)(p\tau_1 + q\tau_2) \\
&- \frac{N(N-1)}{2} pq(v_1 + v_2 - 1)\tau_1\tau_2 (\frac{h_1}{\tau_1} - \frac{h_2}{\tau_2}).
\end{aligned}
$$

Using (A.1), we get

$$
C_{TAF} = N^2(ph_1 + qh_2)u + C_{NT} - \frac{N(N-1)}{2} pq(v_1 + v_2 - 1)\tau_1\tau_2 (\frac{h_1}{\tau_1} - \frac{h_2}{\tau_2}). \tag{A.8}
$$

Proof of Proposition 3.3.1.

Part (i): From (A.6) and (A.8), we have

$$
C_{TP_1} - C_{TAF} = -\frac{(N-1)(N-2)}{2} \big[pv_1h_1 + q(1-v_2)h_2 \big] u \leq 0.
$$

Part (ii): From (A.1) and (A.7), we have

$$
\begin{aligned}
C_{TP_2} - C_{NT} &= N(N-1)(ph_1 + qh_2)u - \frac{(N-1)(N-2)}{2} \big[p(1-v_1)h_1 + qv_2h_2 \big] u \\
&+ \frac{N(N-1)}{2} pq(v_1 + v_2 - 1)\tau_1\tau_2 (\frac{h_1}{\tau_1} - \frac{h_2}{\tau_2}) \\
&= (N-1)\left(N - \frac{N-2}{2}(1-v_1) \right) ph_1u + (N-1)\left(N - \frac{N-2}{2} v_2 \right) qh_2u \\
&+ \frac{N(N-1)}{2} pq(v_1 + v_2 - 1)\tau_1\tau_2 (\frac{h_1}{\tau_1} - \frac{h_2}{\tau_2}) > 0,
\end{aligned}
$$

where the inequality follows from Assumption 3.2.1. □

Proof of Proposition 3.3.2.

From Equations (A.1) and (A.6), we have

$$\begin{aligned}
C_{TP_1} - C_{NT} &= N(N-1)(ph_1+qh_2)u - \frac{(N-1)(N-2)}{2}\big[pv_1h_1+q(1-v_2)h_2\big]u \\
&- \frac{N(N-1)}{2}pq(v_1+v_2-1)\tau_1\tau_2(\frac{h_1}{\tau_1}-\frac{h_2}{\tau_2}) \\
&= \frac{h_1}{\tau_1}(N-1)\left(Np\tau_1u - \frac{N-2}{2}pv_1\tau_1u - \frac{N}{2}pq(v_1+v_2-1)\tau_1\tau_2\right) \\
&+ \frac{h_2}{\tau_2}(N-1)\left(Nq\tau_2u - \frac{N-2}{2}q(1-v_2)\tau_2u + \frac{N}{2}pq(v_1+v_2-1)\tau_1\tau_2\right).
\end{aligned}$$

Using Assumption 3.2.1, one can see that $Nq\tau_2u - \frac{N-2}{2}q(1-v_2)\tau_2u + \frac{N}{2}pq(v_1+v_2-1)\tau_1\tau_2 \geq 0$. One can then immediately obtain $C_{TP_1} - C_{NT} < 0$ if and only if $\alpha < \beta(p)$. □

Proof of Proposition 3.3.3.

Define $a = \left(\frac{N-2}{N}v_1 - 2\right)\tau_1u < 0$, $b = \left(2 - \frac{N-2}{N}(1-v_2)\right)\tau_2u > 0$, and $c = (v_1+v_2-1)\tau_1\tau_2 > 0$. Then

$$\beta(p) = \max\left\{0, \frac{pa+p(1-p)c}{(1-p)b+p(1-p)c}\right\}.$$

One can then show that $\beta(p) = 0$ for $p = 0$ and $p \geq \hat{p}$, and $\beta(p) > 0$ and differentiable for $p \in (0, \hat{p})$, where $\hat{p} = (a+c)/c$. Then, for $p \in (0, \hat{p})$,

$$\frac{d\beta(p)}{dp} = \frac{(a+b)cp^2 - 2bc \cdot p + b(a+c)}{[(1-p)b+p(1-p)c]^2}.$$

By Rolle's mean value theorem, there exists $0 < p^* < \hat{p}$ such that $d\beta(p^*)/dp = 0$. We can also show that

$$\lim_{p\to 0}\frac{d\beta(p)}{dp} = \frac{a+c}{b} > 0, \quad \lim_{p\to\hat{p}}\frac{d\beta(\hat{p})}{dp} = \frac{c(a+c)}{a(a+b+c)} < 0. \tag{A.9}$$

Both of the inequalities above follow from the fact that $\beta(p) > 0$ when $p \in (0, \hat{p})$, which implies that $a + c \geq a + qc > 0$.

By (A.9) and the differentiability of $\beta(p)$, the number of stationary points of $\beta(p)$ in $[0,\ \hat{p}]$ can only be odd. Since $d\beta(p)/dp = 0$ can have at most two solutions, p^* is the only stationary point, which means $d\beta(p)/dp > 0$ for $0 < p < p^*$, $d\beta(p)/dp < 0$ for $p^* < p < \hat{p}$. Hence, $\beta(p)$ is quasi-concave and is first non-decreasing and non-increasing over $(0, 1)$, which therefore implies the existence of the interval $I(\alpha)$. The rest of the proposition immediately follows. □

Proof of Proposition 3.3.4.

It is obvious that $\frac{C_{NT} - C_{T_1}}{C_{NT}} > \eta$ is equivalent to $C_{T_1} < (1 - \eta)C_{NT}$. Similar to the proofs of Proposition 3.3.2, we can obtain (i) and (iii). To show (ii), we restrict ourselves to p where $\beta(p) > 0$. The reason is explained as follows.

If $\beta(p) = 0$, then we must have

$$p\tau_1\big[\frac{N-2}{N}v_1 - 2\big]u + pq(v_1 + v_2 - 1)\tau_1\tau_2 \leq 0,$$

since the denominator of $\beta(p)$ is positive. Hence, for any $\eta > 0$,

$$p\tau_1\big[\frac{N-2}{N}v_1 - 2\big]u + pq(v_1 + v_2 - 1)\tau_1\tau_2 - \eta p\tau_1[\frac{2}{N-1}\tau_1 + p\tau_1 + q\tau_2] < 0,$$

i.e. $\beta(p, \eta) = 0$, which is not of interest. Define

$$\begin{aligned}
A &= p\tau_1\big[\frac{N-2}{N}v_1 - 2\big]u + pq(v_1 + v_2 - 1)\tau_1\tau_2, \\
B &= p\tau_1[\frac{2}{N-1}\tau_1 + p\tau_1 + q\tau_2], \\
C &= q\tau_2\big[2 - \frac{N-2}{N}(1 - v_2)\big]u + pq(v_1 + v_2 - 1)\tau_1\tau_2, \\
D &= q\tau_2[\frac{2}{N-1}\tau_2 + p\tau_1 + q\tau_2],
\end{aligned}$$

then, $A \geq 0,\ B \geq 0,\ C \geq 0,\ D \geq 0$. We can rewrite $\beta(p, \eta)$ as

$$\beta(p, \eta) = \frac{A - B\eta}{C + D\eta}.$$

If $\eta_1 > \eta_2$, then

$$\beta(p,\eta_1) - \beta(p,\eta_2) = \frac{A - B\eta_1}{C + D\eta_1} - \frac{A - B\eta_2}{C + D\eta_2} = -\frac{(AD + BC)(\eta_1 - \eta_2)}{(C + D\eta_1)(C + D\eta_2)} < 0. \qquad \square$$

Proof of Proposition 3.3.5.

From (A.6) and (A.7), we have

$$\begin{aligned} C_{TP_1} - C_{TP_2} &= -(N-1)(N-2)(ph_1(v_1 - 1/2) - qh_2(v_2 - 1/2))u \\ &- N(N-1)pq(v_1 + v_2 - 1)\tau_1\tau_2(\frac{h_1}{\tau_1} - \frac{h_2}{\tau_2}) \\ &= -\frac{h_1}{\tau_1}(N-1)\Big[(N-2)p\tau_1(v_1 - 1/2)u + Npq(v_1 + v_2 - 1)\tau_1\tau_2\Big] \\ &+ \frac{h_2}{\tau_2}(N-1)\Big[(N-2)q\tau_2(v_2 - 1/2)u + Npq(v_1 + v_2 - 1)\tau_1\tau_2\Big]. \qquad \text{(A.10)} \end{aligned}$$

If $v_2 < 1/2 - \frac{Np\tau_1}{(N-2)u}(v_1 + v_2 - 1)$, using the second part of Assumption 3.2.1, one can show that $(N-2)q\tau_2(v_2 - 1/2)u + Npq(v_1 + v_2 - 1)\tau_1\tau_2 < 0$ and $(N-2)p\tau_1(v_1 - 1/2)u + Npq(v_1 + v_2 - 1)\tau_1\tau_2 > 0$, which in turn imply that $C_{TP_1} < C_{TP_2}$ for all $\alpha \in (0,1)$. This completes the proof of part (i).

Similarly, if $v_1 < 1/2 - \frac{Nq\tau_2}{(N-2)u}(v_1 + v_2 - 1)$, using the second part of Assumption 3.2.1, one can show that $(N-2)q\tau_2(v_2 - 1/2)u + Npq(v_1 + v_2 - 1)\tau_1\tau_2 > 0$ and $(N-2)p\tau_1(v_1 - 1/2)u + Npq(v_1 + v_2 - 1)\tau_1\tau_2 < 0$, which imply that $C_{TP_1} > C_{TP_2}$ for all $\alpha \in (0,1)$. This completes the proof of part (ii).

To prove part (iii), $v_1 > 1/2 - \frac{Nq\tau_2}{(N-2)u}(v_1 + v_2 - 1)$ and $v_2 > 1/2 - \frac{Np\tau_1}{(N-2)u}(v_1 + v_2 - 1)$ imply that $(N-2)q\tau_2(v_2 - 1/2)u + Npq(v_1 + v_2 - 1)\tau_1\tau_2 > 0$ and $(N-2)p\tau_1(v_1 - 1/2)u + Npq(v_1 + v_2 - 1)\tau_1\tau_2 > 0$, respectively. Then, from (A.10), it follows that $C_{TP_1} - C_{TP_2} < 0$ if and only if $\alpha < \theta(p)$.

It now remains to show that $\beta(p) < \theta(p)$. If $\beta(p) = 0$ or $\theta(p) = 1$, the result immediately follows since $\theta(p)$ is guaranteed to be positive and $\beta(p)$ is guaranteed to be less than 1. Now, when $\beta(p) > 0$

and $\theta(p) < 1$,

$$
\begin{aligned}
&\beta(p) - \theta(p) \\
=&\Bigg[\Big(p\tau_1\big[\frac{N-2}{N}v_1 - 2\big]u + pq(v_1+v_2-1)\tau_1\tau_2\Big)\Big(\frac{N-2}{N}q\tau_2(v_2-1/2)u + pq(v_1+v_2-1)\tau_1\tau_2\Big) \\
&-\Big(\frac{N-2}{N}p\tau_1(v_1-1/2)u + pq(v_1+v_2-1)\tau_1\tau_2\Big)\Big(q\tau_2\big[2-\frac{N-2}{N}(1-v_2)\big]u + pq(v_1+v_2-1)\tau_1\tau_2\Big)\Bigg] \\
\Bigg/&\Bigg[\Big(q\tau_2\big[2-\frac{N-2}{N}(1-v_2)\big]u + pq(v_1+v_2-1)\tau_1\tau_2\Big)\Big(\frac{N-2}{N}q\tau_2(v_2-1/2)u + pq(v_1+v_2-1)\tau_1\tau_2\Big)\Bigg]
\end{aligned}
$$

The denominator is positive, therefore, $\beta(p) - \theta(p) < 0$ if and only if

$$
\begin{aligned}
&\big(p\tau_1\big[\tfrac{N-2}{N}v_1 - 2\big]u + pq(v_1+v_2-1)\tau_1\tau_2\big)\big(\tfrac{N-2}{N}q\tau_2(v_2-1/2)u + pq(v_1+v_2-1)\tau_1\tau_2\big) \\
&-\big(\tfrac{N-2}{N}p\tau_1(v_1-1/2)u + pq(v_1+v_2-1)\tau_1\tau_2\big)\big(q\tau_2\big[2-\tfrac{N-2}{N}(1-v_2)\big]u + pq(v_1+v_2-1)\tau_1\tau_2\big) < 0.
\end{aligned}
$$

By using basic algebra, one can show that the left-hand side of the above inequality is

$$
-\frac{3N+2}{2N}pq\tau_1\tau_2u(v_1+v_2-1)(\frac{N-2}{N}u + p\tau_1 + q\tau_2),
$$

which is negative. Hence, $\theta(p) > \beta(p)$ for $p \in (0,1)$. □

Proof of Proposition 3.3.6.

Part (i): The result immediately follows by observing

$$
\frac{\partial C_{TP_1}}{\partial v_1} = -\frac{(N-1)(N-2)}{2}ph_1u - \frac{N(N-1)}{2}pq\tau_1\tau_2(\frac{h_1}{\tau_1} - \frac{h_2}{\tau_2}) < 0.
$$

Part (ii):

$$
\begin{aligned}
\frac{\partial C_{TP_1}}{\partial v_2} &= \frac{(N-1)(N-2)}{2}qh_2u - \frac{N(N-1)}{2}pq\tau_1\tau_2(\frac{h_1}{\tau_1} - \frac{h_2}{\tau_2}) \\
&= \frac{N-1}{2}q\left[(N-2)\tau_2u\frac{h_2}{\tau_2} - Np\tau_1\tau_2(\frac{h_1}{\tau_1} - \frac{h_2}{\tau_2})\right]
\end{aligned}
$$

$$= \frac{N(N-1)}{2} q\tau_2 \left[\frac{h_2}{\tau_2} \cdot \left(p\tau_1 + \tfrac{N-2}{N}u\right) - \frac{h_1}{\tau_1} \cdot p\tau_1 \right].$$

Now, if $\alpha > \gamma(p)$, i.e. $\frac{h_2/\tau_2}{h_1/\tau_1} > \frac{p\tau_1}{p\tau_1 + \frac{N-2}{N}u}$, we get $\frac{\partial C_{TP_1}}{\partial v_2} > \frac{N(N-1)}{2} q\tau_2 \left(\frac{h_1}{\tau_1} Np\tau_1 - \frac{h_1}{\tau_1} Np\tau_1 \right) = 0$.

Taking the derivative of $\gamma(p)$ with respect to p, we find $\frac{d\gamma(p)}{dp} = \frac{\frac{N-2}{N}\tau_1 u}{\left[p\tau_1 + \frac{N-2}{N}u\right]^2} > 0$, i.e., $\gamma(p)$ is an increasing function of p. Finally, if $\beta(p) = 0$, then $\gamma(p) > \beta(p)$ is immediate. Otherwise,

$$\begin{aligned} \gamma(p) - \beta(p) &= \frac{p\tau_1}{p\tau_1 + \frac{N-2}{N}u} - \frac{p\tau_1\left[\frac{N-2}{N}v_1 - 2\right]u + pq(v_1 + v_2 - 1)\tau_1\tau_2}{q\tau_2\left[2 - \frac{N-2}{N}(1 - v_2)\right]u + pq(v_1 + v_2 - 1)\tau_1\tau_2} \\ &= \frac{p\tau_1 u \left(p\tau_1 + q\tau_2 + \frac{N-2}{N}u\right)\left(2 - \frac{N-2}{N}v_1\right)}{\left(p\tau_1 + \frac{N-2}{N}u\right)\left(q\tau_2\left[2 - \frac{N-2}{N}(1 - v_2)\right]u + pq(v_1 + v_2 - 1)\tau_1\tau_2\right)} > 0, \end{aligned}$$

where we make use of Assumption 3.2.1 in establishing the inequality. □

Proof of Theorem 3.4.1.

We first prove the following lemma:

Lemma A.0.1. *For all $(i, k_1, k_2) \in \mathcal{S}$, we have*

(i) $V(i, k_1 + 1, k_2) \geq V(i, k_1, k_2) + c_1$.

(ii) $V(i, k_1, k_2 + 1) \geq V(i, k_1, k_2) + c_2$.

(iii) If $u \geq T$, then $a^*(i, k_1, k_2) \neq \mathrm{Tr}$.

Proof of Lemma A.0.1:

Part (i): The proof uses a coupling argument. Consider two systems. System 1 and System 2 are identical except that System 1 starts in state $(i, k_1 + 1, k_2)$ and uses the optimal policy and System 2 starts in state (i, k_1, k_2) and uses policy π, which takes whatever action System 1 takes until System 1 starts serving the extra class-1 job System 2 lacks. While System 1 serves the extra class-1 job, System 2 idles and then follows the same actions as System 1 until all jobs are cleared.

Let the total expected cost under policy π be denoted by $V_\pi(i, k_1, k_2)$. The difference between $V(i, k_1 + 1, k_2)$ and $V_\pi(i, k_1, k_2)$ is at least as large as the expected cost incurred during the service of the additional class-1 job. Hence,

$$
\begin{aligned}
V(i,k_1+1,k_2)-V(i,k_1,k_2) &= V(i,k_1+1,k_2)-V_\pi(i,k_1,k_2)+V_\pi(i,k_1,k_2)-V(i,k_1,k_2)\\
&\geq V(i,k_1+1,k_2)-V_\pi(i,k_1,k_2)\geq c_1.
\end{aligned}
$$

Part (ii): The proof is similar to that for part (i) and is therefore omitted.

Part (iii): In any state (i,k_1,k_2), the total expected cost of first doing *triage* then following the optimal policy is

$$J_T(i,k_1,k_2)=PC_1V(i-1,k_1+1,k_2)+PC_2V(i-1,k_1,k_2+1)+(ir+k_1r_1+k_2r_2)u.$$

The total expected cost of first doing *serve without triage* then following the optimal policy is

$$J_{NT}(i,k_1,k_2)=V(i-1,k_1,k_2)+c+[(i-1)r+k_1r_1+k_2r_2]T.$$

By parts (i) and (ii) of the lemma,

$$
\begin{aligned}
J_T(i,k_1,k_2) &= PC_1V(i-1,k_1+1,k_2)+PC_2V(i-1,k_1,k_2+1)+(ir+k_1r_1+k_2r_2)u\\
&\geq PC_1\left[V(i-1,k_1,k_2)+c_1\right]+PC_2\left[V(i-1,k_1,k_2)+c_2\right]+(ir+k_1r_1+k_2r_2)u\\
&= V(i-1,k_1,k_2)+PC_1\cdot c_1+PC_2\cdot c_2+(ir+k_1r_1+k_2r_2)u\\
&\geq V(i-1,k_1,k_2)+c+[(i-1)r+k_1r_1+k_2r_2]u\\
&\geq V(i-1,k_1,k_2)+c+[(i-1)r+k_1r_1+k_2r_2]T=J_{NT}(i,k_1,k_2),
\end{aligned}
$$

where the last inequality follows from the assumption of part (iii) of the lemma. Hence, taking action SU (*serve without triage*) is always at least as good as taking action Tr (*triage*) if $u\geq T$. $\square$

Proof of Theorem 3.4.1:

Part (i): Let $k=i+k_1+k_2$. If $k=1$, the result trivially holds since $k=k_1=1$, i.e., the only job in the system is of class-1. Now assume the result is true for some $k\geq 1$. Using interchange arguments we will show that it holds for $k+1$ as well. One by one, we will show that *serve class 1* (SC1) is better than every other possible action.

(i)-1: Action **SC1** is better than **SC2**

Define policy π_1 as the policy that first serves a class-2 job (assuming there is one) and then follows the optimal policy. Then, under policy π_1, the second job served must be of class-1 by the induction assumption. Now consider policy γ_1 that switches the order of the first two jobs under policy π_1 and then follows the same set of actions. The expected cost of the two policies are $C_{\pi_1} = c_2 + c_1 + r_1 T_2 + C_1$ and $C_{\gamma_1} = c_1 + c_2 + r_2 T_1 + C_1$, where C_1 denotes the expected waiting cost and service cost for the remaining $(k-1)$ jobs, which is the same under both policy π_1 and γ_1. Then,

$$C_{\gamma_1} - C_{\pi_1} = r_2 T_1 - r_1 T_2 = -\frac{pq\tau_1\tau_2\,(v_1 + v_2 - 1)\,(h_1/\tau_1 - h_2/\tau_2)}{PC_1 \cdot PC_2} < 0.$$

Hence, by Assumption 3.2.1, serving a class-1 job is better than serving a class-2 job.

(i)-2: Action **SC1** is better than **SU**

Define policy π_2 as the policy that first serves an untriaged job (assuming there is one) without triage and then follows the optimal policy. Then, under policy π_2, the second job served must be of class-1. Now consider policy γ_2 that switches the order of the first two jobs under policy π_2 and then follows the same set of actions. The expected cost under the two policies are respectively $C_{\pi_2} = c + c_1 + r_1 T + C_2$ and $C_{\gamma_2} = c_1 + c + r T_1 + C_2$, where C_2 denotes the expected waiting cost and service cost for the remaining $k-1$ jobs, which is the same under both policy π_2 and γ_2. Then,

$$C_{\gamma_2} - C_{\pi_2} = r T_1 - r_1 T = -\frac{pq\tau_1\tau_2\,(v_1 + v_2 - 1)\,(h_1/\tau_1 - h_2/\tau_2)}{PC_1} < 0.$$

Hence, by Assumption 3.2.1, serving the class 1 first is better than serving without triage.

(i)-3: Action **SC1** is better than **Tr**

Define policy $\pi_3(m)$ as the policy that first triages m untriaged jobs then serves one class 1 job and follows the optimal policy where $0 \le m \le i$. As we established above, there must exist $0 \le m^* \le i$ and policy π_{m^*} is optimal. If $m^* = 0$, the proof is done. Otherwise, consider policy $\gamma_3(m),\ 1 \le m \le i$, which triages $m-1$ jobs first, then serves a class-1 job, performs one more triage and then follows the optimal policy. The expected cost of policy $\pi_3(m)$ and $\gamma_3(m)$ are $C_{\pi_3(m)} = \psi + (r + r_1)u + c_1 + PC_1 \cdot r_1 T_1 + PC_2 \cdot r_2 T_1 + C_3$ and $C_{\gamma_3(m)} = \psi + c_1 + r(T_1 + u) + C_3$, where ψ denotes the expected cost incurred during the triage of the first $(m-1)$ jobs, C_3 denotes the expected waiting cost and service cost of the remaining jobs excluding the two jobs of which we exchanged their order in $\pi_3(m)$ and $\gamma_3(m)$. Note that these costs are the same in both policy $\pi_3(m)$ and $\gamma_3(m)$.

$$C_{\gamma_3(m)} - C_{\pi_3(m)} = (r - PC_1 r_1 - PC_2 r_2)T_1 - r_1 u = -r_1 u < 0,\ 1 \le m \le i.$$

Hence, policy $\pi_3(m)$ is outperformed by policy $\gamma_3(m)$ and can not be the optimal policy, $1 \le m \le i$. The optimal policy must be $\pi_3(0)$, i.e., the server should first serve a class-1 job instead of doing triage. This completes the proof of part *(i)*.

