Optimal Planning of Tiered Emergency Medical Services

by

Soovin Yoon

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Industrial and Systems Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2019

Date of final oral examination: 01/14/2019

The dissertation is approved by the following members of the Final Oral Committee:

Laura A. Albert, Associate Professor, Industrial and Systems Engineering
James R. Luedtke, Associate Professor, Industrial and Systems Engineering
Oguzhan Alagoz, Professor, Industrial and Systems Engineering
Gabriel Zayas-Caban, Assistant Professor, Industrial and Systems Engineering
Po-Ling Loh, Assistant Professor, Electrical and Computer Engineering

For my grandfather.

ACKNOWLEDGMENTS

First and foremost, I acknowledge my academic advisor Laura Albert for everything I learned from her. I am deeply indebted to her patience and dedication over the years. She has been not only a great thesis advisor but also an unparalleled role model for me in every aspect. I extend my acknowledgment to my dissertation committees James Luedtke, Oguzhan Alagoz, Gabriel Zayas-Caban, and Po-Ling Loh. Their valuable feedback on my dissertation proposal was essential to substantially improve this manuscript.

I am grateful to fellow students and post-docs in the Department of Industrial and Systems Engineering and Wisconsin Institutes for Discovery. In addition to Forough Enayati, Amanda Smith, and Eli Towle, I especially appreciate my academic siblings for the entire ups and downs of graduate student life we have been through together: Kaiyue Zheng for being a cheerful buddy and supportive mentor at the same time, Suzan Iloglu for her positive energy and being one of the most thoughtful person I've met, and Eric DuBois for consistently providing not only homemade sweets but also intellectual nourishment.

I thank my parents for being an endless source of wisdom and motivation. My special gratitude goes to my grandfather who always has been confident that I am ultimately going to be a "doc"—even when I was a slow-learning kid. Believing that his spell worked, I dedicate this dissertation to him. More than anything, I thank Wooyoung Kim for all the fun and troubles we shared, and for all the upcoming ones we will go through gladly as a team.

This dissertation would not have been at all possible without the help and support of many people. Any errors which remain are my sole responsibility. This manuscript is based upon work supported by National Science Foundation [Awards 1444219, 1361448]. The views and conclusions in this manuscript are those of the author and do not necessarily represent the official policies of the National Science Foundation.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In tiered emergency medical services (EMS) systems, in which multiple types of response vehicles are involved, how to match the resource to patients is a critical issue. Responses to emergency patients must be not only prompt but also capable of providing the type of services that patients require. I propose four discrete optimization models that design EMS systems to achieve both objectives.

This dissertation begins by studying the cutoff priority scheme, which gives priority to more emergent calls for service when the system is congested by reserving vehicles exclusively for high priority calls. An iterative framework composed of a spatial queuing approximation model and a Mixed Integer Linear Program, which is computationally efficient, evaluates and designs public safety systems with a cutoff priority scheme. The numerical analysis identifies the trade-off between improving the expected coverage for high priority calls at the expense of losing more low priority calls by introducing a cutoff. It also sheds light on how to identify the best number of vehicles to reserve, which can help EMS practitioners in setting the cutoff.

Next, I focus on emergency responses on a congested network with two types of ambulances. I develop a model that dynamically assigns emergency vehicles to patients based on the number of idle vehicles. This dissertation addresses these issue by formulating and studying a Markov decision process model that determines which type of ambulance to send to patients in real-time. The base model considers a loss system over a finite time horizon, and it dynamically assigns vehicles to patients. I also provide three extensions to the base Markov decision process model, including one that considers an infinite time horizon, a second that allows for patient queues, and a third that approximately incorporates spatial aspects of the problem. Structural properties of the optimal policy are derived to characterize the optimal resource assignment strategy and keep the problem computationally tractable. I prove for all models there exists an optimal policy that is class separable. I show the conditions under which there exists an optimal policy that is a signal threshold type and a state control limit type for the base model and extensions. Computational experiments using real-world

emergency response datasets show that the dynamic policy significantly improves the system performance by informing how to dynamically update ambulance-patient matching.