Part (ii): If $k_1 > 0$, then $a^*(i, k_1, k_2) =$ SC1 by part (i), hence we only need to consider the case where $k_1 = 0$ and $i > 0$ and $k_2 > 0$. We will show that SC2 is not the optimal decision, meaning that either Tr or S is more preferable than SC2, by induction on the number of remaining jobs k, as in the proof of part (i).

Suppose that $k = 2$, i.e. there is one class 2 and one untriaged job. Consider two policies: policy π serves the class-2 job, then serves the untriaged job without triage (since there is only one job left, doing triage is clearly inferior); policy γ first serves the untriaged job without triage then serves the class-2 job. The expected costs for policies π and γ are respectively

$$C_\pi = c_2 + rT_2 + c,\ C_\gamma = c + r_2 T + c_2,$$

and

$$C_\pi - C_\gamma = rT_2 - r_2 T = \frac{pq\tau_1\tau_2\,(v_1 + v_2 - 1)\,(h_1/\tau_1 - h_2/\tau_2)}{PC_2} > 0.$$

Now, assume $a^*(i, 0, k_2) \neq$ SC2 for some $i + k_2 = k \ge 2$. We will show that $a^*(i, 0, k_2) \neq$ SC2 when $i + k_2 = k + 1$. Suppose Policy π first serves a class-2 job in state $(i, 0, k_2)$, then follows the optimal policy. By the induction hypothesis, in $(i, 0, k_2 - 1)$ the optimal policy will work on an untriaged job, by either serving without triage (SU) or performing triage (Tr). Consider another policy

γ, which, when in state $(i, 0, k_2)$, does whatever π does in $(i, 0, k_2 - 1)$, then serves the class-2 job that π serves at $(i, 0, k_2)$ and goes on to follow policy π.

If π takes action SU in state $(i, 0, k_2 - 1)$, then the expected cost under policy π and γ are $C_\pi = c_2 + rT_2 + c + \Gamma_1$ and $C_\gamma = c + r_2T + c_2 + \Gamma_1$, where Γ_1 denotes the expected waiting and service cost incurred by the remaining $i - 1$ untriaged and $k_2 - 1$ class-2 jobs, which is the same under the two policies. Then,

$$C_\pi - C_\gamma = rT_2 - r_2T = \frac{pq\tau_1\tau_2\,(v_1 + v_2 - 1)\,(h_1/\tau_1 - h_2/\tau_2)}{PC_2} > 0. \tag{A.11}$$

If π takes action Tr in state $(i, 0, k_2 - 1)$, i.e., $a^*(i, 0, k_2 - 1) =$ Tr, first, by Lemma A.0.1 we must have $u < T$. The expected cost under policy π and γ are $C_\pi = c_2 + rT_2 + ru + \Gamma_2$ and $C_\gamma = ru + r_2u + c_2 + \Gamma_2$, where Γ_2 denotes the expected waiting and service cost incurred by the remaining jobs ($i - 1$ untriaged jobs, one job which has just been triaged, and $k_2 - 1$ already waiting class-2 jobs), which is the same under the two policies. Then,

$$C_\pi - C_\gamma = rT_2 - r_2u = rT_2 - r_2T + r_2T - r_2u > r_2(T - u) > 0,$$

where the first inequality follows from Assumption (3.2.1). Thus, we can conclude that SC2 is not an optimal action in state $(i, 0, k_2)$ when $i > 0$, i.e., $a^*(i, 0, k_2) \neq$ SC2 as long as $i > 0$. □

Proof of Theorem 3.4.2.

The proof will proceed by establishing a sequence of lemmas, which will eventually lead to the proof of the theorem.

Lemma A.0.2. *We have* $a^*(1, 0, k_2) =$ SU *for all* $k_2 \geq 0$.

Proof of Lemma A.0.2: In state $(1, 0, k_2)$, by Theorem 3.4.1, SC2 is suboptimal. Hence, the only possible optimal actions are Tr and SU. Let J_T denote the expected cost of taking action Tr next and then following the optimal policy until all jobs are served, and J_{NT} denote the expected cost of taking action SU next and then following the optimal policy until all jobs are served. Then,

$$
\begin{aligned}
J_T(1,0,k_2) &= PC_1V(0,1,k_2) + PC_2V(0,0,k_2+1) + (r+k_2r_2)u \\
&= PC_1(V(0,0,k_2) + c_1 + k_2r_2T_1) + PC_2(V(0,0,k_2) + c_2 + k_2r_2T_2) + (r+k_2r_2)u \\
&= V(0,0,k_2) + c + k_2r_2T + (r+k_2r_2)u, \\
J_{NT}(1,0,k_2) &= V(0,0,k_2) + c + k_2r_2T.
\end{aligned}
$$

Hence, $J_T(1,0,k_2) - J_{NT}(1,0,k_2) = (r+k_2r_2)u > 0$. $\square$

Lemma A.0.3. *For all* $(i,k_1,k_2) \in \mathcal{S}$*, we have*

$$
V(i,k_1,k_2+1) - V(i,k_1,k_2) \geq c_2 + r_2(iT + k_1T_1 + k_2T_2).
$$

Proof of Lemma A.0.3: By Theorem 3.4.1, we know that under the optimal policy, first, all class-1 jobs are served. Therefore,

$$
V(i,k_1,k_2+1) - V(i,k_1,k_2) = V(i,0,k_2+1) - V(i,0,k_2) + r_2k_1T_1. \tag{A.12}
$$

Define $\tilde{V}(i,0,k_2)$ for any $i+k_2 \geq 1$ as the total expected cost, starting from state $(i,0,k_2)$, under the policy that uses the action that is optimal for state $(\tilde{i},0,\tilde{k}_2+1)$ whenever the system state is $(\tilde{i},0,\tilde{k}_2)$. Thus, $\tilde{V}(i,0,k_2) \geq V(i,0,k_2)$ and therefore

$$
\begin{aligned}
V(i,0,k_2+1) - V(i,0,k_2) &= V(i,0,k_2+1) - \tilde{V}(i,0,k_2) + \tilde{V}(i,0,k_2) - V(i,0,k_2) \\
&\geq V(i,0,k_2+1) - \tilde{V}(i,0,k_2). \qquad \text{(A.13)}
\end{aligned}
$$

The only difference between $V(i,0,k_2+1)$ and $\tilde{V}(i,0,k_2)$ is the expected cost incurred by the extra class-2 job in the former, which includes the expected service cost plus the expected waiting cost during the service of the previous $i+k_2$ jobs. Now, the expected time for serving a job without triage, which is T, is less than that for first triaging then serving this job, which is $u + PC_1 \cdot T_1 + PC_2 \cdot T_2 = u + T$. Hence, the expected waiting time of the last class-2 job is greater than or equal to $iT + k_2T_2$.

Therefore,

$$V(i,0,k_2+1) - \tilde{V}(i,0,k_2) \geq c_2 + r_2(iT + k_2T_2). \tag{A.14}$$

Combining (A.12), (A.13) and (A.14),

$$V(i,k_1,k_2+1) - V(i,k_1,k_2) \geq c_2 + r_2(iT + k_1T_1 + k_2T_2).$$

$\square$

Lemma A.0.4. *If* $u \geq \tilde{u} = PC_2(rT_2 - r_2T)/r$*, then* $a^*(i,k_1,k_2) \neq \text{Tr}$ *for any* $(i,k_1,k_2) \in \mathcal{S}$.

Proof of Lemma A.0.4: Suppose the current state is (i,k_1,k_2). It is sufficient to consider the case $k_1 = 0$ and $i > 0$, because $a^*(i,k_1,k_2) = \text{SC1}$ when $k_1 > 0$ and Tr is not a feasible action when $i = 0$.

Suppose $k_1 = 0,\ i > 0$. Theorem 3.4.1 says that SC2 is suboptimal. Hence, the only possible optimal actions are Tr and SU. Let J_T denote the expected cost of choosing Tr first and then using the optimal policy until all jobs are served, and J_{NT} denote the expected cost of choosing SU first then using the optimal policy until all jobs are served. Thus, we have

$$\begin{aligned}
J_T(i,0,k_2) &= PC_1V(i-1,1,k_2) + PC_2V(i-1,0,k_2+1) + (ir + k_2r_2)u \\
&= PC_1[V(i-1,0,k_2) + c_1 + (i-1)rT_1 + k_2r_2T_1] + PC_2V(i-1,0,k_2+1) + (ir + k_2r_2)u, \\
J_{NT}(i,0,k_2) &= V(i-1,0,k_2) + c + [(i-1)r + k_2r_2]T.
\end{aligned}$$

Then,

$$\begin{aligned}
&J_T(i,0,k_2) - J_{NT}(i,0,k_2) \\
&= PC_2\left[V(i-1,0,k_2+1) - V(i-1,0,k_2)\right] - (c - PC_1c_1) - (i-1)r\left(T - PC_1T_1\right) \\
&\quad - k_2r_2(T - PC_1T_1) + (ir + k_2r_2)u \\
&= PC_2\left[V(i-1,0,k_2+1) - V(i-1,0,k_2)\right] - PC_2c_2 - [(i-1)r + k_2r_2]PC_2T_2 + (ir + k_2r_2)u \\
&= PC_2[V(i-1,0,k_2+1) - V(i-1,0,k_2) - c_2 - (i-1)rT_2] + (ir + k_2r_2)u - k_2r_2PC_2T_2.
\end{aligned}$$

Then, from Lemma A.0.3,

$$\begin{aligned}
&J_T(i,0,k_2) - J_{NT}(i,0,k_2) \\
&\geq PC_2[c_2 + (i-1)r_2T + r_2k_2T_2 - c_2 - (i-1)rT_2] + (ir + k_2r_2)u - k_2r_2PC_2T_2 \\
&= PC_2(i-1)(r_2T - rT_2) + (ir + k_2r_2)u.
\end{aligned}$$

By the assumption that $u \geq \tilde{u} = PC_2(rT_2 - r_2T)/r$,

$$\begin{aligned}
J_T(i,0,k_2) - J_{NT}(i,0,k_2) &\geq -PC_2(i-1)(rT_2 - r_2T) + (ir + k_2r_2)PC_2(rT_2 - r_2T)/r \\
&= [(ir + k_2r_2)/r - (i-1)]\, PC_2(rT_2 - r_2T) \\
&= (1 + k_2r_2/r)pq(v_1 + v_2 - 1)\tau_1\tau_2(h_1/\tau_1 - h_2/\tau_2) \\
&> 0,
\end{aligned}$$

where the inequality follows from Assumption 3.2.1. Hence, $a^*(i,k_1,k_2) \neq$ Tr, $\forall\ (i,k_1,k_2) \in \mathcal{S}$. $\square$

Lemma A.0.5. *If $a^*(i,0,k_2) =$ SU, then $a^*(j,0,k_2) =$ SU for all $1 \leq j \leq i$, i.e., the optimal policy starting from $(i,0,k_2)$ is "first serve all i untriaged jobs without triage, then serve all k_2 class-2 jobs."*

Proof of Lemma A.0.5: If $u \geq \tilde{u}$, by Theorem 3.4.1 and Lemma A.0.4, we know $a^*(i,0,k_2) =$ SU for all $i \geq 1, k_2 \geq 0$. Thus, the lemma holds trivially.

Let us now assume that $u < \tilde{u}$. Let policy π be the policy that first serves the i untriaged jobs without triage, then serves the k_2 class-2 jobs, and C_π denotes the expected total cost under policy π. Assume that policy π is not optimal, then there must exist at least one policy that does better than π and satisfies the following properties: *The policy first serves $1 \leq k \leq i-1$ untriaged jobs without triage, then performs triage for the next untriaged job. And in conformance with the properties of the optimal policy as established in Theorem 3.4.1, if the job that goes through triage is classified as class-1, γ_1 serves that job right away. Otherwise, the job is served at the end together with all the other class-2 jobs.* Suppose that among the policies which satisfy these properties, γ_1 is the policy for which k is the smallest, and let $k_{\min}$ denote that smallest value for k, i.e., γ_1 first serves $k_{\min}$ untriaged jobs without triage, then performs triage for the next job. Note that by definition, we have $C_{\gamma_1} < C_\pi$, where C_{γ_1} is the total expected cost under policy γ_1.

Now, consider another policy γ_2, which serves $k_{\min} - 1$ untriaged jobs without triage, performs triage on the next job, serves the next job without triage, and then takes the same actions as γ_1. As in γ_1, if triage results in identification of a class-1 job, that job is served immediately; otherwise, the job is served at the end with all the other class-2 jobs. Thus, the only difference between γ_1 and γ_2 is that while γ_1 serves the $k_{\min}$th untriaged job without triage and triages the $(k_{\min}+1)$th untriaged job, γ_2 triages the $k_{\min}$th untriaged job and serves the $(k_{\min}+1)$th untriaged job without triage. Since by definition, policy γ_1 is the one with the smallest k among those policies that perform better than policy π, we have

$$C_{\gamma_2} \geq C_\pi > C_{\gamma_1}. \tag{A.15}$$

If the only triaged job among the first $k_{\min}+1$ jobs is of class-1, denote Γ_1 as the expected total cost that will incur after the triage and service of the $(k_{\min}+1)$th job in policy γ_1 (or service without triage of the $(k_{\min}+1)$th job in policy γ_2). If the only triaged job is of class-2, denote Γ_2 as the expected total cost that will incur after the triage of the $(k_{\min}+1)$th job in policy γ_1 (or service without triage of the $(k_{\min}+1)$th job in policy γ_2). The total expected cost under policy γ_1 and γ_2 are respectively

$$\begin{aligned}
C_{\gamma_1} &= \Phi + c + \left[(i-k_{\min})r + k_2 r_2\right] T + \left[(i-k_{\min})r + k_2 r_2\right] u \\
&\quad + PC_1\left[c_1 + (i-k_{\min}-1)rT_1 + k_2 r_2 T_1 + \Gamma_1\right] + PC_2\Gamma_2, \\
C_{\gamma_2} &= \Phi + \left[(i-k_{\min}+1)r + k_2 r_2\right] u \\
&\quad + PC_1\left[c_1 + (i-k_{\min})rT_1 + k_2 r_2 T_1 + c + (i-k_{\min}-1)rT + k_2 r_2 T + \Gamma_1\right] \\
&\quad + PC_2\left[c + (i-k_{\min}-1)rT + (k_2+1)r_2 T + \Gamma_2\right],
\end{aligned}$$

where Φ is the total expected cost to be incurred during the service of the first $k_{\min}-1$ untriaged jobs without triage. Then,

$$C_{\gamma_1} - C_{\gamma_2} = -ru + rT - PC_2 r_2 T - PC_1 rT_1 = -ru + PC_2(rT_2 - r_2 T) = r(\tilde{u} - u) > 0.$$

Hence, $C_{\gamma_2} < C_{\gamma_1}$, which is a contradiction to (A.15). $\square$

Lemma A.0.6. *(i) If $a^*(i,0,k_2) = \text{SU}$, then $a^*(\tilde{i},0,\tilde{k}_2) = \text{SU}$ for any $1 \le \tilde{i} \le i$ and $k_2 \le \tilde{k}_2 \le N - \tilde{i}$.*

(ii) If $a^*(i,0,k_2) = \text{Tr}$, then $a^*(\tilde{i},0,\tilde{k}_2) = \text{Tr}$ for any $0 \le \tilde{k}_2 \le k_2$ and $i \le \tilde{i} \le N - \tilde{k}_2$.

Proof of Lemma A.0.6: Part (i): If $u \ge \tilde{u}$, by Theorem 3.4.1 and Lemma A.0.4 we know $a^*(i,0,k_2) = \text{SU}$ for all $i \ge 1, k_2 \ge 0$. Then, the result is immediate. Now, assume that $u < \tilde{u}$. We will use an induction argument to show that if $a^*(i,0,k_2) = \text{SU}$, then $a^*(i-1,0,k_2) = a^*(i,0,k_2+1) = \text{SU}$. When $i = 1$, the result holds since $a^*(1,0,k_2) = \text{SU}$ for any $k_2 \ge 0$ by Lemma A.0.2.

Now, for induction we assume that if $a^*(i-1,0,k_2) = \text{SU}$, then $a^*(i-2,0,k_2) = a^*(i-1,0,k_2+1) = \text{SU}$ where $i \ge 2$. From the lemma assumption, we have $a^*(i,0,k_2) = \text{SU}$. Then, by Lemma A.0.5, we know that $a^*(i-1,0,k_2) = \text{SU}$, and by the induction assumption, $a^*(i-1,0,k_2+1) = \text{SU}$. It remains to show that $a^*(i,0,k) = \text{SU}$ for any $k \ge k_2$. Let $C_{\text{Tr}}(i,k_2+1)$ denote the total expected cost of performing triage in state $(i,0,k_2+1)$ and then following the optimal policy. Then,

$$C_{\text{Tr}}(i,k_2+1) = [ir + (k_2+1)r_2]u + PC_1V(i-1,1,k_2+1) + PC_2V(i-1,0,k_2+2).$$

Similarly, let $C_{\text{SU}}(i,k_2+1)$ denote the total expected cost choosing to serve an untriaged job in state $(i,0,k_2+1)$ and then following the optimal policy. Then,

$$C_{\text{SU}}(i,k_2+1) = c + [(i-1)r + (k_2+1)r_2]T + V(i-1,0,k_2+1).$$

Using the induction assumption and Lemma A.0.5,

$$\begin{aligned}
V(i-1,0,k_2+1) &= (i-1)\left[c + \frac{i-2}{2}rT + (k_2+1)r_2T\right] + (k_2+1)c_2 + \frac{k_2(k_2+1)}{2}r_2T_2,\\
V(i-1,1,k_2+1) &= c_1 + [(i-1)r + (k_2+1)r_2]T_1 + (i-1)c\\
&\quad + \frac{(i-1)(i-2)}{2}rT + (k_2+1)r_2(i-1)T + (k_2+1)c_2 + \frac{k_2(k_2+1)}{2}r_2T_2,\\
&= c_1 + [(i-1)r + (k_2+1)r_2]T_1 + V(i-1,0,k_2+1),\\
V(i-1,0,k_2+2) &= (i-1)\left[c + \frac{i-2}{2}rT + (k_2+2)r_2T\right] + (k_2+2)c_2 + \frac{(k_2+1)(k_2+2)}{2}r_2T_2,
\end{aligned}$$

$$= c_2 + r_2[(i-1)T + (k_2+1)T_2] + V(i-1, 0, k_2+1).$$

Plugging them into the expression for $C_{\text{Tr}}(i, k_2+1)$,

$$\begin{aligned}
C_{\text{Tr}}(i, k_2+1) = &\ (ir + (k_2+1)r_2)\, u + PC_1\,(c_1 + [(i-1)r + (k_2+1)r_2]T_1 + V(i-1, 0, k_2+1)) \\
&+ PC_2\,(c_2 + r_2[(i-1)T + (k_2+1)T_2] + V(i-1, 0, k_2+1)) \\
= &\ [ir + (k_2+1)r_2]u + c + V(i-1, 0, k_2+1) \\
&+ PC_1\,[(i-1)r + (k_2+1)r_2]\,T_1 + PC_2 r_2\,[(i-1)T + (k_2+1)T_2] \\
= &\ [ir + (k_2+1)r_2]u + c + (k_2+1)r_2T + V(i-1, 0, k_2+1) \\
&+ PC_1(i-1)rT_1 + PC_2r_2(i-1)T.
\end{aligned}$$

Hence,

$$C_{\text{SU}}(i, k_2+1) - C_{Tr}(i, k_2+1) = (i-1)\,(rT - PC_1rT_1 - PC_2r_2T) - [ir + (k_2+1)r_2]u.$$

We know that $a^*(i, 0, k_2) = \text{SU}$, which implies

$$C_{\text{SU}}(i, k_2) - C_{\text{Tr}}(i, k_2) = (i-1)\,(rT - PC_1rT_1 - PC_2r_2T) - (ir + k_2r_2)u \leq 0.$$

Therefore, $C_{\text{SU}}(i, k_2+1) - C_{\text{Tr}}(i, k_2+1) < C_{\text{SU}}(i, k_2) - C_{\text{Tr}}(i, k_2) \leq 0$, i.e., $a^*(i, 0, k_2+1) = \text{SU}$.

Part (ii): Given part *(i)*, the proof of *(ii)* is trivial. Assume $a^*(i, 0, k_2) = \text{Tr}$, and there exists $\bar{i} > i$ (or $\bar{k}_2 < k_2$) such that $a^*(\bar{i}, 0, k_2) = \text{SU}$ (or $a^*(i, 0, \bar{k}_2) = \text{SU}$), which is a direct contradiction to *(i)*, which implies that $a^*(i, 0, k_2) = \text{SU}$. $\square$

Proof of Theorem 3.4.2: First, note that if $\tilde{u} < u$, then by Lemma A.0.4 and Theorem 3.4.1, the optimal action in all states is SU. One can check to see that when $\tilde{u} < u$, $L(\cdot)$ has a negative slope and thus the theorem trivially holds. Hence, in the following, it is sufficient to consider the case where $\tilde{u} \geq u$.

By Theorem 3.4.1, we can write the system states $\{(i, 0, k_2) : i \geq 1, k_2 \geq 0\}$ as the union of the following three disjoint sets:

$$\mathcal{S}_1 = \{(i,0,k_2) : a^*(i,0,k_2) = \text{SU}\},$$

$$\mathcal{S}_2 = \{(i,0,k_2) : a^*(i,0,k_2) = \text{Tr},\ a^*(i-1,0,k_2) = \text{SU}\},$$

$$S_3 = \{(i,0,k_2) : a^*(i,0,k_2) = \text{Tr},\ a^*(i-1,0,k_2) = \text{Tr}\}.$$

We show that all the states in $\mathcal{S}_1$ reside above $L(i)$ and all the states in $\mathcal{S}_2$ and $\mathcal{S}_3$ reside below $L(i)$.

First, suppose that $(i,0,k_2) \in \mathcal{S}_1$. Consider a policy γ that serves an untriaged job without triage in $(i,0,k_2)$, then follows the optimal policy. Consider another policy π that performs triage in $(i,0,k_2)$ then follows the optimal policy. The total expected costs, (respectively, $C_\gamma(i,0,k_2)$ and $C_\pi(i,0,k_2)$) can be written as follows:

$$\begin{aligned}
C_\gamma(i,0,k_2) &= c + [(i-1)r + k_2 r_2]T + V(i-1,0,k_2),\\
C_\pi(i,0,k_2) &= PC_1 V(i-1,1,k_2) + PC_2 V(i-1,0,k_2+1) + (ir + k_2 r_2)u\\
&= PC_1\Big(c_1 + [(i-1)r + k_2 r_2]\,T_1 + V(i-1,0,k_2)\Big) + PC_2 V(i-1,0,k_2+1)\\
&\quad\ + (ir + k_2 r_2)u\\
&= PC_1\Big(c_1 + [(i-1)r + k_2 r_2]\,T_1 + V(i-1,0,k_2)\Big) + (ir + k_2 r_2)u +\\
&\quad\ + PC_2\Big(V(i-1,0,k_2) + c_2 + r_2\,[(i-1)T + k_2 T_2]\Big)\\
&= V(i-1,0,k_2) + c + k_2 r_2 T + (i-1)(PC_1 r T_1 + PC_2 r_2 T) + (ir + k_2 r_2)u.
\end{aligned}$$

Since γ is the optimal policy, $C_\pi(i,0,k_2) - C_\gamma(i,0,k_2) = (ir + k_2 r_2)u - (i-1)PC_2(rT_2 - r_2 T) = (ir + k_2 r_2)u - (i-1)r\tilde{u} \geq 0$, i.e.,

$$k_2 \geq \left(\frac{r(\tilde{u} - u)}{r_2 u}\right) i - \frac{r\tilde{u}}{r_2 u}.$$

Now, suppose that $(i,0,k_2) \in \mathcal{S}_2$. Then, with γ and π exactly as defined above, policy π is the optimal policy, and thus $C_\pi(i,0,k_2) - C_\gamma(i,0,k_2) < 0$, i.e.,

$$k_2 < \left(\frac{r(\tilde{u} - u)}{r_2 u}\right) i - \frac{r\tilde{u}}{r_2 u}.$$

Finally, consider a state $(i,0,k_2) \in \mathcal{S}_3$. Then, from Lemma A.0.6, we know that there exists

$\bar{i} < i$ such that $(\bar{i}, 0, k_2) \in \mathcal{S}_2$. Then since $\tilde{u} - u \geq 0$ and we know that, as established above, $k_2 < \left(\frac{r(\tilde{u}-u)}{r_2 u}\right)\bar{i} - \frac{r\tilde{u}}{r_2 u}$, we must have

$$k_2 < \left(\frac{r(\tilde{u}-u)}{r_2 u}\right) i - \frac{r\tilde{u}}{r_2 u},$$

which completes the proof. $\square$

Proof of Proposition 3.4.1.

Part (i): From Theorem 3.4.2, we know that NT policy is optimal if and only if either $u \geq \tilde{u}$ (so that $L(\cdot)$, as defined in (3.4), has a negative slope) or $L(\cdot)$ has a positive slope but the x-intercept of $L(\cdot)$, denoted by x_{int}, is greater than N, which can be written as $x_{int} = \frac{\tilde{u}}{\tilde{u}-u} \geq N$, or equivalently $u \geq u_1 = \frac{N-1}{N}\tilde{u}$. Then, the result follows from the fact that if $u \geq \tilde{u}$ then we must have $u \geq u_1$.

Part (ii): From Theorems 3.4.1 and 3.4.2, we know that TP_1 policy is optimal if and only if $u \leq \tilde{u}$, $1 \leq x_{int} \leq 2$, and the x-coordinate of the intersection of line $L(\cdot)$ and the line expressed by $i+k_2 = N$ (where i is the number of untriaged jobs and k_2 is the number of class-2 jobs) is also between 1 and 2. The last two conditions can be expressed as $1 \leq \frac{\tilde{u}}{\tilde{u}-u} \leq 2$ and $1 \leq \frac{Nr_2u+r\tilde{u}}{(r_2-r)u+r\tilde{u}} \leq 2$, respectively. First, if $u \leq \tilde{u}$, then $1 \leq \frac{\tilde{u}}{\tilde{u}-u}$. It also follows that $1 \leq \frac{Nr_2u+r\tilde{u}}{(r_2-r)u+r\tilde{u}}$. When $u \leq \tilde{u}$, the condition $\frac{\tilde{u}}{\tilde{u}-u} \leq 2$ can equivalently be written as $u \leq \tilde{u}/2$ and the condition $\frac{Nr_2u+r\tilde{u}}{(r_2-r)u+r\tilde{u}} \leq 2$ can be written as $u \leq u_2 = \frac{r}{2r+(N-2)r_2}\tilde{u}$. The last condition implies $u \leq \tilde{u}/2$ which completes the proof. $\square$

APPENDIX B: PROOF OF RESULTS IN CHAPTER 4

Proof of Proposition 4.3.1.

Assume the system is empty at time 0. Consider a policy π that serves every job without triage. The system acts as an $M/M/1$ queue and the expected average cost is $g_\pi = \lambda\tau/(1-\lambda\tau)\cdot r_0 < \infty$. If we apply the policy iteration algorithm with initial policy π, Theorem 5.1 in Meyn (1997) states that the policy iteration algorithm will guarantee to find the optimal solution to the Bellman's equations. Theorem 5.2 in Meyn (1997) implies that there exists an optimal stationary policy. □

Proof of Proposition 4.3.3.

This is an minimization problem and the cost at each state is bounded below, the positivity assumption is satisfied. The action set at each state is finite, hence, by Proposition 3.1.6 in Bertsekas (2007), the above result holds. □

Proof of Lemma 4.3.1.

Proof of Property (e.1) and (e.2).

Consider two possible cases: $k_1 \geq 2$ and $k_1 = 1$. If $k_1 \geq 2$,

$$Tv(i-1,k_1+1,k_2) = \lambda v(i,k_1+1,k_2) + \frac{1}{\tau}v(i-1,k_1,k_2) + \frac{1}{u}v(i-1,k_1+1,k_2) + (i-1)r$$
$$+ (k_1+1)r_1 + k_2r_2.$$
$$Tv(i-1,k_1,k_2+1) = \lambda v(i,k_1,k_2+1) + \frac{1}{\tau}v(i-1,k_1-1,k_2+1) + \frac{1}{u}v(i-1,k_1,k_2+1)$$
$$+ (i-1)r + k_1r_1 + (k_2+1)r_2.$$
$$Tv(i,k_1,k_2) = \lambda v(i+1,k_1,k_2) + \frac{1}{\tau}v(i,k_1-1,k_2) + \frac{1}{u}v(i,k_1,k_2) + (ir + k_1r_1 + k_2r_2)$$
$$Tv(i,k_1-1,k_2) = \lambda v(i+1,k_1-1,k_2) + \frac{1}{\tau}v(i,k_1-2,k_2) + \frac{1}{u}v(i,k_1-1,k_2) + ir$$
$$+ (k_1-1)r_1 + k_2r_2.$$
$$Tv(i-1,k_1,k_2) = \lambda v(i,k_1,k_2) + \frac{1}{\tau}v(i-1,k_1-1,k_2) + \frac{1}{u}v(i-1,k_1,k_2) + (i-1)r$$

$+ k_1 r_1 + k_2 r_2.$

Therefore,

$$\begin{aligned}
&\frac{1}{u}\big[PC_1 Tv(i-1,k_1+1,k_2)+PC_2 Tv(i-1,k_1,k_2+1)\big]+\frac{1}{\tau}Tv(i,k_1,k_2)\\
&-\frac{1}{u}Tv(i,k_1,k_2)-\frac{1}{\tau}Tv(i,k_1-1,k_2)\\
&=\lambda\Big\{\frac{1}{u}\big[PC_1 v(i,k_1+1,k_2)+PC_2 v(i,k_1,k_2+1)\big]+\frac{1}{\tau}v(i+1,k_1,k_2)-\frac{1}{u}v(i+1,k_1,k_2)\\
&-\frac{1}{u}v(i+1,k_1,k_2)\Big\}+\frac{1}{\tau}\Big\{\frac{1}{u}\big[PC_1 v(i-1,k_1,k_2)+PC_2 v(i-1,k_1-1,k_2+1)\big]\\
&+\frac{1}{\tau}v(i,k_1-1,k_2)-\frac{1}{u}v(i,k_1-1,k_2)-\frac{1}{\tau}v(i,k_1-2,k_2)\Big\}+\frac{1}{u}\Big\{\frac{1}{u}\big[PC_1 v(i-1,k_1+1,k_2)\\
&+PC_2 v(i-1,k_1,k_2+1)\big]+\frac{1}{\tau}v(i,k_1,k_2)-\frac{1}{u}v(i,k_1,k_2)-\frac{1}{\tau}v(i,k_1-1,k_2)\Big\}+\frac{r_1}{\tau}>0.
\end{aligned}$$

$$\begin{aligned}
&Tv(i-1,k_1,k_2)-Tv(i,k_1-1,k_2)\\
&=\lambda[v(i,k_1,k_2)-v(i+1,k_1-1,k_2)]+\frac{1}{\tau}[v(i-1,k_1-1,k_2)-v(i,k_1-2,k_2)]\\
&+\frac{1}{u}[v(i-1,k_1,k_2)-v(i,k_1-1,k_2)]+(r_1-r)>0.
\end{aligned}$$

If $k_1 = 1$, then,

$$\begin{aligned}
&Tv(i-1,2,k_2)=\lambda v(i,2,k_2)+\frac{1}{\tau}v(i-1,1,k_2)+\frac{1}{u}v(i-1,2,k_2)+(i-1)r+2r_1+k_2r_2,\\
&Tv(i-1,1,k_2+1)=\lambda v(i,1,k_2+1)+\frac{1}{\tau}v(i-1,0,k_2+1)+\frac{1}{u}v(i-1,1,k_2+1)\\
&\quad+(i-1)r+r_1+(k_2+1)r_2,\\
&Tv(i,1,k_2)=\lambda v(i+1,1,k_2)+\frac{1}{\tau}v(i,0,k_2)+\frac{1}{u}v(i,1,k_2)+(ir+r_1+k_2r_2),\\
&Tv(i,0,k_2)=\lambda v(i+1,0,k_2)+\min\Big\{\frac{1}{u}[PC_1 v(i-1,1,k_2)+PC_2 v(i-1,0,k_2+1)]\\
&+\frac{1}{\tau}v(i,0,k_2),\frac{1}{\tau}v(i-1,0,k_2)+\frac{1}{u}v(i,0,k_2)\Big\}+(ir+k_2r_2).
\end{aligned}$$

Hence,

$$\frac{1}{u}\big[PC_1 Tv(i-1,2,k_2)+PC_2 Tv(i-1,1,k_2+1)\big]+\frac{1}{\tau}Tv(i,1,k_2)-\frac{1}{u}Tv(i,1,k_2)-\frac{1}{\tau}Tv(i,0,k_2)$$

$$
\begin{aligned}
=&\lambda\Big[\frac{1}{u}\big[PC_1v(i,2,k_2)+PC_2v(i,1,k_2+1)\big]+\frac{1}{\tau}v(i+1,1,k_2)-\frac{1}{u}v(i+1,1,k_2)-\frac{1}{u}v(i+1,1,k_2)\Big]\\
+&\frac{1}{\tau}\bigg[\frac{1}{u}[PC_1v(i-1,1,k_2)+PC_2v(i-1,0,k_2+1)]+\frac{1}{\tau}v(i,0,k_2)-\min\Big\{\frac{1}{u}[PC_1v(i-1,1,k_2)\\
+&PC_2v(i-1,0,k_2+1)]+\frac{1}{\tau}v(i,0,k_2),\frac{1}{\tau}v(i-1,0,k_2)+\frac{1}{u}v(i,0,k_2)\Big\}\bigg]+\frac{r_1}{\tau}\\
+&\frac{1}{u}\Big\{\frac{1}{u}[PC_1v(i-1,2,k_2)+PC_2v(i-1,1,k_2+1)]+\frac{1}{\tau}v(i,1,k_2)-\frac{1}{u}v(i,1,k_2)\Big\}\geq\frac{r_1}{\tau}>0.
\end{aligned}
$$

$$
\begin{aligned}
&Tv(i-1,1,k_2)-Tv(i,0,k_2)\\
\geq&\lambda[v(i,1,k_2)-v(i+1,0,k_2)]+\frac{1}{\tau}[v(i-1,0,k_2)-v(i-1,0,k_2)]\\
+&\frac{1}{u}[v(i-1,k_1,k_2)-v(i,0,k_2)]+(r_1-r)>0.
\end{aligned}
$$

Proof of Property (e.3).

Now we show that Property **(e.3)** is preserved by considering three cases:

Case (e.3)-1 $k_1\geq 2$,

$$
\begin{aligned}
&Tv(i,k_1,k_2-1)-Tv(i,k_1-1,k_2)\\
=&\big[\lambda v(i+1,k_1,k_2-1)+\frac{1}{\tau}v(i,k_1-1,k_2-1)+\frac{1}{u}v(i,k_1,k_2-1)+ir+k_1r_1+(k_2-1)r_2\big]\\
-&\big[\lambda v(i+1,k_1-1,k_2)+\frac{1}{\tau}v(i,k_1-2,k_2)+\frac{1}{u}v(i,k_1-1,k_2)+ir+(k_1-1)r_1+k_2r_2\big]\\
=&\lambda[v(i+1,k_1,k_2-1)-v(i+1,k_1-1,k_2)]+\frac{1}{\tau}[v(i,k_1-1,k_2-1)-v(i,k_1-2,k_2)]\\
+&\frac{1}{u}[v(i,k_1,k_2-1)-v(i,k_1-1,k_2)]+(r_1-r_2)>0.
\end{aligned}
$$

Case (e.3)-2 $k_1=1, i\geq 1$,

$$
\begin{aligned}
&Tv(i,1,k_2-1)-Tv(i,0,k_2)\\
=&\big[\lambda v(i+1,1,k_2-1)+\frac{1}{\tau}v(i,0,k_2-1)+\frac{1}{u}v(i,1,k_2-1)+ir+r_1+(k_2-1)r_2\big]\\
-&\bigg[\lambda v(i+1,0,k_2)+\min\Big\{\frac{1}{u}[PC_1v(i-1,1,k_2)+PC_2v(i-1,0,k_2+1)]+\frac{1}{\tau}v(i,0,k_2),\\
&\frac{1}{\tau}v(i-1,0,k_2)+\frac{1}{u}v(i,0,k_2)\Big\}+(ir+k_2r_2)\bigg]
\end{aligned}
$$

$$\geq\lambda[v(i+1,k_1,k_2-1)-v(i+1,0,k_2)]+\frac{1}{\tau}[v(i,0,k_2-1)-v(i-1,0,k_2)]$$
$$+\frac{1}{u}[v(i,1,k_2-1)-v(i,0,k_2)]+(r_1-r_2)>0.$$

Case (e.3)-3 $k_1=1, i=0,$

$$\begin{aligned}
&Tv(0,1,k_2-1)-Tv(0,0,k_2)\\
=&\big[\lambda v(1,1,k_2-1)+\frac{1}{\tau}v(0,0,k_2-1)+\frac{1}{u}v(0,1,k_2-1)+r_1+(k_2-1)r_2\big]\\
-&\big[\lambda v(1,0,k_2)+\frac{1}{\tau}v(0,0,k_2-1)+\frac{1}{u}v(0,0,k_2)+k_2r_2\big]\\
\geq&\lambda[v(1,1,k_2-1)-v(1,0,k_2)]+\frac{1}{u}[v(0,1,k_2-1)-v(0,0,k_2)]+(r_1-r_2)>0.
\end{aligned}$$

Proof of Property (e.4).

Now we show that Property **(e.4)** is preserved by considering three cases: **Case (e.4)-1** $k_1\geq 1,$

$$\begin{aligned}
&Tv(i,k_1,k_2-1)-Tv(i-1,k_1,k_2)\\
=&\big[\lambda v(i+1,k_1,k_2-1)+\frac{1}{\tau}v(i,k_1-1,k_2-1)+\frac{1}{u}v(i,k_1,k_2-1)+ir+k_1r_1+(k_2-1)r_2\big]\\
-&\big[\lambda v(i,k_1,k_2)+\frac{1}{\tau}v(i-1,k_1-1,k_2)+\frac{1}{u}v(i-1,k_1,k_2)+(i-1)r+k_1r_1+k_2r_2\big]\\
=&\lambda[v(i+1,k_1,k_2-1)-v(i,k_1,k_2)]+\frac{1}{\tau}[v(i,k_1-1,k_2-1)-v(i-1,k_1-1,k_2)]\\
+&\frac{1}{u}[v(i,k_1,k_2-1)-v(i-1,k_1,k_2)]+(r-r_2)>0.
\end{aligned}$$

Case (e.4)-2 $k_1=0, i\geq 2,$

$$\begin{aligned}
&Tv(i,0,k_2-1)-Tv(i-1,0,k_2)\\
=&\Big[\lambda v(i+1,0,k_2-1)+\min\Big\{\frac{1}{u}\big[PC_1v(i-1,1,k_2-1)+PC_2v(i-1,0,k_2)\big]\\
&+\frac{1}{\tau}v(i,0,k_2-1),\frac{1}{\tau}v(i-1,0,k_2-1)+\frac{1}{u}v(i,0,k_2-1)\Big\}+ir+(k_2-1)r_2\Big]\\
-&\Big[\lambda v(i,0,k_2)+\min\Big\{\frac{1}{u}\big[PC_1v(i-2,1,k_2)+PC_2v(i-2,0,k_2+1)\big]+\frac{1}{\tau}v(i-1,0,k_2),\\
&\frac{1}{\tau}v(i-2,0,k_2)+\frac{1}{u}v(i-1,0,k_2)\Big\}+(i-1)r+k_2r_2\Big]\\
>&\lambda[v(i+1,0,k_2-1)-v(i,0,k_2)]+(r-r_2)>0.
\end{aligned}$$

The first inequality holds since by **(e.2)**$\sim$**(e.4)**,

$$v(i-1,0,k_2)] > v(i-2,0,k_2+1),\ v(i-1,1,k_2-1) > v(i-2,1,k_2),$$
$$v(i,0,k_2-1) > v(i-1,0,k_2),\ v(i-1,0,k_2-1) > v(i-2,0,k_2).$$

Case (e.4)-3 $k_1 = 0, i = 1,$

$$\begin{aligned}
&Tv(1,0,k_2-1) - Tv(0,0,k_2)\\
=&\lambda v(2,0,k_2-1) + \min\Big\{\frac{1}{u}[PC_1 v(0,1,k_2-1) + PC_2 v(0,0,k_2)] + \frac{1}{\tau}v(1,0,k_2-1),\\
&\frac{1}{\tau}v(0,0,k_2-1) + \frac{1}{u}v(1,0,k_2-1)\Big\} + r + (k_2-1)r_2 - \Big[\lambda v(1,0,k_2) + \frac{1}{\tau}v(0,0,k_2-1)\\
+&\frac{1}{u}v(0,0,k_2) + k_2 r_2\Big]\\
>&\lambda[v(2,0,k_2-1) - v(0,0,k_2)] + (r - r_2) > 0.
\end{aligned}$$

The first inequality holds since by **(e.2)**$\sim$**(e.5)**,

$$v(0,1,k_2-1) > v(0,0,k_2) > v(0,0,k_2-1),$$
$$v(1,0,k_2-1) > v(0,0,k_2) > v(0,0,k_2-1).$$

Proof of Property (e.5).