Furthermore, I extend and expand the decision context of the earlier model to a tandem queueing approach. The new model considers the service process in more detail by separating the response phase and the transport phase, in order to consider various options of dispatch, such as sending multiple vehicles to a single call (multiple response) or non-transport vehicles. Multiple response allows for a faster first-aid and successfully matches the patient's uncertain needs at the potential cost of making more vehicles unavailable at any given time. To investigate the value of multiple response, I formulate and implement a Markov decision process model that dynamically determines which type of vehicle(s) to dispatch based on resource availability and future demand volume. Additionally, I provide two extensions to the base Markov decision process model. First, the approximate spatial analogue of the base model studies how to design dynamic response districts and dynamically update the districts as resource availability changes. The second extension considers non-transport vehicles that must be accompanied by additional ambulances. The empirical study shows that the optimal policy sends ALS NTVs more frequently as the signal increases, and as often as possible after the signal exceeds the threshold.

Lastly, I propose a stochastic programming approach with focus on a tiered EMS system with two types of ambulances which can often employ multiple response. I provide a two-stage stochastic programming model that address how to deploy two types of ambulances in the first stage and dispatch them to prioritized emergency patients in the second stage after call arrival scenario is disclosed. I solve the model using sample average approximation and demonstrate the value of stochastic solution with a case study and a simulation analysis. Because stochastic programming models are computationally challenging for large-scale problem instances, a computationally effective solution method based on Benders cuts is also proposed. I additionally provide two model extensions to consider stochastic ambulance travel times, patient-ambulance matching utilities, and non-transport vehicles.

Successful investigation of suggested models enables the performance evaluation for tiered EMS systems and furnish policy insights for improving them. This dissertation can assist decisions makers to strategically plan and operate emergency vehicles in consideration of patient-resource matching.

1 INTRODUCTION

1.1 Background

Emergency medical services (EMS) systems provide pre-hospital care to those who are in need of urgent medical treatment and transports these patients to hospitals for more definitive care. When an emergency medical 911 call is received, the dispatch center estimates the severity of the call and dispatches appropriate vehicles to the scene. Once the medical unit arrives at the scene, on-scene treatment is provided. Patients are transported to the hospital if needed. Then the emergency vehicle returns to the base and becomes available for other emergency calls. In particular, I focus on an EMS system with multiple types of ambulances and multiple patient priority classes. Many EMS systems fall under this category, as emergency calls are often classified into priority groups based on their severity, and these systems' fleets consist of several types of medical units.

EMS systems typically contain various types of medical units with different capabilities (*tiered system*). Advanced life support (ALS) ambulances are staffed by paramedics who can provide advanced care, while basic life support (BLS) ambulances are staffed by basic emergency medical technicians. Some EMS systems also have vehicles other than ambulances for responding to 911 calls, such as non-transport quick-response vehicles (NTVs), fire vehicles, and even police cars (McLay, 2010). The matching of calls and emergency vehicles considering their types can be critical to the patient outcomes. For example, ALS treatment, as opposed to BLS treatment, to non-traumatic cardiac arrest patients can significantly improve patient survival (Bakalos et al., 2011).

Although the ultimate goal of emergency medical services is to provide prompt service to all patients, emergency medical personnel and ambulances are limited, and therefore, triage may be necessary to better match resources with patients. Calls with higher priorities are more time-sensitive, while other accident types with lower priorities can wait without significant deterioration in patient conditions. For example, the chance of survival for cardiac

arrest patients decreases by up to 10% for every additional minute of delay in treatment (Larsen et al., 1993). Therefore, it is important to efficiently ration limited resources, not only between different geographic locations but also among call priorities (McLay and Mayorga, 2013a).

To evaluate the performance of an EMS system, I apply the commonly used *coverage* performance measure, under which a call is considered to be *covered* when an ambulance reaches the patient in a given period of time, e.g., nine minutes from dispatch. This is called the response time threshold (RTT) and is set by most EMS departments. Although the ultimate goal is to maximize patient survivability, this is hard to measure, so these RTTs serve as proxies for patient survivability. Average response time is also a popular performance measure. Erkut et al. (2008) investigates the issue of measuring patient survival through response time. Alternative patient survival models are suggested by Larsen et al. (1993); Valenzuela et al. (1997); Waalewijn et al. (2001); Maio et al. (2003).