If $k_1 \geq 1$,

$$\begin{aligned}
&Tv(i,k_1,k_2+1) - Tv(i,k_1,k_2)\\
=&\lambda v(i+1,k_1,k_2+1) + ir + k_1 r_1 + (k_2+1)r_2 + \frac{1}{\tau}v(i,k_1-1,k_2+1) + \frac{1}{u}v(i,k_1,k_2+1)\\
-&\Big[\lambda v(i+1,k_1,k_2) + ir + k_1 r_1 + k_2 r_2 + \frac{1}{\tau}v(i,k_1-1,k_2) + \frac{1}{u}v(i,k_1,k_2)\Big]\\
=&\lambda\big[v(i+1,k_1,k_2+1) - v(i+1,k_1,k_2)\big] + r_2 + \frac{1}{\tau}\big[v(i,k_1-1,k_2+1) - v(i,k_1-1,k_2)\big]\\
+&\frac{1}{u}\big[v(i,k_1,k_2+1) - v(i,k_1,k_2)\big] > r_2 > 0.
\end{aligned}$$

Next we consider the case that $k_1 = 0$. If $i = 0,\ k_2 = 0$, then

$$\begin{aligned}
&Tv(0,0,1) - Tv(0,0,0) \\
=&\lambda v(1,0,1) + r_2 + \frac{1}{\tau}v(0,0,0) + \frac{1}{u}v(0,0,1) - \Big[\lambda v(1,0,0) + \frac{1}{\tau}v(0,0,0) + \frac{1}{u}v(0,0,0)\Big] \\
=&\lambda\big[v(1,0,1) - v(1,0,0)\big] + \frac{1}{u}\big[v(0,0,1) - v(0,0,0)\big] + r_2 > r_2 > 0.
\end{aligned}$$

If $i = 0,\ k_2 \geq 1$, then

$$\begin{aligned}
&Tv(0,0,k_2+1) - Tv(0,0,k_2) \\
=&\Big[\lambda v(1,0,k_2+1) + (k_2+1)r_2 + \frac{1}{\tau}v(0,0,k_2) + \frac{1}{u}v(0,0,k_2+1)\Big] - \Big[\lambda v(1,0,k_2) \\
+&\frac{1}{\tau}v(0,0,k_2-1) + \frac{1}{u}v(0,0,k_2) + k_2 r_2\Big] \\
=&\lambda\big[v(1,0,k_2+1) - v(1,0,k_2)\big] + \frac{1}{\tau}\big[v(0,0,k_2) - v(0,0,k_2-1)\big] \\
+&\frac{1}{u}\big[v(0,0,k_2+1) - v(0,0,k_2)\big] + r_2 > r_2 > 0.
\end{aligned}$$

Otherwise, $i \geq 1,\ k_2 \geq 0$. We consider two separate cases:

(i) $G(i, k_2+1) \geq 0$.

$$\begin{aligned}
&Tv(i,0,k_2+1) - Tv(i,0,k_2) \\
\geq&\lambda v(i+1,0,k_2+1) + ir + (k_2+1)r_2 + \frac{1}{\tau}v(i-1,0,k_2+1) + \frac{1}{u}v(i,0,k_2+1) \\
-&\Big[\lambda v(i+1,0,k_2) + ir + k_2 r_2 + \frac{1}{\tau}v(i-1,0,k_2) + \frac{1}{u}v(i,0,k_2)\Big] \\
=&\lambda\big[v(i+1,0,k_2+1) - v(i+1,0,k_2)\big] + r_2 + \frac{1}{\tau}\big[v(i-1,0,k_2+1) - v(i-1,0,k_2)\big] \\
+&\frac{1}{u}\big[v(i,0,k_2+1) - v(i,0,k_2)\big] > r_2 > 0.
\end{aligned}$$

(ii) $G(i, k_2+1) < 0$.

$$\begin{aligned}
&Tv(i,0,k_2+1) - Tv(i,0,k_2) \\
=&\Big[\lambda v(i+1,0,k_2+1) + \frac{1}{u}[PC_1 v(i-1,1,k_2+1) + PC_2 v(i-1,0,k_2+2)]
\end{aligned}$$

$$
\begin{aligned}
&+\frac{1}{\tau}v(i,0,k_2+1)+ir+(k_2+1)r_2)\Big]\\
&-\Big[\lambda v(i+1,0,k_2)+\min\Big\{\frac{1}{u}[PC_1v(i-1,1,k_2)+PC_2v(i-1,0,k_2+1)]\\
&+\frac{1}{\tau}v(i,0,k_2),\frac{1}{\tau}v(i-1,0,k_2)+\frac{1}{u}v(i,0,k_2)\Big\}+(ir+k_2r_2)\Big]\\
&\geq\lambda\big[v(i+1,0,k_2+1)-v(i+1,0,k_2)\big]+\frac{PC_1}{u}\big[v(i-1,1,k_2+1)-v(i-1,1,k_2)\big]\\
&+\frac{PC_2}{u}\big[v(i-1,0,k_2+2)-v(i-1,0,k_2+1)\big]+\frac{1}{\tau}\big[v(i,0,k_2+1)-v(i,0,k_2)\big]+r_2\\
&>r_2>0.
\end{aligned}
$$

Proof of Property (f.1).

$$
\begin{aligned}
&Tv(i,1,k_2)-Tv(i+1,0,k_2)\\
=&\Big[\lambda v(i+1,1,k_2)+\frac{1}{\tau}v(i,0,k_2)+\frac{1}{u}v(i,1,k_2)+ir+r_1+k_2r_2\Big]-\Big[\lambda v(i+2,0,k_2)\\
&+\min\Big\{\frac{1}{u}\Big[PC_1v(i,1,k_2)+PC_2v(i,0,k_2+1)\Big]+\frac{1}{\tau}v(i+1,0,k_2),\frac{1}{\tau}v(i,0,k_2)\\
&+\frac{1}{u}v(i+1,0,k_2)\Big\}+(i+1)r+k_2r_2\Big]\\
=&\lambda[v(i+1,1,k_2)-v(i+2,0,k_2)]-\min\big\{G(i+1,k_2),0\big\}+\frac{1}{u}[v(i,1,k_2)-v(i+1,0,k_2)]\\
&+r_1-r\\
=&\lambda[v(i+1,1,k_2)-v(i+2,0,k_2)]+\frac{1}{u}[v(i,1,k_2)-v(i+1,0,k_2)]+r_1-r.
\end{aligned}
$$

Hence,

$$
\begin{aligned}
&\frac{PC_1}{u}\big[Tv(i,1,k_2)-Tv(i+1,0,k_2)\big]\\
=&\lambda\frac{PC_1}{u}[v(i+1,1,k_2)-v(i+2,0,k_2)]+\frac{1}{u}\frac{PC_1}{u}[v(i,1,k_2)-v(i+1,0,k_2)]+\frac{PC_1}{u}(r_1-r)\\
\leq&(\lambda+1/u)r+\frac{PC_1(r_1-r)}{u}=r+\frac{r(1+\alpha\tau)}{\tau u}(\tilde{u}(\alpha)-u)\leq r.
\end{aligned}
$$

The last inequality holds because of our assumption $u<\tilde{u}(\alpha)$.

Proof of Property (f.2).

We show that Property (**f.2**) is preserved by considering three cases:

Case (f.2)-1 $i = 1,\ k_2 = 0,$

$$
\begin{aligned}
& TG(1,0) \\
=& \frac{1}{u}[PC_1 Tv(0,1,0) + PC_2 Tv(0,0,1)] + \frac{1}{\tau}Tv(1,0,0) - \frac{1}{u}Tv(1,0,0) - \frac{1}{\tau}Tv(0,0,0) \\
=& \frac{PC_1}{u}\Big[\lambda v(1,1,0) + \frac{1}{\tau}v(0,0,0) + \frac{1}{u}v(0,1,0) + r_1\Big] + \frac{PC_2}{u}\Big[\lambda v(1,0,1) + \frac{1}{\tau}v(0,0,0) + \frac{1}{u}v(0,0,1) \\
+& r_2\Big] - \left(\frac{1}{u} - \frac{1}{\tau}\right)\Bigg\{\lambda v(2,0,0) + r + \min\Bigg\{\frac{1}{u}\Big[PC_1 v(0,1,0) + PC_2 v(0,0,1)\Big] + \frac{1}{\tau}v(1,0,0), \\
& \frac{1}{\tau}v(0,0,0) + \frac{1}{u}v(1,0,0)\Bigg\}\Bigg\} - \frac{1}{\tau}\Big[\lambda v(1,0,0) + \frac{1}{\tau}v(0,0,0) + \frac{1}{u}v(0,0,0)\Big] \\
=& \lambda G(2,0) + \frac{1}{u}\max\big\{G(1,0),0\big\} + \frac{1}{\tau}\min\big\{G(1,0),0\big\} + \frac{r}{\tau} \geq \frac{r}{\tau} > 0.
\end{aligned}
$$

Case (f.2)-2 $i = 1,\ k_2 \geq 1,$

$$
\begin{aligned}
& TG(1,k_2) \\
=& \frac{1}{u}[PC_1 Tv(0,1,k_2) + PC_2 Tv(0,0,k_2+1)] + \frac{1}{\tau}Tv(1,0,k_2) - \frac{1}{u}Tv(1,0,k_2) - \frac{1}{\tau}Tv(0,0,k_2) \\
=& \lambda G(2,k_2) + \frac{1}{u}\max\big\{G(1,k_2),0\big\} + \frac{1}{\tau}\min\big\{G(1,k_2),0\big\} + \frac{1}{\tau^2}\Big[v(0,0,k_2) - v(0,0,k_2-1)\Big] + \frac{r}{\tau} \\
\geq& \frac{1}{\tau^2}\Big[v(0,0,k_2) - v(0,0,k_2-1)\Big] \geq 0.
\end{aligned}
$$

Case (f.2)-3 $i \geq 2,\ k_2 \geq 0,$

$$
\begin{aligned}
& TG(i,k_2) \\
=& \frac{1}{u}[PC_1 Tv(i-1,1,k_2) + PC_2 Tv(i-1,0,k_2+1)] + \frac{1}{\tau}Tv(i,0,k_2) - \frac{1}{u}Tv(i,0,k_2) \\
-& \frac{1}{\tau}Tv(i-1,0,k_2) \\
=& \lambda G(i+1,k_2) + \frac{1}{u}\max\big\{G(i,k_2),0\big\} + \frac{1}{\tau}\min\big\{G(i,k_2),0\big\} + \frac{PC_2}{u}\min\big\{G(i-1,k_2+1),0\big\} \\
+& \frac{1}{\tau}\max\big\{G(i-1,k_2),0\big\} + \frac{PC_1}{\tau u}\Big[v(i-1,0,k_2) - v(i-2,1,k_2)\Big] + \frac{r}{\tau} \\
\geq& \frac{PC_1}{\tau u}\Big[v(i-1,0,k_2) - v(i-2,1,k_2)\Big] + \frac{r}{\tau} = \frac{1}{\tau}\Big(r - \frac{PC_1}{u}\big[v(i-2,1,k_2) - v(i-1,0,k_2)\big]\Big)
\end{aligned}
$$

$\geq 0.$

The last inequality holds because of (**f.1**). $\square$

Proof of Lemma 4.3.2.

The proof of the preservation on Property (**e.1**)$\sim$(**e.5**) is exactly the same as in Lemma 4.3.1. We only present the proof for Property (**g.1**)$\sim$(**g.4**).

Proof of Property (g.1).

We first show Property (**g.1**) by considering three possible cases:

Case (g.1)-1 $i \geq 2,\ k_2 \geq 0,$

$$
\begin{aligned}
&Tv(i-1,1,k_2) = \lambda v(i,1,k_2) + \frac{1}{\tau}v(i-1,0,k_2) + \frac{1}{u}v(i-1,1,k_2) + (i-1)r + r_1 + k_2 r_2,\\
&Tv(i,0,k_2) = \lambda v(i+1,0,k_2) + \min\Big\{\frac{1}{u}\Big[PC_1 v(i-1,1,k_2) + PC_2 v(i-1,0,k_2+1)\Big]\\
&+\frac{1}{\tau}v(i,0,k_2),\ \frac{1}{\tau}v(i-1,0,k_2) + \frac{1}{u}v(i,0,k_2)\Big\} + ir + k_2 r_2\\
&= \lambda v(i+1,0,k_2) + \min\big\{G(i,k_1),0\big\} + \frac{1}{\tau}v(i-1,0,k_2) + \frac{1}{u}v(i,0,k_2) + ir + k_2 r_2,\\
&Tv(i-1,0,k_2+1) = \lambda v(i,0,k_2+1) + \min\big\{G(i-1,k_2+1),0\big\} + \frac{1}{\tau}v(i-2,0,k_2+1)\\
&+\frac{1}{u}v(i-1,0,k_2+1) + (i-1)r + (k_2+1)r_2,\\
&Tv(i-1,0,k_2) = \lambda v(i,0,k_2) + \min\big\{G(i-1,k_2),0\big\} + \frac{1}{\tau}v(i-2,0,k_2) + \frac{1}{u}v(i-1,0,k_2)\\
&+(i-1)r + k_2 r_2.
\end{aligned}
$$

Hence,

$$
\begin{aligned}
&TG(i,k_2) = \frac{1}{u}[PC_1 Tv(i-1,1,k_2) + PC_2 Tv(i-1,0,k_2+1)] + \frac{1}{\tau}Tv(i,0,k_2) - \frac{1}{u}Tv(i,0,k_2)\\
&-\frac{1}{\tau}Tv(i-1,0,k_2)\\
&= \lambda G(i+1,k_2) + \frac{1}{u}\max\big\{G(i,k_2),0\big\} + \frac{1}{\tau}\min\big\{G(i,k_2),0\big\} + \frac{PC_2}{u}\min\big\{G(i-1,k_2+1),0\big\}\\
&\quad + \frac{1}{\tau}\max\big\{G(i-1,k_2),0\big\} + \frac{PC_1}{\tau u}\Big[v(i-1,0,k_2) - v(i-2,1,k_2)\Big] + \frac{r}{\tau}.
\end{aligned}
$$

$$
\begin{aligned}
&TG(i,k_2+1) = \lambda G(i+1,k_2+1) + \frac{1}{u}\max\left\{G(i,k_2+1),0\right\} + \frac{1}{\tau}\min\left\{G(i,k_2+1),0\right\} \\
&+ \frac{PC_2}{u}\min\left\{G(i-1,k_2+2),0\right\} + \frac{1}{\tau}\max\left\{G(i-1,k_2+1),0\right\} \\
&+ \frac{PC_1}{\tau u}\Big[v(i-1,0,k_2+1) - v(i-2,1,k_2+1)\Big] + \frac{r}{\tau}.
\end{aligned}
$$

By **(g.1)** & **(g.2)**, $TG(i,k_2) \le TG(i,k_2+1)$.

Case (g.1)-2 $i=1,\ k_2 \ge 1$,

$$
\begin{aligned}
&Tv(0,1,k_2) = \lambda v(1,1,k_2) + \frac{1}{\tau}v(1,0,k_2) + \frac{1}{u}v(0,1,k_2) + r_1 + k_2 r_2, \\
&Tv(0,0,k_2+1) = \lambda v(1,0,k_2+1) + \frac{1}{\tau}v(0,0,k_2) + \frac{1}{u}v(0,0,k_2+1) + (k_2+1)r_2, \\
&Tv(1,0,k_2) = \lambda v(2,0,k_2) + \min\left\{G(1,k_1),0\right\} + \frac{1}{\tau}v(0,0,k_2) + \frac{1}{u}v(1,0,k_2) + r + k_2 r_2, \\
&Tv(0,0,k_2) = \lambda v(1,0,k_2) + \frac{1}{\tau}v(0,0,k_2-1) + \frac{1}{u}v(0,0,k_2) + k_2 r_2. \\
&TG(1,k_2) \\
=&\frac{1}{u}[PC_1 Tv(0,1,k_2) + PC_2 Tv(0,0,k_2+1)] + \frac{1}{\tau}Tv(1,0,k_2) - \frac{1}{u}Tv(1,0,k_2) - \frac{1}{\tau}Tv(0,0,k_2) \\
=&\lambda G(2,k_2) + \frac{1}{u}\max\left\{G(1,k_2),0\right\} + \frac{1}{\tau}\min\left\{G(1,k_2),0\right\} + \frac{1}{\tau^2}\Big[v(0,0,k_2) - v(0,0,k_2-1)\Big] + \frac{r}{\tau}, \\
&TG(1,k_2+1) = \lambda G(2,k_2+1) + \frac{1}{u}\max\left\{G(1,k_2+1),0\right\} + \frac{1}{\tau}\min\left\{G(1,k_2+1),0\right\} \\
&+ \frac{1}{\tau^2}\Big[v(0,0,k_2+1) - v(0,0,k_2)\Big] + \frac{r}{\tau}.
\end{aligned}
$$

By **(g.1)** & **(g.3)**,

$$
TG(1,k_2) - TG(1,k_2+1) \le \frac{1}{\tau^2}\Big[v(0,0,k_2) - v(0,0,k_2-1)\Big] - \frac{1}{\tau^2}\Big[v(0,0,k_2+1) - v(0,0,k_2)\Big] \le 0.
$$

Case (g.1)-3 $i=1,\ k_2=0$,

$$
\begin{aligned}
&Tv(0,1,0) = \lambda v(1,1,0) + \frac{1}{\tau}v(1,0,0) + \frac{1}{u}v(0,1,0) + r_1, \\
&Tv(0,0,1) = \lambda v(1,0,1) + \frac{1}{\tau}v(0,0,0) + \frac{1}{u}v(0,0,1) + r_2, \\
&Tv(1,0,0) = \lambda v(2,0,0) + \min\left\{G(1,0),0\right\} + \frac{1}{\tau}v(0,0,0) + \frac{1}{u}v(1,0,0) + r,
\end{aligned}
$$

$$
\begin{aligned}
&Tv(0,0,0) = \lambda v(1,0,0) + \frac{1}{\tau} v(0,0,0) + \frac{1}{u} v(0,0,0).\\
&TG(1,0) = \frac{1}{u}[PC_1 Tv(0,1,0) + PC_2 Tv(0,0,1)] + \frac{1}{\tau} Tv(1,0,0) - \frac{1}{u} Tv(1,0,0) - \frac{1}{\tau} Tv(0,0,0)\\
&\qquad\quad = \lambda G(2,0) + \frac{1}{u} \max\left\{G(1,0), 0\right\} + \frac{1}{\tau} \min\left\{G(1,0), 0\right\} + \frac{r}{\tau}.
\end{aligned}
$$

$$
TG(1,1) = \lambda G(2,1) + \frac{1}{u} \max\left\{G(1,1), 0\right\} + \frac{1}{\tau} \min\left\{G(1,1), 0\right\} + \frac{1}{\tau^2}\Big[v(0,0,1) - v(0,0,0)\Big] + \frac{r}{\tau}.
$$

By **(g.1**) & **(e.5**), $TG(1,0) - TG(1,1) \leq -\frac{1}{\tau^2}\big[v(0,0,1) - v(0,0,0)\big] \leq 0.$

Proof of Property (g.2).

$$
\begin{aligned}
&Tv(i+1,0,k_2) - Tv(i,1,k_2)\\
=&\lambda v(i+2,0,k_2) + (i+1)r + k_2 r_2 + \min\Big\{\frac{1}{u}\Big[PC_1 v(i,1,k_2) + PC_2 v(i,0,k_2+1)\Big]\\
+&\frac{1}{\tau} v(i+1,0,k_2), \frac{1}{\tau} v(i,0,k_2) + \frac{1}{u} v(i+1,0,k_2)\Big\} - \Big[\lambda v(i+1,1,k_2) + \frac{1}{\tau} v(i,0,k_2)\\
+&\frac{1}{u} v(i,1,k_2) + ir + r_1 + k_2 r_2\Big]\\
=&\lambda[v(i+2,0,k_2) - v(i+1,1,k_2)] + \min\left\{G(i+1,k_2), 0\right\}\\
+&\frac{1}{u}[v(i+1,0,k_2) - v(i,1,k_2)] + (r - r_1),
\end{aligned}
$$

$$
\begin{aligned}
&Tv(i+1,0,k_2+1) - Tv(i,1,k_2+1)\\
=&\lambda[v(i+2,0,k_2+1) - v(i+1,1,k_2+1)] + \min\left\{G(i+1,k_2+1), 0\right\}\\
+&\frac{1}{u}[v(i+1,0,k_2+1) - v(i,1,k_2+1)] + (r - r_1),
\end{aligned}
$$

By **(g.1**) and **(g.2**), it is obvious that

$$
Tv(i+1,0,k_2) - Tv(i,1,k_2) \leq Tv(i+1,0,k_2+1) - Tv(i,1,k_2+1).
$$

Proof of Property (g.3).

First we consider $i = 0,\ k_2 = 0$.

$$
\begin{aligned}
& Tv(0,0,1) - Tv(0,0,0) \\
=& \Big[\lambda v(1,0,1) + r_2 + \frac{1}{\tau} v(0,0,0) + \frac{1}{u} v(0,0,1)\Big] - \Big[\lambda v(1,0,0) + \frac{1}{\tau} v(0,0,0) + \frac{1}{u} v(0,0,0)\Big] \\
=& \lambda\big[v(1,0,1) - v(1,0,0)\big] + \frac{1}{u}\big[v(0,0,1) - v(0,0,0)\big] + r_2,
\end{aligned}
$$

$$
\begin{aligned}
& Tv(0,0,2) - Tv(0,0,1) \\
=& \Big[\lambda v(1,0,2) + 2r_2 + \frac{1}{\tau} v(0,0,1) + \frac{1}{u} v(0,0,2)\Big] - \Big[\lambda v(1,0,1) + r_2 + \frac{1}{\tau} v(0,0,0) + \frac{1}{u} v(0,0,1)\Big] \\
=& \lambda\big[v(1,0,2) - v(1,0,1)\big] + \frac{1}{\tau}\big[v(0,0,1) - v(0,0,0)\big] + \frac{1}{u}\big[v(0,0,2) - v(0,0,1)\big] + r_2.
\end{aligned}
$$

Hence, by **(e.5)** & **(g.3)**, $Tv(0,0,1) - Tv(0,0,0) \le Tv(0,0,2) - Tv(0,0,1)$. When $i = 0,\ k_2 \ge 1$,

$$
\begin{aligned}
& Tv(0,0,k_2+1) - Tv(0,0,k_2) \\
=& \Big[\lambda v(1,0,k_2+1) + (k_2+1)r_2 + \frac{1}{\tau} v(0,0,k_2) + \frac{1}{u} v(0,0,k_2+1)\Big] - \Big[\lambda v(1,0,k_2) + k_2 r_2 \\
&+ \frac{1}{\tau} v(0,0,k_2-1) + \frac{1}{u} v(0,0,k_2)\Big] \\
=& \lambda\big[v(1,0,k_2+1) - v(1,0,k_2)\big] + \frac{1}{\tau}\big[v(0,0,k_2) - v(0,0,k_2-1)\big] + \frac{1}{u}\big[v(0,0,k_2+1) \\
&- v(0,0,k_2)\big] + r_2,
\end{aligned}
$$

$$
\begin{aligned}
& Tv(0,0,k_2+2) - Tv(0,0,k_2+1) \\
=& \lambda\Big[v(1,0,k_2+2) - v(1,0,k_2+1)\Big] + \frac{1}{\tau}\Big[v(0,0,k_2+1) - v(0,0,k_2)\Big] \\
&+ \frac{1}{u}\Big[v(0,0,k_2+2) - v(0,0,k_2+1)\Big] + r_2.
\end{aligned}
$$

Hence, by **(g.3)**, $Tv(0,0,k_2+1) - Tv(0,0,k_2) \le Tv(0,0,k_2+2) - Tv(0,0,k_2+1)$. When $i \ge 1,\ k_2 \ge 0$,

$$
Tv(i,0,k_2+1) - Tv(i,0,k_2)
$$

$$=\Big[\lambda v(i+1,0,k_2+1)+ir+(k_2+1)r_2+\min\Big\{\frac{1}{u}\Big[PC_1v(i-1,1,k_2+1)$$
$$+PC_2v(i-1,0,k_2+2)\Big]+\frac{1}{\tau}v(i,0,k_2+1),\ \frac{1}{\tau}v(i-1,0,k_2+1)+\frac{1}{u}v(i,0,k_2+1)\Big\}\Big]$$
$$-\Big[\lambda v(i+1,0,k_2)+ir+k_2r_2+\min\Big\{\frac{1}{u}\Big[PC_1v(i-1,1,k_2)+PC_2v(i-1,0,k_2+1)\Big]$$
$$+\frac{1}{\tau}v(i,0,k_2),\frac{1}{\tau}v(i-1,0,k_2)+\frac{1}{u}v(i,0,k_2)\Big\}\Big]$$
$$=\lambda\big[v(i+1,0,k_2+1)-v(i+1,0,k_2)\big]+r_2+\frac{1}{\tau}\big[v(i-1,0,k_2+1)-v(i-1,0,k_2)\big]$$
$$+\frac{1}{u}\big[v(i,0,k_2+1)-v(i,0,k_2)\big]+\min\big\{G(i,k_2+1),0\big\}-\min\big\{G(i,k_2),0\big\},$$

$$Tv(i,0,k_2+2)-Tv(i,0,k_2+1)$$
$$=\lambda\big[v(i+1,0,k_2+2)-v(i+1,0,k_2+1)\big]+r_2+\frac{1}{\tau}\big[v(i-1,0,k_2+2)-v(i-1,0,k_2+1)\big]$$
$$+\frac{1}{u}\big[v(i,0,k_2+2)-v(i,0,k_2+1)\big]+\min\big\{G(i,k_2+2),0\big\}-\min\big\{G(i,k_2+1),0\big\},$$

We look at three separate cases:

(i) $G(i,k_2+2)\geq 0,\ G(i,k_2)\geq 0$, by **(g.3)** it is obvious that

$$\big[Tv(i,0,k_2+2)-Tv(i,0,k_2+1)\big]-\big[Tv(i,0,k_2+1)-Tv(i,0,k_2)\big]\geq 0.$$

(ii) $G(i,k_2+2)\geq 0,\ G(i,k_2)<0$,

$$Tv(i,0,k_2+1)-Tv(i,0,k_2)$$
$$\leq\lambda\big[v(i+1,0,k_2+1)-v(i+1,0,k_2)\big]+r_2+\frac{1}{u}\big[PC_1v(i-1,1,k_2+1)$$
$$+PC_2v(i-1,0,k_2+2)\big]+\frac{1}{\tau}v(i,0,k_2+1)-\frac{1}{u}\big[PC_1v(i-1,1,k_2)$$
$$+PC_2v(i-1,0,k_2+1)\big]-\frac{1}{\tau}v(i,0,k_2)$$
$$=\lambda\big[v(i+1,0,k_2+1)-v(i+1,0,k_2)\big]+r_2+\frac{PC_1}{u}\big[v(i-1,1,k_2+1)-v(i-1,1,k_2)\big]$$
$$+\frac{PC_2}{u}\big[v(i-1,0,k_2+2)-v(i-1,0,k_2+1)\big]+\frac{1}{\tau}\big[v(i,0,k_2+1)-v(i,0,k_2)\big],$$

$$Tv(i,0,k_2+2)-Tv(i,0,k_2+1)$$

$$
\geq \lambda \big[v(i+1,0,k_2+2) - v(i+1,0,k_2+1)\big] + r_2 + \frac{1}{\tau}\big[v(i-1,0,k_2+2)
$$
$$
-v(i-1,0,k_2+1)\big] + \frac{1}{u}\big[v(i,0,k_2+2) - v(i,0,k_2+1)\big].
$$

By **(g.2)**, **(g.3)** & **(g.4)**

$$
\begin{aligned}
&\big[Tv(i,0,k_2+2) - Tv(i,0,k_2+1)\big] - \big[Tv(i,0,k_2+1) - Tv(i,0,k_2)\big] \\
\geq &\lambda \big[v(i+1,0,k_2+2) - v(i+1,0,k_2+1) - v(i+1,0,k_2+1) + v(i+1,0,k_2)\big] \\
+&\frac{1}{\tau}\big[v(i-1,0,k_2+2) - v(i-1,0,k_2+1)\big] + \frac{1}{u}\big[v(i,0,k_2+2) - v(i,0,k_2+1)\big] \\
&- \frac{PC_1}{u}\big[v(i-1,1,k_2+1) - v(i-1,1,k_2)\big] - \frac{PC_2}{u}\big[v(i-1,0,k_2+2) \\
-&v(i-1,0,k_2+1)\big] - \frac{1}{\tau}\big[v(i,0,k_2+1) - v(i,0,k_2)\big] \\
\geq &\frac{PC_1}{u}\big[\big(v(i,0,k_2+2) - v(i,0,k_2+1)\big) - \big(v(i,0,k_2+1) - v(i,0,k_2)\big)\big] \\
+&\frac{PC_1}{u}\big[\big(v(i,0,k_2+1) - v(i,0,k_2)\big) - \big(v(i-1,1,k_2+1) - v(i-1,1,k_2)\big)\big] \\
+&\frac{PC_2}{u}\big[\big(v(i,0,k_2+2) - v(i,0,k_2+1)\big) - \big(v(i-1,0,k_2+2) - v(i-1,0,k_2+1)\big)\big] \\
&- \frac{1}{\tau}\big[\big(v(i,0,k_2+1) - v(i,0,k_2)\big) - \big(v(i-1,0,k_2+2) - v(i-1,0,k_2+1)\big)\big] \\
\geq &\left(\frac{PC_2}{u} - \frac{1}{\tau}\right)\big[\big(v(i,0,k_2+1) - v(i,0,k_2)\big) - \big(v(i-1,0,k_2+2) - v(i-1,0,k_2+1)\big)\big] \\
\geq &0.
\end{aligned}
$$

In the last inequality, $PC_2/u > 1/\tau$ because of $u < \tilde{u}(\alpha)$.

(iii) $G(i,k_2) < 0,\ G(i,k_2+2) < 0$

$$
\begin{aligned}
&Tv(i,0,k_2+1) - Tv(i,0,k_2) \\
=&\lambda \Big[v(i+1,0,k_2+1) - v(i+1,0,k_2)\Big] + r_2 \\
+&\frac{1}{u}\Big[PC_1 v(i-1,1,k_2+1) + PC_2 v(i-1,0,k_2+2)\Big] + \frac{1}{\tau}v(i,0,k_2+1) \\
-&\frac{1}{u}\Big[PC_1 v(i-1,1,k_2) + PC_2 v(i-1,0,k_2+1)\Big] - \frac{1}{\tau}v(i,0,k_2).
\end{aligned}
$$

$$
Tv(i,0,k_2+2) - Tv(i,0,k_2+1)
$$

$$
\begin{aligned}
=&\lambda\Big[v(i+1,0,k_2+2)-v(i+1,0,k_2+1)\Big]+r_2+\frac{1}{\tau}v(i,0,k_2+2)\\
+&\frac{1}{u}\Big[PC_1v(i-1,1,k_2+2)+PC_2v(i-1,0,k_2+3)\Big]\\
-&\frac{1}{u}\Big[PC_1v(i-1,1,k_2+1)+PC_2v(i-1,0,k_2+2)\Big]-\frac{1}{\tau}v(i,0,k_2+1),
\end{aligned}
$$

By **(g.3)**, it is obvious that

$$
\begin{aligned}
&\big[Tv(i,0,k_2+2)-Tv(i,0,k_2+1)\big]-\big[Tv(i,0,k_2+1)-Tv(i,0,k_2)\big]\\
=&\lambda\big[v(i+1,0,k_2+2)-v(i+1,0,k_2+1)-v(i+1,0,k_2+1)+v(i+1,0,k_2)\big]\\
+&\frac{PC_1}{u}\big[v(i-1,1,k_2+2)-v(i-1,1,k_2+1)-v(i-1,1,k_2+1)+v(i-1,1,k_2)\big]\\
+&\frac{PC_2}{u}\big[v(i-1,0,k_2+3)-v(i-1,0,k_2+2)-v(i-1,0,k_2+2)+v(i-1,0,k_2+1)\big]\\
+&\frac{1}{\tau}\big[v(i,0,k_2+2)-v(i,0,k_2+1)-v(i,0,k_2+1)+v(i,0,k_2)\big]\geq 0.
\end{aligned}
$$

Proof of Property (g.4).

For the cases that $k_1 \geq 1$,

$$
\begin{aligned}
&\big[Tv(i,k_1,k_2+1)-Tv(i,k_1,k_2)\big]-\big[Tv(i-1,k_1,k_2+2)-Tv(i-1,k_1,k_2+1)\big]\\
=&\lambda\Big(\big[v(i+1,k_1,k_2+1)-v(i+1,k_1,k_2)\big]-\big[v(i,k_1,k_2+2)-v(i,k_1,k_2+1)\big]\Big)\\
+&\frac{1}{u}\Big(\big[v(i,k_1,k_2+1)-v(i,k_1,k_2)\big]-\big[v(i-1,k_1,k_2+2)-v(i-1,k_1,k_2+1)\big]\Big)\\
+&\frac{1}{\tau}\Big(\big[v(i,k_1-1,k_2+1)-v(i,k_1-1,k_2)\big]-\big[v(i-1,k_1-1,k_2+2)\\
-&v(i-1,k_1-1,k_2+1)\big]\Big)\geq 0.
\end{aligned}
$$

Next, we assume that $k_1 = 0$. First we consider $i = 1,\ k_2 \geq 0$.

$$
\begin{aligned}
&\big[Tv(1,0,k_2+1)-Tv(1,0,k_2)\big]-\big[Tv(0,0,k_2+2)-Tv(0,0,k_2+1)\big]\\
=&\lambda\Big(\big[v(2,0,k_2+1)-v(2,0,k_2)\big]-\big[v(1,0,k_2+2)-v(1,0,k_2+1)\big]\Big)\\
+&\frac{1}{u}\Big(\big[v(1,0,k_2+1)-v(1,0,k_2)\big]-\big[v(0,0,k_2+2)-v(0,0,k_2+1)\big]\Big)\\
&+\min\{G(i,k_2+1),0\}-\min\{G(i,k_2),0\}\geq 0.
\end{aligned}
$$

For $i \geq 2,\ k_2 \geq 0$.

$$\begin{aligned}
&\big[Tv(i,0,k_2+1) - Tv(i,0,k_2)\big] - \big[Tv(i-1,0,k_2+2) - Tv(i-1,0,k_2+1)\big] \\
=&\lambda\Big(\big[v(i+1,0,k_2+1) - v(i+1,0,k_2)\big] - \big[v(i,0,k_2+2) - v(i,0,k_2+1)\big]\Big) \\
+&\frac{1}{\tau}\Big(\big[v(i-1,0,k_2+1) - v(i-1,0,k_2)\big] - \big[v(i-2,0,k_2+2) - v(i-2,0,k_2+1)\big]\Big) \\
+&\frac{1}{u}\Big(\big[v(i,0,k_2+1) - v(i,0,k_2)\big] - \big[v(i-1,0,k_2+2) - v(i-1,0,k_2+1)\big]\Big) \\
+&\min\{G(i,k_2+1),0\} + \min\{G(i-1,k_2+1),0\} - \min\{G(i,k_2),0\} - \min\{G(i-1,k_2+2),0\}.
\end{aligned}$$

If $G(i,k_2+1) > 0,\ G(i-1,k_2+1) > 0$,

$$\begin{aligned}
&\big[Tv(i,0,k_2+1) - Tv(i,0,k_2)\big] - \big[Tv(i-1,0,k_2+2) - Tv(i-1,0,k_2+1)\big] \\
=&\lambda\Big(\big[v(i+1,0,k_2+1) - v(i+1,0,k_2)\big] - \big[v(i,0,k_2+2) - v(i,0,k_2+1)\big]\Big) \\
+&\frac{1}{\tau}\Big(\big[v(i-1,0,k_2+1) - v(i-1,0,k_2)\big] - \big[v(i-2,0,k_2+2) - v(i-2,0,k_2+1)\big]\Big) \\
+&\frac{1}{u}\Big(\big[v(i,0,k_2+1) - v(i,0,k_2)\big] - \big[v(i-1,0,k_2+2) - v(i-1,0,k_2+1)\big]\Big) \geq 0.
\end{aligned}$$

If $G(i,k_2+1) < 0,\ G(i-1,k_2+1) > 0$, then $G(i,k_2) < 0,\ G(i-1,k_2+2) > 0$, hence

$$\begin{aligned}
&\big[Tv(i,0,k_2+1) - Tv(i,0,k_2)\big] - \big[Tv(i-1,0,k_2+2) - Tv(i-1,0,k_2+1)\big] \\
=&\lambda\Big(\big[v(i+1,0,k_2+1) - v(i+1,0,k_2)\big] - \big[v(i,0,k_2+2) - v(i,0,k_2+1)\big]\Big) \\
+&\frac{1}{\tau}\Big(\big[v(i-1,0,k_2+1) - v(i-1,0,k_2)\big] - \big[v(i-2,0,k_2+2) - v(i-2,0,k_2+1)\big]\Big) \\
+&\frac{1}{u}\Big(\big[v(i,0,k_2+1) - v(i,0,k_2)\big] - \big[v(i-1,0,k_2+2) - v(i-1,0,k_2+1)\big]\Big) \\
+&G(i,k_2+1) - G(i,k_2) > 0.
\end{aligned}$$

Otherwise,

$$\begin{aligned}
&\big[Tv(i,0,k_2+1) - Tv(i,0,k_2)\big] - \big[Tv(i-1,0,k_2+2) - Tv(i-1,0,k_2+1)\big] \\
\geq&\lambda\Big(\big[v(i+1,0,k_2+1) - v(i+1,0,k_2)\big] - \big[v(i,0,k_2+2) - v(i,0,k_2+1)\big]\Big)
\end{aligned}$$

$$
\begin{aligned}
&+\frac{1}{\tau}\Big(\big[v(i,0,k_2+1)-v(i,0,k_2)\big]-\big[v(i-1,0,k_2+2)-v(i-1,0,k_2+1)\big]\Big)\\
&+\frac{PC_1}{u}\Big(\big[v(i-1,1,k_2+1)-v(i-1,1,k_2)\big]-\big[v(i-2,1,k_2+2)-v(i-2,1,k_2+1)\big]\Big)\\
&+\frac{PC_2}{u}\Big(\big[v(i-1,0,k_2+2)-v(i-1,0,k_2+1)\big]-\big[v(i-2,0,k_2+3)-v(i-2,0,k_2+2)\big]\Big)\\
&\geq 0.
\end{aligned}
$$

Lemma B.0.7. *Assume*

$$
\lambda \leq \frac{\tau-u}{\tau^2}\left[1-\frac{\tau}{\tau-u}\frac{r_2}{\left(\frac{\tilde{u}(0)}{u}-1\right)r+r_2}\right], \tag{B.1}
$$

then β_1, β_2*, as defined in* (4.6) *and* (4.7)*, satisfy*

$$
\begin{aligned}
&\frac{r_2}{\alpha}(1-\beta_1) \leq \left(\frac{PC_2}{u}-\frac{1}{\tau}\right)(r-r_2)\tau^2 && \text{(B.2)}\\
&0<\beta_i<1,\ i=1,2. && \text{(B.3)}\\
&\lambda\beta_1(1-\beta_2)-\frac{1}{\tau}(1-\beta_1)+\alpha\beta_1 \leq 0 && \text{(B.4)}\\
&\lambda\beta_2(1-\beta_2)-\frac{1}{\tau}(1-\beta_2)+\alpha\beta_2 \leq 0 && \text{(B.5)}\\
&\lambda\beta_2(1-\beta_2)-\frac{1}{u}(\beta_1-\beta_2)+\alpha\beta_2 \leq 0 && \text{(B.6)}
\end{aligned}
$$

Proof. We can rewrite (B.2)

$$
\beta_1(\alpha) \geq 1-\frac{\alpha\tau^2}{r_2}\left(\frac{PC_2}{u}-\frac{1}{\tau}\right)(r-r_2). \tag{B.7}
$$

Here we emphasize the dependence on α by using $\beta_1(\alpha)$ instead of β_1. For convenience, we denote the right-hand-side of (B.7) as $f(\alpha)$. By assumption, we know $\lambda\tau/(\tau-u) < 1/\tau$. Hence, it is easy to show that $\beta_1(0) = f(0) = 1$.

$$
\begin{aligned}
\beta_1'(\alpha) &= \frac{1}{2\lambda}\left[1-\frac{\frac{\lambda\tau}{\tau-u}+1/\tau+\alpha}{\sqrt{(\frac{\lambda\tau}{\tau-u}+1/\tau+\alpha)^2-\frac{4\lambda}{\tau-u}}}\right] \leq 0,\ \beta_1'(0) = -\frac{\frac{\tau}{\tau-u}}{1/\tau-\frac{\lambda\tau}{\tau-u}},\\
f'(\alpha) &= -\frac{\tau^2}{r_2}\left(\frac{PC_2}{u}-\frac{1}{\tau}\right)(r-r_2) = f'(0).
\end{aligned}
$$

By the assumption in (B.1), $\beta_1'(0) \geq f'(0)$. We conclude that $\beta_1'(\alpha) \geq f'(\alpha)$ for all $\alpha \in [0,1]$ because

$$\beta_1''(\alpha) \quad = \quad \frac{2/(\tau-u)}{\left[(\frac{\lambda\tau}{\tau-u}+1/\tau+\alpha)^2-\frac{4\lambda}{\tau-u}\right]^{3/2}} > f''(\alpha) = 0.$$

Hence, $\beta_1(\alpha) \geq f(\alpha)$ for all $\alpha \in [0,1]$.

Since $\beta_1(0) = 1$ and $\beta_1'(\alpha) \leq 0$, it is obvious that $\beta_1(\alpha) < 1$ for any $\alpha > 0$. The positivity of $\beta_1(\alpha)$ is also obvious. The expression of β_2 implies that $\beta_2 < 1$.

Plug (4.7) into (B.4), we get

$$-\frac{\lambda\tau}{\tau-u}\beta_1^2 + \left(\frac{\lambda\tau}{\tau-u}+\frac{1}{\tau}+\alpha\right)\beta_1 - \frac{1}{\tau} \leq 0.$$

It is easy to check that the expression of β_1 in (4.6) is a solution to

$$-\frac{\lambda\tau}{\tau-u}\beta_1^2 + \left(\frac{\lambda\tau}{\tau-u}+\frac{1}{\tau}+\alpha\right)\beta_1 - \frac{1}{\tau} = 0.$$

Hence, (B.4) holds. Take the differences of the left-hand sides of (B.4) & (B.5), (B.4) & (B.6),

$$\begin{aligned}
&\left[\lambda\beta_1(1-\beta_2) - \frac{1}{\tau}(1-\beta_1) + \alpha\beta_1\right] - \left[\lambda\beta_2(1-\beta_2) - \frac{1}{\tau}(1-\beta_2) + \alpha\beta_2\right] \\
=&\lambda(1-\beta_2)(\beta_1-\beta_2) + \frac{1}{\tau}(\beta_1-\beta_2) + \alpha(\beta_1-\beta_2) > 0. \\
&\left[\lambda\beta_1(1-\beta_2) - \frac{1}{\tau}(1-\beta_1) + \alpha\beta_1\right] - \left[\lambda\beta_2(1-\beta_2) - \frac{1}{u}(\beta_1-\beta_2) + \alpha\beta_2\right] \\
=&\lambda(1-\beta_2)(\beta_1-\beta_2) + \alpha(\beta_1-\beta_2) > 0.
\end{aligned}$$

Hence, (B.5) and (B.6) hold. □

Proof of Lemma 4.3.3.

The proof of the preservation on Property **(e.1)**∼**(e.5)** is exactly the same as in Lemma 4.3.2. We only present the proof for Property **(h.1)**∼**(h.7)**.

Proof of Property (h.1).

We consider three cases:

Case (h.1)-1 $i \geq 2,\ k_2 \geq 0,$

$$
\begin{aligned}
&TG(i,k_2)\\
=&\lambda G(i+1,k_2) + \frac{1}{u}\max\big\{G(i,k_2),0\big\} + \frac{1}{\tau}\min\big\{G(i,k_2),0\big\} + \frac{PC_2}{u}\min\big\{G(i-1,k_2+1),0\big\}\\
+&\frac{1}{\tau}\max\big\{G(i-1,k_2),0\big\} + \frac{PC_1}{\tau u}\Big[v(i-1,0,k_2) - v(i-2,1,k_2)\Big] + \frac{r}{\tau}.
\end{aligned}
$$

$$
\begin{aligned}
&TG(i+1,k_2)\\
=&\lambda G(i+2,k_2) + \frac{1}{u}\max\big\{G(i+1,k_2),0\big\} + \frac{1}{\tau}\min\big\{G(i+1,k_2),0\big\} + \frac{PC_2}{u}\min\big\{G(i,k_2+1),0\big\}\\
+&\frac{1}{\tau}\max\big\{G(i,k_2),0\big\} + \frac{PC_1}{\tau u}\Big[v(i,0,k_2) - v(i-1,1,k_2)\Big] + \frac{r}{\tau}.
\end{aligned}
$$

By (**h.1**) & (**h.2**), $TG(i,k_2) \geq TG(i+1,k_2)$.