The promptness of response is critical for many patient conditions such as cardiac arrest since the condition of the patient can deteriorate when the service is delayed. EMS systems operate under uncertainty with respect to call arrival times, call origins and call severity. Resources such as medical vehicles and personnel are limited and become unavailable when they are serving other patients. Recently, rising call volumes and increasing medical costs make real-time resource allocation decisions even more challenging. The design of public safety systems is crucial for effectively using scarce emergency medical resources for responding to patients in a timely manner. Specifically, patients with time-critical conditions who require immediate care are more vulnerable to the delay of service resulting from congestion in the system. This motivates the development of response plans that depend on the level of available resources in the system as well as the specific needs of the patients.

1.2 Literature Review

Ambulance planning problems have been widely studied in the operations research literature. I classify related works in two groups: strategic ambulance planning and operational ambulance planning. The first group of models typically deal with long-term decisions including the size and composition of the fleet and the location of ambulances. Most research in this group models their problem using mixed linear integer programming.

Daskin (1983) introduces one of the first probabilistic location models that takes ambulance unavailability into account to maximize expected coverage by assuming that each ambulance has the same busy probability of being unavailable for service. Batta et al. (1989) extend Daskin's expected coverage model to account for the underlying spatial queueing dynamics of the ambulances using a Hypercube queueing model. Restrepo et al. (2009) present two models based on the Erlang loss formula that can be used to screen potential ambulance allocations. Few papers propose ambulance location models with multiple types of servers. Marianov and ReVelle (1992) develop models that locate two types of vehicles to maximizes the number of patients that can be served by both types. Mandell (1998) proposes a covering model with ALS and BLS servers focusing on the first-responder. McLay (2009) introduces a coverage model with ambulances and non-transport vehicles that serve multiple customer types. For a more extensive review on ambulance location models, I refer the readers to the review article by Brotcorne et al. (2003), and two recent reviews by Aringhieri et al. (2017) and Reuter-Oppermann et al. (2017).

The second group of decisions, operational ambulance planning problems, cover topics including ambulance dispatching, ambulance redeployment, and patient classification. Since these problems involve dynamic ambulance management, in which the real-time decision is required, the intractability of the problem is a critical issue.

The problem of classifying patients is widely studied in the literature using Markov Decision Process. Argon and Ziya (2009) consider priority assignment decisions for a service system under imperfect information on customer type identities. They provide insight into

what kind of information is useful and how to use that information to optimally classify customers. Jacobson et al. (2012) investigate how to determine patient priorities based on resource limitations and the scale of the event. Argon et al. (2011) provides a review of operations research approaches to improve patient triage and prioritization in the aftermath of mass casualty incidents (MCIs), classifying existing works to a group of linear programming approaches and a group of stochastic scheduling problems.

While the aforementioned papers investigate patient priority assignment in MCIs, this dissertation studies how to efficiently coordinate multiple types of vehicles when there is a varying payoff depending on the type of patient and the responding vehicles. Studies in MCIs triage focus on how to manage patients in a long queue when demand overwhelms supply. In contrast, the focus of this dissertation is on the low-traffic systems where the queue length of patients waiting for service is typically zero. McLay (2009) formulates a MILP model to study how to optimally locate two types of servers to improve patient survivability. Then the problem of dispatching servers to prioritized customers when there are classification errors in patient priorities is studied with an MDP model by McLay and Mayorga (2013a). Chong et al. (2016) construct an MDP model and an integer program to study the EMS system with both ALS and BLS units and high and low priority calls and examine the value of a mixed fleet system.

There is a stream of literature for ambulance redeployment. Maxwell et al. (2010) use approximate dynamic programming for the tractability of the ambulance redeployment dynamic program. Yue et al. (2012) propose a simulation-based approach for the dynamic redeployment problem and also discussed a stochastic upper bound on the performance of redeployment policies. Alanis et al. (2013) study ambulance repositioning using a compliance table policy by analyzing a two-dimensional Markov chain model. Ambulance redeployment is also studied by Kolesar and Walker (1974); Gendreau et al. (2001); Rajagopalan et al. (2008); Rettke et al. (2016).

Although the 'send the closest idle ambulance' policy is often assumed in the ambulance

location and relocation literature, some studies have discussed the potential improvement from using different dispatch rules. McLay and Mayorga (2013a) formulate an MDP model for determining how to optimally dispatch servers while taking account of equity as well as efficiency. Jagtenberg et al. (2017) propose an efficient heuristic for ambulance dispatching as an alternative to the 'closest idle ambulance' rule.

This dissertation fills several important knowledge gaps in the literature. In practice, even a very small EMS system maintains several types of vehicles and classify their demands into different priority groups. Emergency vehicles responding to emergency patients should be not only prompt but also capable of providing the appropriate type of service that patients require. I focus on the importance of *dynamic matching* between prioritized patients and heterogeneous servers in both long-term ambulance planning and real-time fleet management. The matching is called dynamic in a sense that the decision of which vehicle to dispatch must depend on the system status, such as server availability and varying demand volumes. Although the ambulance planning problem has been widely studied, a majority of papers assume homogeneous servers and demands and just focus on minimizing the response time. Existing models that consider tiered systems usually assume static matching with fixed priority. I notice that the optimal planning of dynamic matching in tiered EMS systems is underrepresented in the EMS literature, albeit practical to EMS applications. I cover this gap in the literature in this manuscript by proposing and implementing several discrete optimization models.

1.3 Dissertation Overview

Throughout this manuscript, the objective of this dissertation is to provide insights for designing an EMS system to efficiently manage limited medical resources to promptly and properly serve prioritized patients. EMS systems contain many challenges that cannot be addressed by a single model. A practical approach is to divide the problem into answerable

ones and solve each issue step by step. This dissertation examines several facets of EMS decision making in tiered systems from strategic decisions to operational decisions. I view patient-resource matching as a dynamic variable, not a fixed input, and study how to maximize patient coverage in a tiered EMS system with the dynamic matching. Chapters 2 – 5 are proposed to attain this objective. Overall, throughout all chapters, I provide various decision-making tools and insights for EMS providers in prioritizing and managing demands by efficiently operating limited resources.

Queueing non-urgent emergency calls when the number of available servers falls below the threshold is a strategy in use in Europe (Aringhieri et al., 2016). However, the cutoff priority scheme, which gives priority to more emergent calls for service when the system is congested by reserving capacity for high priority calls, has not been studied in the literature in the emergency response context. Therefore, in Chapter 2, I propose the use of cutoff priority queue to reserve resources exclusively for high priority customers when the system is congested. Under the cutoff priority scheme, the optimal system design has to be different from the design without the cutoff. Therefore, I investigate how to deploy and dispatch servers optimally with a cutoff priority queue by formulating an integer programming model.

The model proposed in Chapter 2 does not consider tiered resources, for instance, a mixed fleet of ALS and BLS vehicles. This motivates the next chapters to extend the cutoff priority queue scheme to a more sophisticated dynamic resource assignment for tiered systems.

Chapter 3 and 4 propose Markov decision process (MDP) models that dynamically assign resources to patients based on the resource availability to improve patient coverage. When an emergency call arrives, the dispatcher has to decide which servers to dispatch to the call. Responses should be rapid before the patient condition deteriorates. When there are multiple types of servers, a proper matching between the patient's health needs and the server type affects the patient outcomes. I aim to achieve both goals by dynamically reassigning resources to patients. A cutoff priority queue scheme, which is studied by the previous hypothesis, is a simple form of dynamic resource assignment in a non-tiered setting. Here, I focus on a

more complicated dynamic resource assignment that takes account of the resource level and other information available such as non-stationary demands and imperfect signals on patient conditions.

In Chapter 3, I consider the matching of prioritized patients to tiered resources. I formulate the problem as an MDP model and show its structural properties. Characterizing the optimal policy alleviates the curse of dimensionality. Moreover, it provides insights for decision makers in designing resource assignment heuristics that is simple yet effective. I also show structural properties of the optimal policy and conduct a numerical analysis. Then, Chapter 4 is based on a similar dynamic resource assignment idea, but additionally, consider sending multiple types of server to a single call. The service process is modeled as a tandem queue to expand the decision context. In both Chapter 3 and 4, to model the problem as an MDP, Poisson arrival and Exponential service time assumptions need to be justified. Moreover, the curse of dimensionality in MDP makes it difficult to explicitly examine the location of servers and calls. These issues motivate us to develop a different model that can implement dynamic matching by explicitly optimizing ambulance deployment and dispatching that involves fewer assumptions.

Chapter 5 aims to inform EMS system design decisions with fewer distributional assumptions using a data-driven stochastic programming model. EMS systems with prioritized patients and heterogeneous servers are intricate to study. First, EMS systems are exposed to the various sources of uncertainty, including the arrival time and origin of emergency calls, the severity of patients, and the availability of servers. Also, there exist interdependency between the servers' availability, and explicitly modeling the relationship between servers leads to a complicated queueing model or a nonlinear program that is computationally inefficient. Finally, stochastic ambulance planning models often require certain assumptions on call arrival or service time distribution, which are not guaranteed to be appropriate to all problem instances.