Case (h.1)-2 $i = 1,\ k_2 \geq 1,$

$$
\begin{aligned}
TG(1,k_2) =& \lambda G(2,k_2) + \frac{1}{u}\max\big\{G(1,k_2),0\big\} + \frac{1}{\tau}\min\big\{G(1,k_2),0\big\} + \frac{r}{\tau}\\
&+ \frac{1}{\tau^2}\Big[v(0,0,k_2) - v(0,0,k_2-1)\Big],
\end{aligned}
$$

$$
\begin{aligned}
TG(2,k_2) =& \lambda G(3,k_2) + \frac{1}{u}\max\big\{G(2,k_2),0\big\} + \frac{1}{\tau}\min\big\{G(2,k_2),0\big\} + \frac{PC_2}{u}\min\big\{G(1,k_2+1),0\big\}\\
&+ \frac{1}{\tau}\max\big\{G(1,k_2),0\big\} - \frac{PC_1}{\tau u}\Big[v(0,1,k_2) - v(1,0,k_2)\Big] + \frac{r}{\tau}.
\end{aligned}
$$

If $G(1,k_2) \leq 0$, by (**e.2**), (**e.5**) & (**h.1**),

$$
TG(2,k_2) - TG(1,k_2) \leq -\frac{PC_1}{\tau u}\Big[v(0,1,k_2) - v(1,0,k_2)\Big] - \frac{1}{\tau^2}\Big[v(0,0,k_2) - v(0,0,k_2-1)\Big] < 0.
$$

If $G(1,k_2) > 0$, by (**h.1**), (**h.3**) & (**h.4**),

$$
TG(2,k_2) - TG(1,k_2)
$$

$$\leq \frac{1}{\tau}G(1,k_2) - \frac{PC_1}{\tau u}\Big[v(0,1,k_2) - v(1,0,k_2)\Big] - \frac{1}{\tau^2}\Big[v(0,0,k_2) - v(0,0,k_2-1)\Big] \leq 0.$$

Case (h.1)-3 $i = 1,\ k_2 = 0,$

$$\begin{aligned}
&TG(1,0)\\
=&\frac{1}{u}[PC_1Tv(0,1,0) + PC_2Tv(0,0,1)] + \frac{1}{\tau}Tv(1,0,0) - \frac{1}{u}Tv(1,0,0) - \frac{1}{\tau}Tv(0,0,0)\\
=&\frac{PC_1}{u}\Big[\lambda v(1,1,0) + \frac{1}{\tau}v(0,0,0) + \frac{1}{u}v(0,1,0) + r_1\Big] + \frac{PC_2}{u}\Big[\lambda v(1,0,1) + \frac{1}{\tau}v(0,0,0) + r_2\\
&+\frac{1}{u}v(0,0,1)\Big] - \left(\frac{1}{u} - \frac{1}{\tau}\right)\Big\{\lambda v(2,0,0) + r + \min\Big\{\frac{1}{u}\Big[PC_1v(0,1,0) + PC_2v(0,0,1)\Big]\\
&+\frac{1}{\tau}v(1,0,0), \frac{1}{\tau}v(0,0,0) + \frac{1}{u}v(1,0,0)\Big\}\Big\} - \frac{1}{\tau}\Big[\lambda v(1,0,0) + \frac{1}{\tau}v(0,0,0) + \frac{1}{u}v(0,0,0)\Big]\\
=&\lambda G(2,0) + \frac{1}{u}\max\big\{G(1,0),0\big\} + \frac{1}{\tau}\min\big\{G(1,0),0\big\} + \frac{r}{\tau}.\\
&TG(2,0)\\
=&\lambda G(3,0) + \frac{1}{u}\max\big\{G(2,0),0\big\} + \frac{1}{\tau}\min\big\{G(2,0),0\big\} + \frac{PC_2}{u}\min\big\{G(1,1),0\big\}\\
&+\frac{1}{\tau}\max\big\{G(1,0),0\big\} + \frac{PC_1}{\tau u}\Big[v(1,0,0) - v(0,1,0)\Big] + \frac{r}{\tau}.
\end{aligned}$$

(i) If $G(1,0) \leq 0$, by **(h.1)** & **(e.2)**, $TG(2,0) - TG(1,0) \leq \frac{PC_1}{\tau u}\Big[v(1,0,0) - v(0,1,0)\Big] < 0.$

(ii) If $G(1,0) > 0$, by **(h.1)**, **(h.3)** & **(h.6)**,

$$TG(2,0) - TG(1,0) \leq \frac{1}{\tau}G(1,0) + \frac{PC_1}{\tau u}\Big[v(1,0,0) - v(0,1,0)\Big] \leq \frac{1}{\tau}r - \frac{1}{\tau}\frac{\tilde{u}(\alpha)}{u}r < 0.$$

Proof of Property (h.2).

$$\begin{aligned}
&Tv(i+1,0,k_2) - Tv(i,1,k_2) = \lambda[v(i+2,0,k_2) - v(i+1,1,k_2)] + \min\big\{G(i+1,k_2),0\big\}\\
&+\frac{1}{u}[v(i+1,0,k_2) - v(i,1,k_2)] + (r - r_1)\\
&Tv(i+2,0,k_2) - Tv(i+1,1,k_2) = \lambda[v(i+3,0,k_2) - v(i+2,1,k_2)] + \min\big\{G(i+2,k_2),0\big\}\\
&+\frac{1}{u}[v(i+2,0,k_2) - v(i+1,1,k_2)] + (r - r_1)
\end{aligned}$$

By **(h.1)** & **(h.2)**, it is obvious that $Tv(i+1,0,k_2)-Tv(i,1,k_2) \geq Tv(i+2,0,k_2)-Tv(i+1,1,k_2)$.

Proof of Property (h.3).

$$\begin{aligned}
Tv(i,1,k_2) &= \lambda v(i+1,1,k_2) + \frac{1}{\tau}v(i,0,k_2) + \frac{1}{u}v(i,1,k_2) + ir + r_1 + k_2 r_2,\\
Tv(i+1,0,k_2) &= \lambda v(i+2,0,k_2) - \big[(i+1)r + k_2 r_2\big] + \min\Big\{\frac{1}{u}\Big[PC_1 v(i,1,k_2)\\
&\qquad +PC_2 v(i,0,k_2+1)\Big] + \frac{1}{\tau}v(i+1,0,k_2), \frac{1}{\tau}v(i,0,k_2) + \frac{1}{u}v(i+1,0,k_2)\Big\}.
\end{aligned}$$

Hence,

$$\begin{aligned}
&\frac{PC_1}{u}\big[Tv(i,1,k_2) - Tv(i+1,0,k_2)\big]\\
=&\frac{PC_1}{u}\Big[\lambda[v(i+1,1,k_2) - v(i+2,0,k_2)] - \min\big\{G(i+1,k_2),0\big\} + \frac{1}{u}[v(i,1,k_2)\\
&-v(i+1,0,k_2)] + (r_1 - r)\Big]\\
\geq&\frac{PC_1}{u}\Big[\lambda[v(i+1,1,k_2) - v(i+2,0,k_2)] + \frac{1}{u}[v(i,1,k_2) - v(i+1,0,k_2)] + (r_1 - r)\Big]\\
\geq&(\lambda + 1/u)\frac{\tilde{u}(\alpha)}{u}r + \frac{PC_1(r_1 - r)}{u} = \frac{\tilde{u}(\alpha)}{u}r + \frac{PC_1(r_1-r)}{u} - (\alpha + 1/\tau)\frac{\tilde{u}(\alpha)}{u}r = \frac{\tilde{u}(\alpha)}{u}r.
\end{aligned}$$

Hence, Property **(h.3)** is preserved under operator T. $\Box$

Proof of Property (h.4).

We consider three cases. For $i = 1,\ k_2 = 1$, by **(h.4)** & **(h.5)**,

$$\begin{aligned}
&TG(1,1) - \frac{1}{\tau}\big[Tv(0,0,1) - Tv(0,0,0)\big]\\
=&\lambda G(2,1) + \frac{1}{u}\max\big\{G(1,1),0\big\} + \frac{1}{\tau}\min\big\{G(1,1),0\big\} + \frac{1}{\tau^2}\Big[v(0,0,1) - v(0,0,0)\Big] + \frac{r}{\tau}\\
&-\frac{1}{\tau}\Big[\lambda\big[v(1,0,1) - v(1,0,0)\big] + \frac{1}{u}\big[v(0,0,1) - v(0,0,0)\big] + r_2\Big]\\
\leq&\lambda\Big[G(2,1) - \frac{1}{\tau}\big[v(1,0,1) - v(1,0,0)\big]\Big] + \frac{1}{u}\Big[\max\big\{G(1,1),0\big\} - \frac{1}{\tau}\big[v(0,0,1) - v(0,0,0)\big]\Big]\\
&+\frac{r - r2}{\tau} + \frac{1}{\tau^2}\Big[v(0,0,1) - v(0,0,0)\Big]
\end{aligned}$$

$$
\begin{aligned}
\leq&(\lambda+1/u)\frac{\tilde{u}(\alpha)}{u}r+\frac{1}{u}\Big[\max\big\{G(1,1),0\big\}-\frac{1}{\tau}\big[v(0,0,1)-v(0,0,0)\big]-\frac{\tilde{u}(\alpha)}{u}r\Big]+\frac{PC_2(r-r_2)}{u}\\
=&\frac{\tilde{u}(\alpha)}{u}r+\frac{1}{u}\Big[\max\big\{G(1,1),0\big\}-\frac{1}{\tau}\big[v(0,0,1)-v(0,0,0)\big]-\frac{\tilde{u}(\alpha)}{u}r\Big]\leq\frac{\tilde{u}(\alpha)}{u}r.
\end{aligned}
$$

For $i=1,\ k_2\geq 2$, by **(h.4)** & **(h.5)**,

$$
\begin{aligned}
&TG(1,k_2)-\frac{1}{\tau}\big[Tv(0,0,k_2)-Tv(0,0,k_2-1)\big]\\
=&\Big[\lambda G(2,k_2)+\frac{1}{u}\max\big\{G(1,k_2),0\big\}+\frac{1}{\tau}\min\big\{G(1,k_2),0\big\}+\frac{1}{\tau^2}\big[v(0,0,k_2)-v(0,0,k_2-1)\big]\\
+&\frac{r}{\tau}\Big]-\frac{1}{\tau}\Big[\lambda\big[v(1,0,k_2)-v(1,0,k_2-1)\big]+\frac{1}{\tau}\big[v(0,0,k_2-1)-v(0,0,k_2-2)\big]\\
+&\frac{1}{u}\big[v(0,0,k_2)-v(0,0,k_2-1)\big]+r_2\Big]\\
\leq&\lambda\Big[G(2,k_2)-\frac{1}{\tau}\big[v(1,0,k_2)-v(1,0,k_2-1)\big]\Big]+\frac{1}{u}\Big[\max\big\{G(1,k_2),0\big\}-\frac{1}{\tau}\big[v(0,0,k_2)\\
-&v(0,0,k_2-1)\big]\Big]+\frac{1}{\tau^2}\Big[\big[v(0,0,k_2)-v(0,0,k_2-1)\big]-\big[v(0,0,k_2-1)-v(0,0,k_2-2)\big]\Big]\\
+&(r-r_2)/\tau\\
\leq&(\lambda+1/u)\frac{\tilde{u}(\alpha)}{u}r+\frac{1}{u}\Big[\max\big\{G(1,k_2),0\big\}-\frac{1}{\tau}\big[v(0,0,k_2)-v(0,0,k_2-1)\big]-\frac{\tilde{u}(\alpha)}{u}r\Big]\\
+&PC_2(r-r_2)/u\\
=&\frac{\tilde{u}(\alpha)}{u}r+\frac{1}{u}\Big[\max\big\{G(1,k_2),0\big\}-\frac{1}{\tau}\big[v(0,0,k_2)-v(0,0,k_2-1)\big]-\frac{\tilde{u}(\alpha)}{u}r\Big]\leq\frac{\tilde{u}(\alpha)}{u}r.
\end{aligned}
$$

For $i\geq 2,\ k_2\geq 1$,

$$
\begin{aligned}
&TG(i,k_2)-\frac{1}{\tau}\big[Tv(i-1,0,k_2)-Tv(i-1,0,k_2-1)\big]\\
=&\Big[\lambda G(i+1,k_2)+\frac{1}{u}\max\big\{G(i,k_2),0\big\}+\frac{PC_1}{\tau u}\big[v(i-1,0,k_2)-v(i-2,1,k_2)\big]+\frac{r}{\tau}\\
+&\frac{1}{\tau}\max\big\{G(i-1,k_2),0\big\}+\frac{1}{\tau}\min\big\{G(i,k_2),0\big\}+\frac{PC_2}{u}\min\big\{G(i-1,k_2+1),0\big\}\Big]\\
-&\frac{1}{\tau}\Big[\lambda\big[v(i,0,k_2)-v(i,0,k_2-1)\big]+\frac{1}{\tau}\big[v(i-2,0,k_2)-v(i-2,0,k_2-1)\big]+r_2\\
+&\frac{1}{u}\big[v(i-1,0,k_2)-v(i-1,0,k_2-1)\big]+\min\big\{G(i-1,k_2),0\big\}-\min\big\{G(i-1,k_2-1),0\big\}\Big]\\
=&\lambda\Big[G(i+1,k_2)-\frac{1}{\tau}\big[v(i,0,k_2)-v(i,0,k_2-1)\big]\Big]+\frac{1}{u}\Big[\max\big\{G(i,k_2),0\big\}-\frac{1}{\tau}\big[v(i-1,0,k_2)\\
-&v(i-1,0,k_2-1)\big]\Big]+\frac{1}{\tau}\Big[\max\big\{G(i-1,k_2),0\big\}-\frac{1}{\tau}\big[v(i-2,0,k_2)-v(i-2,0,k_2-1)\big]\Big]
\end{aligned}
$$

$$
\begin{aligned}
&-\frac{PC_1}{\tau u}\big[v(i-2,1,k_2)-v(i-1,0,k_2)\big]+\frac{1}{\tau}\big[\min\big\{G(i,k_2),0\big\}-\min\big\{G(i-1,k_2),0\big\}\big]\\
&+\frac{PC_2}{u}\min\big\{G(i-1,k_2+1),0\big\}+\frac{1}{\tau}\min\big\{G(i-1,k_2-1),0\big\}+\frac{r-r_2}{\tau}\\
&\le\lambda\Big[G(i+1,k_2)-\frac{1}{\tau}\big[v(i,0,k_2)-v(i,0,k_2-1)\big]\Big]+\frac{1}{u}\Big[\max\big\{G(i,k_2),0\big\}-\frac{1}{\tau}\big[v(i-1,0,k_2)\\
&-v(i-1,0,k_2-1)\big]\Big]+\frac{1}{\tau}\Big[\max\big\{G(i-1,k_2),0\big\}-\frac{1}{\tau}\big[v(i-2,0,k_2)-v(i-2,0,k_2-1)\big]\Big]\\
&-\frac{\tilde{u}(\alpha)}{\tau u}r+\frac{r-r_2}{\tau}.
\end{aligned}
$$

The last inequality holds because of **(h.1)** & **(h.3)**. By **(h.4)**,

$$
\begin{aligned}
&TG(i,k_2)-\frac{1}{\tau}\big[Tv(i-1,0,k_2)-Tv(i-1,0,k_2-1)\big]\\
&\le(\lambda+1/u)\frac{\tilde{u}(\alpha)}{u}r+\frac{1}{u}\Big[\max\big\{G(i,k_2),0\big\}-\frac{1}{\tau}\big[v(i-1,0,k_2)-v(i-1,0,k_2-1)\big]-\frac{\tilde{u}(\alpha)}{u}r\Big]\\
&+\frac{1}{\tau}\Big[\max\big\{G(i-1,k_2),0\big\}-\frac{1}{\tau}\big[v(i-2,0,k_2)-v(i-2,0,k_2-1)\big]-\frac{\tilde{u}(\alpha)}{u}r\Big]+\frac{r-r_2}{\tau}\\
&=\frac{\tilde{u}(\alpha)}{u}r+\frac{1}{u}\Big[\max\big\{G(i,k_2),0\big\}-\frac{1}{\tau}\big[v(i-1,0,k_2)-v(i-1,0,k_2-1)\big]-\frac{\tilde{u}(\alpha)}{u}r\Big]\\
&+\frac{1}{\tau}\Big[\max\big\{G(i-1,k_2),0\big\}-\frac{1}{\tau}\big[v(i-2,0,k_2)-v(i-2,0,k_2-1)\big]-\frac{\tilde{u}(\alpha)}{u}r\Big]\\
&+\frac{r-r_2}{\tau}-(\alpha+1/\tau)\frac{\tilde{u}(\alpha)}{u}r\\
&\le\frac{\tilde{u}(\alpha)}{u}r+\frac{r-r_2}{\tau}-(\alpha+1/\tau)\frac{\tilde{u}(\alpha)}{u}r=\frac{\tilde{u}(\alpha)}{u}r.
\end{aligned}
$$

Proof of Property (h.5).

For the cases that $k_1\ge 1,\ i\ge 0,\ k_2\ge 0$,

$$
\begin{aligned}
&\big[Tv(i,k_1,k_2+2)-Tv(i,k_1,k_2+1)\big]-\big[Tv(i,k_1,k_2+1)-Tv(i,k_1,k_2)\big]\\
&=\lambda\Big(\big[v(i+1,k_1,k_2+2)-v(i+1,k_1,k_2+1)\big]-\big[v(i+1,k_1,k_2+1)-v(i+1,k_1,k_2)\big]\Big)\\
&+\frac{1}{\tau}\Big(\big[v(i,k_1-1,k_2+2)-v(i,k_1-1,k_2+1)\big]-\big[v(i,k_1-1,k_2+1)-v(i,k_1-1,k_2)\big]\Big)\\
&+\frac{1}{u}\Big(\big[v(i,k_1,k_2+2)-v(i,k_1,k_2+1)\big]-\big[v(i,k_1,k_2+1)-v(i,k_1,k_2)\big]\Big)\\
&\le(1-\alpha)\left(\frac{PC_2}{u}-\frac{1}{\tau}\right)(r-r_2)\tau^2\le\left(\frac{PC_2}{u}-\frac{1}{\tau}\right)(r-r_2)\tau^2.
\end{aligned}
$$

Next, we assume that $k_1 = 0$. First we consider $i = 0,\ k_2 = 0$.

$$
\begin{aligned}
&\big[Tv(0,0,2) - Tv(0,0,1)\big] - \big[Tv(0,0,1) - Tv(0,0,0)\big] \\
=&\lambda\Big(\big[v(1,0,2) - v(1,0,1)\big] - \big[v(1,0,1) - v(1,0,0)\big]\Big) + \frac{1}{\tau}\big[v(0,0,1) - v(0,0,0)\big] \\
+&\frac{1}{u}\Big(\big[v(0,0,2) - v(0,0,1)\big] - \big[v(0,0,1) - v(0,0,0)\big]\Big) \\
\leq&(1-\alpha)\left(\frac{PC_2}{u} - \frac{1}{\tau}\right)(r - r_2)\tau^2 \leq \left(\frac{PC_2}{u} - \frac{1}{\tau}\right)(r - r_2)\tau^2.
\end{aligned}
$$

The first inequality holds because of **(h.5)** & **(h.7)**. For $i = 0,\ k_2 \geq 1$,

$$
\begin{aligned}
&\big[Tv(0,0,k_2+2) - Tv(0,0,k_2+1)\big] - \big[Tv(0,0,k_2+1) - Tv(0,0,k_2)\big] \\
=&\lambda\Big(\big[v(1,0,k_2+2) - v(1,0,k_2+1)\big] - \big[v(1,0,k_2+1) - v(1,0,k_2)\big]\Big) \\
+&\frac{1}{\tau}\Big(\big[v(0,0,k_2+1) - v(0,0,k_2)\big] - \big[v(0,0,k_2) - v(0,0,k_2-1)\big]\Big) \\
+&\frac{1}{u}\Big(\big[v(0,0,k_2+2) - v(0,0,k_2+1)\big] - \big[v(0,0,k_2+1) - v(0,0,k_2)\big]\Big) \\
\leq&(1-\alpha)\left(\frac{PC_2}{u} - \frac{1}{\tau}\right)(r - r_2)\tau^2 \leq \left(\frac{PC_2}{u} - \frac{1}{\tau}\right)(r - r_2)\tau^2.
\end{aligned}
$$

At last, for $i \geq 1,\ k_2 \geq 0$ we consider two separate cases: (i) $G(i, k_2+1) \geq 0$; (ii) $G(i, k_2+1) < 0$.

(i) If $G(i, k_2+1) \geq 0$,

$$
\begin{aligned}
&\big[Tv(i,0,k_2+2) - Tv(i,0,k_2+1)\big] - \big[Tv(i,0,k_2+1) - Tv(i,0,k_2)\big] \\
\leq&\lambda\Big(\big[v(i+1,0,k_2+2) - v(i+1,0,k_2+1)\big] - \big[v(i+1,0,k_2+1) - v(i+1,0,k_2)\big]\Big) \\
+&\frac{1}{\tau}\Big(\big[v(i-1,0,k_2+2) - v(i-1,0,k_2+1)\big] - \big[v(i-1,0,k_2+1) - v(i-1,0,k_2)\big]\Big) \\
+&\frac{1}{u}\Big(\big[v(i,0,k_2+2) - v(i,0,k_2+1)\big] - \big[v(i,0,k_2+1) - v(i,0,k_2)\big]\Big) \\
\leq&(1-\alpha)\left(\frac{PC_2}{u} - \frac{1}{\tau}\right)(r - r_2)\tau^2 \leq \left(\frac{PC_2}{u} - \frac{1}{\tau}\right)(r - r_2)\tau^2.
\end{aligned}
$$

(ii) If $G(i, k_2+1) < 0$,

$$
\begin{aligned}
&\big[Tv(i,0,k_2+2) - Tv(i,0,k_2+1)\big] - \big[Tv(i,0,k_2+1) - Tv(i,0,k_2)\big] \\
\leq&\lambda\Big(\big[v(i+1,0,k_2+2) - v(i+1,0,k_2+1)\big] - \big[v(i+1,0,k_2+1) - v(i+1,0,k_2)\big]\Big)
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{PC_1}{u}\Big(\big[v(i-1,1,k_2+2)-v(i-1,1,k_2+1)\big]-\big[v(i-1,1,k_2+1)-v(i-1,1,k_2)\big]\Big)\\
&+\frac{PC_2}{u}\Big(\big[v(i-1,0,k_2+3)-v(i-1,0,k_2+2)\big]-\big[v(i-1,0,k_2+2)-v(i-1,0,k_2+1)\big]\Big)\\
&+\frac{1}{\tau}\Big(\big[v(i,0,k_2+2)-v(i,0,k_2+1)\big]-\big[v(i,0,k_2+1)-v(i,0,k_2)\big]\Big)\\
&\leq(1-\alpha)\left(\frac{PC_2}{u}-\frac{1}{\tau}\right)(r-r_2)\tau^2\leq\left(\frac{PC_2}{u}-\frac{1}{\tau}\right)(r-r_2)\tau^2.
\end{aligned}
$$

Proof of Property (h.6).

$$
\begin{aligned}
&TG(i,0)\\
=&\lambda G(i+1,0)+\frac{1}{u}\max\big\{G(i,0),0\big\}+\frac{1}{\tau}\min\big\{G(i,0),0\big\}+\frac{PC_2}{u}\min\big\{G(i-1,1),0\big\}\\
&+\frac{1}{\tau}\max\big\{G(i-1,0),0\big\}+\frac{PC_1}{\tau u}\Big[v(i-1,0,0)-v(i-2,1,0)\Big]+\frac{r}{\tau}\\
\leq&\lambda G(i+1,0)+\frac{1}{u}\max\big\{G(i,0),0\big\}+\frac{1}{\tau}\max\big\{G(i-1,0),0\big\}+\frac{PC_1}{\tau u}\Big[v(i-1,0,0)\\
&-v(i-2,1,0)\Big]+\frac{r}{\tau}\\
\leq&(\lambda+\frac{1}{u}+\frac{1}{\tau})r+\frac{1}{\tau}\Big(r-\frac{PC_1}{u}\big[v(i-2,1,0)-v(i-1,0,0)\big]\Big)\\
\leq&(1-\alpha)r+\frac{1}{\tau}\Big(r-\frac{\tilde{u}(\alpha)}{u}r\Big)\leq(1-\alpha)r\leq r.
\end{aligned}
$$

The above is true for $i\geq 2$. When $i=1$,

$$
\begin{aligned}
TG(1,0)&=\lambda G(2,0)+\frac{1}{u}\max\big\{G(1,0),0\big\}+\frac{1}{\tau}\min\big\{G(1,0),0\big\}+\frac{r}{\tau}\\
&\leq\lambda G(2,0)+\frac{1}{u}\max\big\{G(1,0),0\big\}+\frac{r}{\tau}\leq(\lambda+\frac{1}{u}+\frac{1}{\tau})r=(1-\alpha)r\leq r.
\end{aligned}
$$

Proof of Property (h.7).