The data-driven approach resolves these factors. I directly sample call arrival scenarios

from the emergency call log, without requiring any assumption on the arrival or service time distribution. Similarly, the model routes ambulances to calls in each scenario instead of parameterizing the servers' busy probabilities. To assure that the sample can represent the population, the number of scenarios should be sufficiently large. I conduct a sample average approximation analysis to investigate the relationship between the scenario size and the optimality gap. A potential concern is that the size of the problem grows rapidly as I increase the sample size. Therefore, I propose a solution method based on a branch-and-benders-cut procedure to improve the solution time.

2 AN EXPECTED COVERAGE MODEL WITH A CUTOFF PRIORITY QUEUE

2.1 Introduction

Emergency medical service (EMS) systems provide pre-hospital care to those who are in need of urgent medical treatment and transports these patients to hospitals. The promptness of response is critical for many patient conditions such as cardiac arrest, since the condition of the patient can deteriorate when the service is delayed. As a result, the design of public safety systems is crucial for effectively using scarce emergency medical resources for responding to patients in a timely manner. In particular, patients with time-critical conditions who require immediate care are more vulnerable to the delay of service resulting from congestion in the system. This motivates the development of response plans that depend on the level of available resources in the system as well as the specific needs of the patients.

We consider a system with multiple ambulances and multiple patient priority classes. We apply the commonly used *coverage* performance measure, under which a call is considered to be *covered* when an ambulance reaches the patient in a given period of time, e.g., nine minutes from dispatch. Although the ultimate goal of emergency medical services is to provide prompt service to all patients, available emergency medical personnel and ambulances are limited, and therefore, triage may be necessary to better match resources with patients. Calls with higher priorities are more time-sensitive, while other accident types with lower priorities can wait without significant deterioration in patient conditions. For example, the chance of survival for cardiac arrest patients decreases by up to 10% for every additional minute of delay in treatment (Larsen et al., 1993). Therefore it is important to efficiently ration limited resources, not only between different geographic locations but also among call priorities (McLay and Mayorga, 2013a).

In this paper, we study how to give priority to more emergent calls for service when the

system is congested by reserving capacity for high priority calls, which is called the *cutoff priority scheme*. Existing ambulance location models generally assume that all calls are immediately served as long as there is an available vehicle (Goldberg, 2004). We lift this assumption by adopting a cutoff priority scheme. We limit our analysis to a two-priority case in this paper and hence we impose a single cutoff number for low-priority calls. In our model, a cutoff level is given for admission of low priority customers, so that low priority calls are either delayed or diverted to neighboring EMS departments (a process called "mutual aid") whenever the number of busy ambulances is equal to or greater than the cutoff. Therefore we always reserve a subset of servers exclusively to serve high priority patients in a timely manner. We also study system design by locating and dispatching ambulances to prioritized patients when there is a low priority cutoff. Mutual aid policies between adjacent EMS departments is commonplace in the United States, and queueing nonurgent emergency calls when the number of available servers fall below the threshold is a strategy in use in Europe (Aringhieri et al., 2016), however, the cutoff priority scheme has not been studied in the literature in the emergency response context. Under the cutoff priority scheme, the optimal system design differs from the design without the cutoff, which leads to improved coverage of emergency calls.

There is a rich operations research literature devoted to the design and evaluation of public safety systems such as fire and EMS. Early models in this area study how to locate vehicles such as ambulances to maximize coverage (Church and ReVelle, 1974), the most common performance measure for EMS systems. A number of papers consider the impact of congestion and backup service when locating vehicles. Daskin (1983) introduces one of the first probabilistic location models that takes ambulance unavailability into account to maximize expected coverage by assuming that each ambulance has the same busy probability of being unavailable for service. Batta et al. (1989) extend Daskin's expected coverage model to account for the underlying spatial queueing dynamics of the ambulances using a Hypercube queueing model. Other models that locate ambulances to maximize expected coverage allow

for multiple types of ambulances (McLay, 2009) and multiple ambulances per station (Ansari et al., 2015). Restrepo et al. (2009) present two models based on the Erlang loss formula that can be used to screen potential ambulance allocations. There is also a stream of literature for ambulance redeployment (Kolesar and Walker, 1974; Gendreau et al., 2001; Rajagopalan et al., 2008; Maxwell et al., 2010). However, all of these models assume that an ambulance is sent to a call if one is available and therefore do not ration service as we consider in this paper. For a more extensive review on ambulance location models, we refer the readers to the review article by Brotcorne et al. (2003), and two recent reviews by Aringhieri et al. (2017) and Reuter-Oppermann et al. (2017).

There is a stream of papers that develop spatial queueing models for evaluating the design of public safety systems. Larson develops an exact Hypercube queueing model to study the behavior of servers such as ambulances or fire engines (Larson, 1974) and provides an approximate Hypercube model that approximates the procedure by computing correction factors (Larson, 1975). Jarvis (1985) generalizes the approximate Hypercube model of Larson (1975) to account for location-specific service times and customer (patient) priorities. There are a variety of other models that lift the assumption in earlier Hypercube models to allow more practical settings (Burwell et al., 1993; Mendonça and Morabito, 2001; Iannoni and Morabito, 2007; Boyaci and Geroliminis, 2015; de Souza et al., 2015). These Hypercube models can evaluate various performance measures once a system has been designed, but they cannot construct the optimal system configuration. Instead Hypercube models are used to approximate the inputs for many location models introduced earlier, e.g., Batta et al. (1989), McLay (2009) and Ansari et al. (2015). One exception is Geroliminis et al. (2009) who develop a spatial queueing model that considers spatial characteristics and determines location and dispatch decisions internally by optimization. Another is Cho et al. (2014), who devise an integrated method with iterative problem relaxations and restrictions to endogenously estimate the ambulance busy probabilities in a nonlinear optimization model.

The models considered thus far do not consider cutoff priority queues. The cutoff priority

model was first formulated by Benn (1966) and detailed by Jaiswal (1968), who analyzes many cutoff priority model variations and their solutions. Taylor and Templeton (1980) improve Jaiswal's derivation for the cutoff priority model and solves balance equations to derive steady-state probabilities and waiting time distributions. They investigate two versions of the model: the SCQQ model with an infinite capacity queue for high priority calls and the SCQL model with a zero capacity queue for high priority calls. Both SCQQ and SCQL model maintain an infinite capacity queue for low priority calls. They apply the model to urban ambulance systems with two call priorities. Schaack and Larson (1986) take a decomposition approach that solves the system for any number of priority classes to obtain the customer waiting time distribution. These cutoff priority queueing models provide steady-state probabilities for the number of busy servers, which are one of the building blocks for our spatial queueing model. The relationship between these cutoff priority models to our work is detailed in Section 2.2 and Appendix A.1.

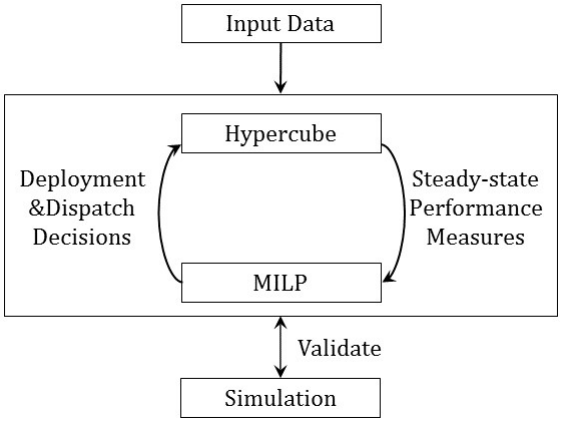
In this paper, we introduce a spatial queueing model and a Mixed Integer Linear Program (MILP) for evaluating and designing public safety systems with a cutoff priority scheme. A structural overview of the models is illustrated in Figure 2.1. The cutoff reserves capacity for high priority calls so that service is not delayed for critical emergency medical patients when the system is congested. In our models, the number of ambulances is fixed, and our performance measure is the expected coverage. To the best of our knowledge, this paper is the first to explicitly exploit the issue of differentiating the queueing policy for different priority demands in emergency systems and use the cutoff to inform system design.

This paper makes the following contributions.

1. We derive the approximate Hypercube queueing model with a cutoff priority scheme that is computationally easy to implement and solve. We demonstrate the accuracy of the model using discrete event simulation.