Inequality (**h.7**) establishes the upper bound for $v(i,k_1,k_2+1)-v(i,k_1,k_2)$. For $k_1\geq 1$,

$$
\begin{aligned}
&Tv(i,k_1,k_2+1)-Tv(i,k_1,k_2)\\
=&\lambda\big[v(i+1,k_1,k_2+1)-v(i+1,k_1,k_2)\big]+r_2+\frac{1}{\tau}\big[v(i,k_1-1,k_2+1)-v(i,k_1-1,k_2)\big]
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{1}{u}\big[v(i,k_1,k_2+1)-v(i,k_1,k_2)\big]\\
\leq&\lambda\frac{r_2}{\alpha}\Big[1-\beta_1^{k_1+k_2+1}\beta_2^{i+1}\Big]+\frac{1}{\tau}\frac{r_2}{\alpha}\Big[1-\beta_1^{k_1+k_2}\beta_2^{i}\Big]+\frac{1}{u}\frac{r_2}{\alpha}\Big[1-\beta_1^{k_1+k_2+1}\beta_2^{i}\Big]+r_2\\
=&\frac{r_2}{\alpha}\Big[1-\beta_1^{k_1+k_2+1}\beta_2^{i}\Big]+\lambda\frac{r_2}{\alpha}\beta_1^{k_1+k_2+1}\beta_2^{i}(1-\beta_2)-\frac{1}{\tau}\frac{r_2}{\alpha}\beta_1^{k_1+k_2}\beta_2^{i}(1-\beta_1)\\
&-\alpha\frac{r_2}{\alpha}\Big[1-\beta_1^{k_1+k_2+1}\beta_2^{i}\Big]+r_2\\
=&\frac{r_2}{\alpha}\Big[1-\beta_1^{k_1+k_2+1}\beta_2^{i}\Big]+\frac{r_2}{\alpha}\beta_1^{k_1+k_2}\beta_2^{i}\big[\lambda\beta_1(1-\beta_2)-\frac{1}{\tau}(1-\beta_1)+\alpha\beta_1\big]\\
\leq&\frac{r_2}{\alpha}\Big[1-\beta_1^{k_1+k_2+1}\beta_2^{i}\Big].
\end{aligned}
$$

The last inequality holds because of Lemma B.0.7. Next we consider the case that $k_1 = 0$. If $i = 0,\ k_2 = 0$, then

$$
\begin{aligned}
&Tv(0,0,1)-Tv(0,0,0)\\
=&\lambda\big[v(1,0,1)-v(1,0,0)\big]+\frac{1}{u}\big[v(0,0,1)-v(0,0,0)\big]+r_2\\
\leq&\lambda\frac{r_2}{\alpha}\big[1-\beta_1\beta_2\big]+\frac{1}{u}\frac{r_2}{\alpha}\big[1-\beta_1\big]+r_2=\frac{r_2}{\alpha}\big[1-\beta_1\big]+\frac{r_2}{\alpha}\big[\lambda\beta_1(1-\beta_2)-\frac{1}{\tau}(1-\beta_1)+\alpha\beta_1\big]\\
\leq&\frac{r_2}{\alpha}\big[1-\beta_1\big].
\end{aligned}
$$

The last inequality holds because of Lemma B.0.7. If $i = 0,\ k_2 \geq 1$ then

$$
\begin{aligned}
&Tv(0,0,k_2+1)-Tv(0,0,k_2)\\
=&\lambda\big[v(1,0,k_2+1)-v(1,0,k_2)\big]+\frac{1}{\tau}\big[v(0,0,k_2)-v(0,0,k_2-1)\big]\\
&+\frac{1}{u}\big[v(0,0,k_2+1)-v(0,0,k_2)\big]+r_2\\
\leq&\lambda\frac{r_2}{\alpha}\Big[1-\beta_1^{k_2+1}\beta_2\Big]+\frac{1}{\tau}\frac{r_2}{\alpha}\Big[1-\beta_1^{k_2}\Big]+\frac{1}{u}\frac{r_2}{\alpha}\Big[1-\beta_1^{k_2+1}\Big]+r_2\\
=&\frac{r_2}{\alpha}\Big[1-\beta_1^{k_2+1}\Big]+\frac{r_2}{\alpha}\beta_1^{k_2}\big[\lambda\beta_1(1-\beta_2)-\frac{1}{\tau}(1-\beta_1)+\alpha\beta_1\big]\\
\leq&\frac{r_2}{\alpha}\Big[1-\beta_1^{k_2+1}\Big].
\end{aligned}
$$

The last inequality holds because of Lemma B.0.7. If $i \geq 1,\ k_2 \geq 0$,

$$Tv(i,0,k_2+1)-Tv(i,0,k_2)$$

$$
\begin{aligned}
=&\lambda v(i+1,0,k_2+1)+ir+(k_2+1)r_2+\min\Big\{\frac{1}{u}\big[PC_1v(i-1,1,k_2+1)\\
+&PC_2v(i-1,0,k_2+2)\big]+\frac{1}{\tau}v(i,0,k_2+1),\frac{1}{\tau}v(i-1,0,k_2+1)+\frac{1}{u}v(i,0,k_2+1)\Big\}\\
-&\Big[\lambda v(i+1,0,k_2)+ir+k_2r_2+\min\Big\{\frac{1}{u}\big[PC_1v(i-1,1,k_2)+PC_2v(i-1,0,k_2+1)\big]\\
+&\frac{1}{\tau}v(i,0,k_2),\frac{1}{\tau}v(i-1,0,k_2)+\frac{1}{u}v(i,0,k_2)\Big\}\Big].
\end{aligned}
$$

We consider two separate cases: If $G(i,k_2)\geq 0$,

$$
\begin{aligned}
&Tv(i,0,k_2+1)-Tv(i,0,k_2)\\
\leq&\lambda\big[v(i+1,0,k_2+1)-v(i+1,0,k_2)\big]+r_2+\frac{1}{\tau}\big[v(i-1,0,k_2+1)-v(i-1,0,k_2)\big]\\
+&\frac{1}{u}\big[v(i,0,k_2+1)-v(i,0,k_2)\big]\\
\leq&\lambda\frac{r_2}{\alpha}\big[1-\beta_1^{k_2+1}\beta_2^{i+1}\big]+\frac{1}{\tau}\frac{r_2}{\alpha}\big[1-\beta_1^{k_2+1}\beta_2^{i-1}\big]+\frac{1}{u}\frac{r_2}{\alpha}\big[1-\beta_1^{k_2+1}\beta_2^{i}\big]+r_2\\
=&\frac{r_2}{\alpha}\big[1-\beta_1^{k_2+1}\beta_2^{i}\big]+\frac{r_2}{\alpha}\beta_1^{k_2+1}\beta_2^{i-1}\big[\lambda\beta_2(1-\beta_2)-\frac{1}{\tau}(1-\beta_2)+\alpha\beta_2\big]\\
\leq&\frac{r_2}{\alpha}\big[1-\beta_1^{k_2+1}\beta_2^{i}\big].
\end{aligned}
$$

The last inequality holds because of Lemma B.0.7. If $G(i,k_2)<0$,

$$
\begin{aligned}
&Tv(i,0,k_2+1)-Tv(i,0,k_2)\\
\leq&\lambda\big[v(i+1,0,k_2+1)-v(i+1,0,k_2)\big]+\frac{PC_1}{u}\big[v(i-1,1,k_2+1)-v(i-1,1,k_2)\big]\\
+&\frac{PC_2}{u}\big[v(i-1,0,k_2+2)-v(i-1,0,k_2+1)\big]+\frac{1}{\tau}\big[v(i,0,k_2+1)-v(i,0,k_2)\big]+r_2\\
\leq&\lambda\frac{r_2}{\alpha}\big[1-\beta_1^{k_2+1}\beta_2^{i+1}\big]+\frac{PC_1}{u}\frac{r_2}{\alpha}\big[1-\beta_1^{k_2+2}\beta_2^{i-1}\big]+\frac{PC_2}{u}\frac{r_2}{\alpha}\big[1-\beta_1^{k_2+2}\beta_2^{i-1}\big]\\
+&\frac{1}{\tau}\frac{r_2}{\alpha}\big[1-\beta_1^{k_2+1}\beta_2^{i}\big]+r_2\\
=&\frac{r_2}{\alpha}\big[1-\beta_1^{k_2+1}\beta_2^{i}\big]+\frac{r_2}{\alpha}\beta_1^{k_2+1}\beta_2^{i-1}\big[\lambda\beta_2(1-\beta_2)-\frac{1}{u}(\beta_1-\beta_2)+\alpha\beta_2\big]\\
\leq&\frac{r_2}{\alpha}\big[1-\beta_1^{k_2+1}\beta_2^{i}\big].
\end{aligned}
$$

The last inequality holds because of Lemma B.0.7. $\square$

Proof of Lemma 4.3.4.

We prove this lemma by verifying the three conditions in Theorem 11.5, Porteus (2002).

Proof of part (i). Let a structured policy be any policy that serves a class-1 job if $k_1 \geq 1$ and serves a class-2 job only if $i = k_1 = 0$; otherwise, serve without triage. Let $\mathcal{U} = \mathcal{E} \cap \mathcal{F}$ be endowed with the L^∞ metric ρ. The metric space $(\rho, \mathcal{U})$ is complete because for any given $\{f_n\}_0^\infty$ that $f_n \in \mathcal{U}$ $(n \geq 0)$ and $\lim_{n\to\infty} f_n = f$, i.e., f_n satisfies **(e.1)**$\sim$**(e.5)** and **(f.1)**$\sim$**(f.2)**, it is obvious that f satisfies them as well. Hence, Theorem 11.5 (a) holds.

Consider a structured policy π such that it chooses to serve class-1 if $k_1 \geq 1$ and serves class-2 only if $i = k_1 = 0$; otherwise, serve without triage. From the optimality equations, it is obvious that policy π is optimal for this one stage minimization problem. Hence, Theorem 11.5 (b) holds.

The *preservation* condition, i.e., Theorem 11.5 (c), holds because of Lemma 4.3.1. Hence, based on Theorem 11.5, the optimal value functions are structured and there exists an optimal structured stationary policy.

Proof of part (ii). Let a structured policy be any policy that serves a class-1 job if $k_1 \geq 1$ and serves a class-2 job only if $i = k_1 = 0$; if the server performs triage on an unknown job in $(i, 0, k_2)$, then performs triage in $(i, 0, k_2')$ as well for $0 \leq k_2' \leq k_2$. Let $\mathcal{U} = \mathcal{E} \cap \mathcal{G}$ be endowed with the L^∞ metric ρ. The metric space $(\rho, \mathcal{U})$ is complete because for any given $\{f_n\}_0^\infty$ that $f_n \in \mathcal{U}$ $(n \geq 0)$ and $\lim_{n\to\infty} f_n = f$, i.e., f_n satisfies **(e.1)**$\sim$**(e.5)** and **(g.1)**$\sim$**(g.4)**, it is obvious that f satisfies them as well. Hence, Theorem 11.5 (a) holds.

Consider a structured policy π such that it chooses to serve a class-1 job if $k_1 \geq 1$ and serves a class-2 job only if $i = k_1 = 0$; if the server performs triage on an unknown job in $(i, 0, k_2)$, then performs triage in $(i, 0, k_2')$ as well for $0 \leq k_2' \leq k_2$; otherwise, follow the optimal policy. From the optimality equations, it is obvious that policy π is optimal for this one stage minimization problem. Hence, Theorem 11.5 (b) holds.

The *preservation* condition, i.e., Theorem 11.5 (c), holds because of Lemma 4.3.2. Hence, based on Theorem 11.5, the optimal value functions are structured and there exists an optimal structured stationary policy.

Proof of part (iii). Let a structured policy be any policy that serves a class-1 job if $k_1 \geq 1$ and serves a class-2 job only if $i = k_1 = 0$; if the server performs triage on an unknown job in $(i, 0, k_2)$, then performs triage in $(i', 0, k_2)$ as well for $i' \geq i$. Let $\mathcal{U} = \mathcal{E} \cap \mathcal{H}$ be endowed with the L^∞ metric ρ. The metric space $(\rho, \mathcal{U})$ is complete because for any given $\{f_n\}_0^\infty$ that $f_n \in \mathcal{U}$ $(n \geq 0)$ and $\lim_{n\to\infty} f_n = f$, i.e., f_n satisfies **(e.1)**$\sim$**(e.5)** and **(h.1)**$\sim$**(h.7)**, it is obvious that f satisfies them as well. Hence, Theorem 11.5 (a) holds.

Consider a structured policy π such that it chooses to serve a class-1 job if $k_1 \geq 1$ and serves a class-2 job only if $i = k_1 = 0$; if the server performs triage on an unknown job in $(i, 0, k_2)$, then performs triage in $(i', 0, k_2)$ as well for $i' \geq i$; otherwise, follow the optimal policy. From the optimality equations, it is obvious that policy π is optimal for this one stage minimization problem. Hence, Theorem 11.5 (b) holds.

The *preservation* condition, i.e., Theorem 11.5 (c), holds because of Lemma 4.3.3. Hence, based on Theorem 11.5, the optimal value functions are structured and there exists an optimal structured stationary policy. $\square$

Proof of Lemma 4.3.5.

We verify one by one that the three SEN conditions hold for our discounted cost model. Let $z = (0, 0, 0)$ be the initial system state and policy π be Policy No-Triage, i.e., no jobs will be triaged; the server serves each job in a first-come-first-serve manner. Hence, this is an $M/M/1$ queue starting at the origin. Denote $V^\pi_{\alpha_1}(z)$ as the total discounted cost under policy π.

$$V^\pi_{\alpha_1}(z) \;=\; \sum_{n=0}^{\infty} {\alpha_1}^n \sum_{k=0}^{\infty} kr \cdot p_k(n) = r \sum_{n=0}^{\infty} {\alpha_1}^n \sum_{k=0}^{\infty} k \cdot p_k(n) = r \sum_{n=0}^{\infty} {\alpha_1}^n Q(n),$$

where $p_k(n)$ is the probability that the queue length is k at time n and $Q(n)$ is the expected queue length at time n given the queue is empty at $n = 0$. From Abate and Whitt (1987),

$$Q(n) \leq Q(n+1) \leq \cdots \leq Q(\infty) = \frac{\rho}{1-\rho}, \quad \text{where } \rho = \lambda\tau.$$

Hence,

$$V^\pi_{\alpha_1}(z) \leq r \sum_{n=0}^{\infty} {\alpha_1}^n \frac{\rho}{1-\rho} = \frac{r\rho}{1-\rho} \cdot \frac{1}{1-\alpha_1},$$

and

$$(1-\alpha_1)V_{\alpha_1}(z) \le (1-\alpha_1)V^{\pi}_{\alpha_1}(z) \le \frac{r\rho}{1-\rho} < \infty, \text{ for } \alpha_1 \in (0,1).$$

Therefore, SEN1 holds. Let (i, k_1, k_2) be any state in $\mathcal{S}$. We need to find an upper bound on

$$h_{\alpha_1}(i,k_1,k_2) =: V_{\alpha_1}(i,k_1,k_2) - V_{\alpha_1}(z).$$

Define policy π as follows: *starting from (i, k_1, k_2), it first follows Policy No-Triage until the state reaches $z = (0, 0, 0)$. After that, it follows the optimal policy.* Assume the time it takes to reach z is T_z which is a random variable. Denote the discounted cost incurred from time 0 to T_z by $M_{\alpha_1}(i, k_1, k_2)$.

$$\begin{aligned} V_{\alpha_1}(i,k_1,k_2) - V_{\alpha_1}(z) &\le V^{\pi}_{\alpha_1}(i,k_1,k_2) - V_{\alpha_1}(z) \\ &= M_{\alpha_1}(i,k_1,k_2) + {\alpha_1}^{T_z} V_{\alpha_1}(z) - V_{\alpha_1}(z) < M_{\alpha_1}(i,k_1,k_2). \end{aligned}$$

By Kulkarni (2009), $E(T) = (i + k_1 + k_2)/(\mu - \lambda)$. Hence,

$$\begin{aligned} M_{\alpha_1}(i,k_1,k_2) <& E\left\{\sum_{n=0}^{T} {\alpha_1}^n \sum_{k=0}^{\infty} (ir + k_1 r_1 + k_2 r_2 + kr) p_k(n)\right\} \\ \le& E\left\{\sum_{n=0}^{T} \sum_{k=0}^{\infty} (ir + k_1 r_1 + k_2 r_2 + kr) p_k(n)\right\} \\ =& E\left\{\sum_{n=0}^{T} (ir + k_1 r_1 + k_2 r_2)\right\} + E\left\{\sum_{n=0}^{T} \sum_{k=0}^{\infty} kr p_k(n)\right\} \\ =& (ir + k_1 r_1 + k_2 r_2) E(T+1) + E\left\{r \sum_{n=0}^{T} Q(n)\right\} \\ \le& (ir + k_1 r_1 + k_2 r_2) E(T+1) + E\left\{r \sum_{n=0}^{T} \frac{\rho}{1-\rho}\right\} \\ =& \left(ir + k_1 r_1 + k_2 r_2 + \frac{r\rho}{1-\rho}\right) E(T+1) \\ =& \left(ir + k_1 r_1 + k_2 r_2 + \frac{r\rho}{1-\rho}\right) \left(\frac{i + k_1 + k_2}{\mu - \lambda} + 1\right), \end{aligned}$$

where $p_k(n)$ and $Q(n)$ are defined before. Hence, for $(i, k_1, k_2) \in \mathcal{S}, \alpha_1 \in (0, 1)$,

$$h^*_{\alpha_1}(i, k_1, k_2) < \left[ir + k_1 r_1 + k_2 r_2 + \frac{r\rho}{1-\rho}\right]\left[\frac{i + k_1 + k_2}{\mu - \lambda} + 1\right].$$

This completes the verification of SEN2. It is straightforward to see SEN3 holds since $h_{\alpha_1}(s) \geq 0$ for any $s \in \mathcal{S}$. $\square$

APPENDIX C: PROOF OF RESULTS IN CHAPTER 5

Proof of Theorem 5.1.1.

We show the following lemma, which will help prove Theorem 5.1.1. The decision vector (s_1, s_2, s_3, s_4) denotes that there are s_1 servers assigned to do Tr, s_2 servers to do SU, s_3 servers to do SC1, s_4 servers to do SC2. Because servers work in a non-cooperative manner, the number of servers working on a certain class should not exceed the number of jobs in that class and $\sum_{i=1}^{4} s_i \leq M$. Starting from state (i, k_1, k_2), the total expected cost under decision (s_1, s_2, s_3, s_4) then follow the optimal policy is denoted by $W(s_1, s_2, s_3, s_4)$.

Lemma C.0.8. *Assume* $i \geq s_1 + s_2 + 1, k_1 \geq s_1 + 2, k_2 \geq s_4 + 1$.

(i) If $W(s_1, s_2, s_3+1, s_4) \leq W(s_1+1, s_2, s_3, s_4)$*, then* $W(s_1-1, s_2, s_3+2, s_4) \leq W(s_1, s_2, s_3+1, s_4),\ s_1 \geq 1$.

(ii) If $W(s_1, s_2, s_3+1, s_4) \leq W(s_1, s_2+1, s_3, s_4)$*, then* $W(s_1, s_2-1, s_3+2, s_4) \leq W(s_1, s_2, s_3+1, s_4),\ s_2 \geq 1$.

(iii) If $W(s_1, s_2, s_3+1, s_4) \leq W(s_1, s_2, s_3, s_4+1)$*, then* $W(s_1, s_2, s_3+2, s_4-1) \leq W(s_1, s_2, s_3+1, s_4),\ s_4 \geq 1$.

Proof. Define $R(s_1, s_2, s_3, s_4) = s_1/u + (s_2 + s_3 + s_4)/\tau$.

(i) After uniformization with the factor $R(s_1, s_2, s_3 + 1, s_4) + R(s_1 + 1, s_2, s_3, s_4)$, the cost function $W(\cdot)$ can be written as

$$
\begin{aligned}
& W(s_1, s_2, s_3+1, s_4) \\
= \quad & \frac{R(s_1, s_2, s_3+1, s_4)}{R(s_1, s_2, s_3+1, s_4) + R(s_1+1, s_2, s_3, s_4)} \Bigg\{ \\
& \frac{s_1/u}{R(s_1, s_2, s_3+1, s_4)} \big[PC_1 V(i-1, k_1+1, k_2) + PC_2 V(i-1, k_1, k_2+1) \big] \\
+ \quad & \frac{s_2/\tau}{R(s_1, s_2, s_3+1, s_4)} V(i-1, k_1, k_2) + \frac{(s_3+1)/\tau}{R(s_1, s_2, s_3+1, s_4)} V(i, k_1-1, k_2) \\
+ \quad & \frac{s_4/\tau}{R(s_1, s_2, s_3+1, s_4)} V(i, k_1, k_2-1) \Bigg\} + \frac{ir + k_1 r_1 + k_2 r_2}{R(s_1, s_2, s_3+1, s_4) + R(s_1+1, s_2, s_3, s_4)}
\end{aligned}
$$

$$
+ \quad \frac{R(s_1+1,s_2,s_3,s_4)}{R(s_1,s_2,s_3+1,s_4)+R(s_1+1,s_2,s_3,s_4)} V(i,k_1,k_2),
$$

$$
\begin{aligned}
& W(s_1+1,s_2,s_3,s_4) \\
= \quad & \frac{R(s_1+1,s_2,s_3,s_4)}{R(s_1,s_2,s_3+1,s_4)+R(s_1+1,s_2,s_3,s_4)} \Bigg\{ \\
& \frac{(s_1+1)/u}{R(s_1+1,s_2,s_3,s_4)} \big[PC_1 V(i-1,k_1+1,k_2) + PC_2 V(i-1,k_1,k_2+1)\big] \\
+ \quad & \frac{s_2/\tau}{R(s_1+1,s_2,s_3,s_4)} V(i-1,k_1,k_2) + \frac{s_3/\tau}{R(s_1+1,s_2,s_3,s_4)} V(i,k_1-1,k_2) \\
+ \quad & \frac{s_4/\tau}{R(s_1+1,s_2,s_3,s_4)} V(i,k_1,k_2-1) \Bigg\} + \frac{ir+k_1r_1+k_2r_2}{R(s_1,s_2,s_3+1,s_4)+R(s_1+1,s_2,s_3,s_4)} \\
+ \quad & \frac{R(s_1,s_2,s_3+1,s_4)}{R(s_1,s_2,s_3+1,s_4)+R(s_1+1,s_2,s_3,s_4)} V(i,k_1,k_2),
\end{aligned}
$$

Take the difference of $W(s_1,s_2,s_3+1,s_4)$ and $W(s_1+1,s_2,s_3,s_4)$,

$$
\begin{aligned}
& W(s_1,s_2,s_3+1,s_4) - W(s_1+1,s_2,s_3,s_4) \\
= \quad & \frac{-[PC_1V(i-1,k_1+1,k_2) + PC_2V(i-1,k_1,k_2+1)]/u + V(i,k_1-1,k_2)/\tau}{R(s_1,s_2,s_3+1,s_4)+R(s_1+1,s_2,s_3,s_4)} \\
& -\frac{R(s_1,s_2,s_3+1,s_4)-R(s_1+1,s_2,s_3,s_4)}{R(s_1,s_2,s_3+1,s_4)+R(s_1+1,s_2,s_3,s_4)} V(i,k_1,k_2) \\
= \quad & \frac{1}{R(s_1,s_2,s_3+1,s_4)+R(s_1+1,s_2,s_3,s_4)} \Bigg\{ \big[V(i,k_1,k_2) - PC_1V(i-1,k_1+1,k_2) \\
& -PC_2V(i-1,k_1,k_2+1)\big]/u - \big[V(i,k_1,k_2) - V(i,k_1-1,k_2)\big]/\tau \Bigg\} \leq 0.
\end{aligned}
$$

Similarly, take the difference of $W(s_1-1,s_2,s_3+2,s_4)$ and $W(s_1,s_2,s_3+1,s_4)$

$$
\begin{aligned}
& W(s_1-1,s_2,s_3+2,s_4) - W(s_1,s_2,s_3+1,s_4) \\
= \quad & \frac{R(s_1,s_2,s_3+1,s_4)+R(s_1+1,s_2,s_3,s_4)}{R(s_1-1,s_2,s_3+2,s_4)+R(s_1,s_2,s_3+1,s_4)} \big(W(s_1,s_2,s_3+1,s_4) - W(s_1+1,s_2,s_3,s_4)\big) \\
\leq \quad & 0.
\end{aligned}
$$

(ii) After uniformization with the factor $R(s_1,s_2,s_3+1,s_4)$, the cost function $W(\cdot)$ can be written as

$$
W(s_1,s_2,s_3+1,s_4)
$$

$$
\begin{aligned}
&= \frac{s_1/u}{R(s_1,s_2,s_3+1,s_4)}\big[PC_1V(i-1,k_1+1,k_2)+PC_2V(i-1,k_1,k_2+1)\big] \\
&+ \frac{s_2/\tau}{R(s_1,s_2,s_3+1,s_4)}V(i-1,k_1,k_2)+\frac{(s_3+1)/\tau}{R(s_1,s_2,s_3+1,s_4)}V(i,k_1-1,k_2) \\
&+ \frac{s_4/\tau}{R(s_1,s_2,s_3+1,s_4)}V(i,k_1,k_2-1)+\frac{ir+k_1r_1+k_2r_2}{R(s_1,s_2,s_3+1,s_4)},
\end{aligned}
$$

$$
\begin{aligned}
& W(s_1,s_2+1,s_3,s_4) \\
&= \frac{s_1/u}{R(s_1,s_2+1,s_3,s_4)}\big[PC_1V(i-1,k_1+1,k_2)+PC_2V(i-1,k_1,k_2+1)\big] \\
&+ \frac{(s_2+1)/\tau}{R(s_1,s_2+1,s_3,s_4)}V(i-1,k_1,k_2)+\frac{s_3/\tau}{R(s_1,s_2+1,s_3,s_4)}V(i,k_1-1,k_2) \\
&+ \frac{s_4/\tau}{R(s_1,s_2+1,s_3,s_4)}V(i,k_1,k_2-1)\frac{ir+k_1r_1+k_2r_2}{R(s_1,s_2,s_3+1,s_4)}.
\end{aligned}
$$

Because $R(s_1,s_2,s_3+1,s_4)=R(s_1,s_2+1,s_3,s_4)$,

$$W(s_1,s_2,s_3+1,s_4)-W(s_1,s_2+1,s_3,s_4)=-\frac{\left[V(i-1,k_1,k_2)-V(i,k_1-1,k_2)\right]/\tau}{R(s_1,s_2,s_3+1,s_4)}\leq 0,$$

and

$$W(s_1,s_2-1,s_3+2,s_4)-W(s_1,s_2,s_3+1,s_4)=W(s_1,s_2,s_3+1,s_4)-W(s_1,s_2+1,s_3,s_4)\leq 0.$$

(iii) The proof is similar to that of (ii).

$$W(s_1,s_2,s_3+1,s_4)-W(s_1,s_2,s_3,s_4+1)=-\frac{\left[V(i,k_1,k_2-1)-V(i,k_1-1,k_2)\right]/\tau}{R(s_1,s_2,s_3+1,s_4)}\leq 0,$$

hence, $W(s_1,s_2,s_3+2,s_4-1)-W(s_1,s_2,s_3+1,s_4)=W(s_1,s_2,s_3+1,s_4)-W(s_1,s_2,s_3,s_4+1)\leq 0$. □

Proof of Theorem 5.1.1: Lemma C.0.8 implies that if it is better, in the sense of lower cost, to assign an available server to serve a class-1 job other than to serve/triage jobs from class-j, then it is better to move another server of serving/triaging class-j to serve class-1. It is better to continue doing this until either all class-1 jobs are being served or all servers are dedicated to serving class-1 jobs. This result can be extended to other actions (proofs are similar thus omitted) and implies that servers should work on the same type of actions, if possible.

Lemma C.0.9. *For any system state* (i, k_1, k_2)*,*

(i) $V(i, k_1 + 1, k_2) \geq V(i, k_1, k_2 + 1)$.

(ii) $V(i, k_1 + 1, k_2) \geq V(i + 1, k_1, k_2)$.

(iii) $V(i + 1, k_1, k_2) \geq V(i, k_1, k_2 + 1)$.

Proof. (i) Consider two systems. System 1 starts in state $(i, k_1 + 1, k_2)$ and uses the optimal policy. System 2 starts in state $(i, k_1, k_2 + 1)$ and uses policy π, which takes whatever action System 1 takes until System 1 starts serving the extra class-1 job. While System 1 serves the extra class-1 job, System 2 serves the extra class-2 job and then follows the same actions as System 1 from now on.

Let the total expected cost under policy π be denoted by $V_\pi(i, k_1, k_2)$. The difference between $V(i, k_1+1, k_2)$ and $V_\pi(i, k_1, k_2+1)$ is at least as large as the expected cost incurred during the service of the additional class-1 job. Hence,

$$
\begin{aligned}
& V(i, k_1 + 1, k_2) - V(i, k_1, k_2 + 1) \\
= \; & V(i, k_1 + 1, k_2) - V_\pi(i, k_1, k_2 + 1) + V_\pi(i, k_1, k_2 + 1) - V(i, k_1, k_2 + 1) \\
\geq \; & V(i, k_1 + 1, k_2) - V_\pi(i, k_1, k_2 + 1) \\
\geq \; & (r_1 - r_2)\tau = \frac{\tau}{PC_1 PC_2} pq(v_1 + v_2 - 1)(h_1 - h_2) \geq 0.
\end{aligned}
$$

(ii) Follow the idea in the proof of (i), we get $V(i, k_1 + 1, k_2) - V(i + 1, k_1, k_2) \geq (r_1 - r)\tau = \frac{\tau}{PC_1} pq(v_1 + v_2 - 1)(h_1 - h_2) \geq 0$.

(iii) Follow the idea in the proof of (i), we get $V(i + 1, k_1, k_2) - V(i, k_1, k_2 + 1) \geq (r - r_2)\tau = \frac{\tau}{PC_2} pq(v_1 + v_2 - 1)(h_1 - h_2) \geq 0$. $\square$

Proof of Theorem 5.2.1.

We first show the following lemmas.

Lemma C.0.10. *The optimal value function* v^* *satisfies*

(a.1) $v^*(i, k_1 - 1, k_2) \leq v^*(i - 1, k_1, k_2),\ i \geq 1,\ k_1 \geq 1.$

(a.2) $v^*(i, k_1 - 1, k_2) \leq v^*(i, k_1, k_2 - 1),\ k_1 \geq 1,\ k_2 \geq 1.$

(a.3) $v^*(i - 1, k_1, k_2) \leq v^*(i, k_1, k_2 - 1),\ i \geq 1,\ k_2 \geq 1.$

Proof. The proof of **(a.1)** uses a coupling argument. Consider two systems. System 1 starts in state $(i - 1, k_1, k_2)$ and uses the optimal policy. System 2 starts in state $(i, k_1 - 1, k_2)$ and uses policy π, which takes whatever action System 1 takes until System 1 starts serving the extra class-1 job. While System 1 serves the extra class-1 job, System 2 serves the extra untriaged job directly without triage and then follows the same actions as System 1 from now on.

Let the total expected discounted cost under policy π be denoted by $v_\pi(i, k_1, k_2)$. The difference between $v^*(i, k_1 - 1, k_2)$ and $v^*(i - 1, k_1, k_2)$ is at least as large as the expected cost incurred during the service of the additional class-1 job. Hence,

$$
\begin{aligned}
&v^*(i, k_1 - 1, k_2) - v^*(i - 1, k_1, k_2) \\
=&v^*(i, k_1 - 1, k_2) - v_\pi(i, k_1 - 1, k_2) + v_\pi(i, k_1 - 1, k_2) - v^*(i - 1, k_1, k_2) \\
\leq&v_\pi(i, k_1 - 1, k_2) - v^*(i - 1, k_1, k_2) < (r - r_1)\tau < 0.
\end{aligned}
$$

The proofs of **(a.2)** and **(a.3)** are similar to that for **(a.1)** thus omitted. $\square$

Let E be the set of functions defined on Z^3 such that if $v \in E$, then

$$
\begin{aligned}
&\lambda v(i + 1, k_1, k_2) + \frac{1}{\tau} v(i, k_1 - 1, k_2) + ir + k_1 r_1 + k_2 r_2 \leq \\
&\qquad PC_1 v(i - 1, k_1 + 1, k_2) + PC_2 v(i - 1, k_1, k_2 + 1) + C,\ i \geq 1,\ k_1 \geq 1.
\end{aligned}
\tag{C.1}
$$

Lemma C.0.11. *If $v \in E$, then $Tv \in E$. Hence, the optimal value function $v^* \in E$.*

Proof. (i) If $k_1 \geq 2$,

$$
\begin{aligned}
Tv(i + 1, k_1, k_2) &= \lambda v(i + 2, k_1, k_2) + \frac{1}{\tau} v(i + 1, k_1 - 1, k_2) + (i + 1)r + k_1 r_1 + k_2 r_2, \\
Tv(i, k_1 - 1, k_2) &= \lambda v(i + 1, k_1 - 1, k_2) + \frac{1}{\tau} v(i, k_1 - 2, k_2) + ir + (k_1 - 1)r_1 + k_2 r_2, \\
Tv(i - 1, k_1 + 1, k_2) &= \lambda v(i, k_1 + 1, k_2) + \frac{1}{\tau} v(i - 1, k_1, k_2) + (i - 1)r + (k_1 + 1)r_1 + k_2 r_2,
\end{aligned}
$$

$$
\begin{aligned}
Tv(i-1,k_1,k_2+1) &= \lambda v(i,k_1,k_2+1) + \frac{1}{\tau}v(i-1,k_1-1,k_2+1)\\
&+ (i-1)r + k_1r_1 + (k_2+1)r_2.
\end{aligned}
$$

Hence,

$$
\begin{aligned}
&PC_1Tv(i-1,k_1+1,k_2) + PC_2Tv(i-1,k_1,k_2+1) + C - \lambda Tv(i+1,k_1,k_2)\\
&- \frac{1}{\tau}Tv(i,k_1-1,k_2) - \big[ir + k_1r_1 + k_2r_2\big]\\
=&\lambda\Big[PC_1v(i,k_1+1,k_2) + PC_2v(i,k_1,k_2+1) + C - \lambda v(i+2,k_1,k_2) - \frac{1}{\tau}v(i+1,k_1-1,k_2)\\
&- \big[(i+1)r + k_1r_1 + k_2r_2\big]\Big] + \frac{1}{\tau}\Big[PC_1v(i-1,k_1,k_2) + PC_2v(i-1,k_1-1,k_2+1) + C\\
&- \lambda v(i+1,k_1-1,k_2) - \frac{1}{\tau}v(i,k_1-2,k_2) - \big[ir + (k_1-1)r_1 + k_2r_2\big]\Big] + \alpha C > 0.
\end{aligned}
$$

(ii) If $k_1 = 1$,

$$
\begin{aligned}
Tv(i+1,1,k_2) &= \lambda v(i+2,1,k_2) + \frac{1}{\tau}v(i+1,0,k_2) + (i+1)r + r_1 + k_2r_2,\\
Tv(i,0,k_2) &= \min\Big\{PC_1v(i-1,1,k_2) + PC_2v(i-1,0,k_2+1) + C,\\
&\qquad \lambda v(i+1,0,k_2) + \frac{1}{\tau}v(i-1,0,k_2) + ir + k_2r_2\Big\},\\
Tv(i-1,2,k_2) &= \lambda v(i,2,k_2) + \frac{1}{\tau}v(i-1,1,k_2) + (i-1)r + 2r_1 + k_2r_2,\\
Tv(i-1,1,k_2+1) &= \lambda v(i,1,k_2+1) + \frac{1}{\tau}v(i-1,0,k_2+1) + (i-1)r + r_1 + (k_2+1)r_2.
\end{aligned}
$$

Hence,

$$
\begin{aligned}
&PC_1Tv(i-1,2,k_2) + PC_2Tv(i-1,1,k_2+1) + C - \lambda Tv(i+1,1,k_2)\\
&- \frac{1}{\tau}Tv(i,0,k_2) - \big[ir + r_1 + k_2r_2\big]\\
\geq&\lambda\Big[PC_1v(i,2,k_2) + PC_2v(i,1,k_2+1) + C - \lambda v(i+2,1,k_2) - \frac{1}{\tau}v(i+1,0,k_2)\\
&- \big[(i+1)r + r_1 + k_2r_2\big]\Big] + \frac{1}{\tau}\Big[PC_1v(i-1,1,k_2) + PC_2v(i-1,0,k_2+1) + C\\
&- \lambda v(i+1,0,k_2) - \frac{1}{\tau}v(i-1,0,k_2) - (ir + k_2r_2)\Big] - \frac{1}{\tau}\max\Big\{PC_1v(i-1,1,k_2)\\
&+ PC_2v(i-1,0,k_2+1) + C - \lambda v(i+1,0,k_2) - \frac{1}{\tau}v(i-1,0,k_2) - (ir + k_2r_2), 0\Big\}
\end{aligned}
$$

$$+ \alpha C > 0.$$

$\square$

Proof of Theorem 5.2.1: By Lemma C.0.10 and Lemma C.0.11, the optimal value functions satisfy (**a.1**)$\sim$(**a.3**) and (C.1), it is obvious to see that class-1 jobs should be prioritized over all other types of jobs; we should serve a class-2 job only when there are no other types of jobs. $\square$

REFERENCES

Abate, J. and Whitt, W. (1987), "Transient behavior of the M/M/l queue: Starting at the origin," *Queueing Systems*, 2, 41–65.

Alagoz, O. and Ayvaci, M. U. (2010), "Uniformization in Markov Decision Processes," *Wiley Encyclopedia of Operations Research and Management Science*.

Alizamir, S., de Véricourt, F., and Sun, P. (2012), "Diagnostic Accuracy Under Congestion," *Management Science*, 59, 157–171.

Argon, N. T. and Ziya, S. (2009), "Priority Assignment Under Imperfect Information on Customer Type Identities," *Manufacturing & Service Operations Management*, 11, 674–693.

Bertsekas, D. P. (2007), "Dynamic programming and optimal control, 2," *Athena Scientific optimization and computation series*.

Budhiraja, A., Ghosh, A., and Liu, X. (2012), "Dynamic scheduling for Markov modulated single-server multiclass queueing systems in heavy traffic," *Submitted*.

Cox, D. and Smith, W. (1961), *Queues*, London : Methuen & Co.

D'Haen, J. and den Poel, D. V. (2013), "Model-supported business-to-business prospect prediction based on an iterative customer acquisition framework," *Industrial Marketing Management*, 42, 544–551.

Dobson, G. and Sainathan, A. (2011), "On the impact of analyzing customer information and prioritizing in a service system." *Decision Support Systems*, 51, 875–883.

Dobson, G., Tezcan, T., and Tilson, V. (2013), "Optimal Workflow Decisions for Investigators in Systems with Interruptions," *Management Science*, Online edition January 8.

Hafen, B. Q., Mistovich, J. J., and Karren, K. J. (1999), *Prehospital Emergency Care*, Prentice Hall, sixth ed.

Harrison, J. (1975), "Dynamic scheduling of a multiclass queue: Discount optimality," *Operations Research*, 23, 270–282.

Hirayama, T., Kijima, M., and Nishimura, S. (1989), "Further results for dynamic scheduling of multiclass G/G/1 queues," *Journal of Applied Probability*, 595–603.

Kakalik, J. and Little, J. (1971), "Optimal Service Policy for the M/G/1 Queue with Multiple Classes of Arrivals." Tech. rep., Rand Corporation Report.

Kaplan, E. H. (2010), "Terror Queues," *Operations Research*, 58, 773–784.

— (2012), "OR Forum - Intelligence Operations Research: The 2010 Philip McCord Morse Lecture," *Operations Research*, 60, 1297–1309.

Klimov, G. (1974), "Time-Sharing Service Systems I," *Theory of Probability & Its Applications*, 19, 532–551.

Kulkarni, V. G. (2009), *Modeling and analysis of stochastic systems*, CRC Press.

Lichtenthal, J. D., Sikri, S., and Folk, K. (1989), "Teleprospecting: an approach for qualifying accounts," *Industrial Marketing Management*, 18, 11–17.

Mandelbaum, A. and Stolyar, A. L. (2004), "Scheduling Flexible Servers with Convex Delay Costs: Heavy-Traffic Optimality of the Generalized $c\mu$-Rule," *Operations Research*, 52, 836–855.

Meyn, S. P. (1997), "The policy iteration algorithm for average reward Markov decision processes with general state space," *Automatic Control, IEEE Transactions on*, 42, 1663–1680.

Nain, P. (1989), "Interchange arguments for classical scheduling problems in queues," *Systems & control letters*, 12, 177–184.

Ni, K. S., Faissol, D., Edmunds, T., and Wheeler, R. (2013), "Exploitation of Ambiguous Cues to Infer Terrorist Activity," *Decision Analysis*, 10, 42–62.

Pinedo, M. (1983), "Stochastic Scheduling with Release Dates and Due Dates," *Operations Research*, 31, 559–572.

— (2008), *Scheduling Theory, Algorithms, and Systems*, Springer.

Porteus, E. L. (2002), *Foundations of stochastic inventory theory*, Stanford University Press.

Righter, R. (1994), "Scheduling," in *Stochastic Orders and Their Applications*, eds. Shaked, M. and Shanthikumar, J. G., Academic Press, Boston, pp. 381–432.

Sacco, W. J., Navin, D. M., Fiedler, K. E., Waddell, R. K., Long, W. B., and Jr, R. F. B. (2005), "Precise formulation and evidence-based application of resource-constrained triage," *Academics of Emergency Medicine*, 12, 759–770.

Saghafian, S., Hopp, W., Van Oyen, M., Desmond, J., and Kronick, S. (2011), "Complexity-Based Triage: A Tool for Improving Patient Safety and Operational Efficiency," *under review*.

— (2012), "Patient Streaming as a Mechanism for Improving Responsiveness in Emergency Departments," *Operations Research*, 60, 1080–1097.

Sennott, L. (1999), "Stochastic Dynamic Programming and the Control of Queueing Systems," .

Shumsky, R. A. and Pinker, E. J. (2003), "Gatekeepers and Referrals in Services," *Management Science*, 49, 839–856.

Smith, W. (1956), "Various Optimizers for Single Stage Production," *Naval Research Logistics Quarterly*, 3, 59–66.

Taghipour, S., Banjevic, D., and Jardine, A. (2011), "Prioritization of medical equipment for maintenance decisions," *Journal of the Operational Research Society*, 62, 1666–1687.

Tcha, D. and Pliska, S. (1977), "Optimal control of single-server queuing networks and multi-class M/G/1 queues with feedback," *Operations Research*, 25, 248–258.

Van Der Zee, S. and Theil, H. (1961), "Priority Assignment in Waiting-Line Problems under Conditions of Misclassification," *Operations Research*, 9, 875–885.

Van Mieghem, J. A. (1995), "Dynamic Scheduling with Convex Delay Costs: The Generalized $c\mu$ Rule," *The Annals of Applied Probability*, 5, 809–833.

Wang, X., Debo, L. G., Scheller-Wolf, A., and Smith, S. F. (2010), "Design and Analysis of Diagnostic Service Centers," *Management Science*, 56, 1873–1890.

Wilson, R. D. (2003), "Using online databases for developing prioritized sales leads," *Journal of Business and Industrial Marketing*, 18, 388–402.