2. We use the the Hypercube model to prescribe how to locate and send ambulances to calls for service by using the approximate Hypercube model to estimate input parameters for

Figure 2.1: Overview of the model structure.



a MILP model we define. Solutions to the MILP model produce deployment and dispatching policies, which update inputs to the approximate Hypercube model. The two procedures alternatively exchange inputs and outputs until they reach convergence or infeasibility.

3. Computational examples illustrate how to implement the iterative procedure with real-world data. The results quantify the improvement in the expected coverage for high-priority patients using different levels of a cutoff. We demonstrate how to use the model to evaluate the tradeoff between expected coverage and the probability of losing or delaying calls when the cutoff is adjusted to help decision-makers set the value of the cutoff. The analysis illustrates how the number of reserved servers should decrease when the relative importance of low priority to high priority calls increases.

The remainder of this paper is organized as follows. Section 2 defines the notation and describes the approximate Hypercube model with cutoff priority queue. In section 3 the MILP model is introduced for locating and dispatching ambulances in an EMS system with a low priority cutoff. We describe the dataset and present computational results in Section 4 that validate the proposed models and report the model results. Then we conclude in Section 5.

2.2 Approximate Hypercube Model for Cutoff Priority Queue

In this section, we propose an approximate Hypercube model with cutoff priority queue. The main goal of the model is to estimate the performance measures including the server busy probabilities, dispatch probabilities, and steady-state probabilities for the number of busy servers through an approximation that is precise and computationally efficient.

We implement two novel features. First, calls are distinguished not only by their locations but also by their priorities. We assume that a call taker performs triage on each call and preassigns one of two priorities to each call depending on the nature of the emergency, either high priority or low priority. Calls with a high priority are more urgent and require a faster response. Next, low priority calls are cut off—either lost or entered into a queue—when the number of busy servers exceeds a threshold in order to reserve servers for high priority call arrivals when the system is congested.

A system with s servers and high (low) priority calls from a set of geographical locations J is considered. Servers are located at their home stations, with one server at a station. Since only a subset of stations has a server located, those stations are denoted as open stations. While servers have identical capabilities, they are distinguished by their home station. The models can allow for multiple servers per location, when there is a preference ordering for dispatching the servers at the same station to calls for service. High (low) priority calls arrive independent of the busy status of the servers according to a Poisson process with rate $\lambda^H(\lambda^L)$ per hour. The Poisson arrival assumption is frequently consistent with real-world datasets (de Souza et al., 2015; Kim and Whitt, 2014). Service is nonpreemptive, and the average service time for calls from j served by server i is τ_{ij}, which includes the travel time from the station to the scene, the time spent at the scene, the time to transport the patient to the hospital and return to the station. We assume exponential service times but this can be relaxed if the system is a Loss system (Jarvis, 1985). Exactly one server is assigned to a

call (when one is assigned) according to a fixed preference list, which is an ordered list of the open stations following the cutoff scheme that is described later. The kth preferred station for the high (low) priority call from node j is given as $b^H_{jk}(b^L_{jk}) \in I^{open}$ in the preference lists $\{b^H_{jk}\}(\{b^L_{jk}\})$. For example, if $b^H_{12} = 3$, then the second preferred server for a call from node 1 is a server in station 3 if the call is high priority.

A cutoff level of s_C is given for admission of low priority customers, with $s_C \leq s$. This means that low priority calls are either placed in a queue or lost whenever there are s_C or more busy servers, instead of being served immediately by the most preferred available server. High priority calls are immediately served by the most preferred available server and either put in a queue or lost only if all servers are busy. This implies that the cutoff level for high priority customers is set to s by convention (Schaack and Larson, 1986). In other words, $s_R = s - s_C$ servers are always reserved exclusively for high priority calls to be prepared for the future arrival.

We consider two types of model that differ by the queue capacity. In the *Loss model*, high priority calls that arrive when all servers are busy or low priority calls that arrive when there are more than or equal to s_C busy servers are "lost," and the queue has zero capacity $(M/M/s/s)$. Lost calls are assumed to be served by external resources, such as neighboring EMS departments through a process called "mutual aid." We also consider a *Queued model*, where the queue is assumed to have infinite capacity $(M/M/s/\infty)$. Calls are put in a queue when they cannot be served immediately when all servers are busy for high-priority calls and when there are no more than s_R servers available for low-priority calls. Calls are served later on a first-come-first-served basis when servers become free. In this model, high priority calls in the queue are served as soon as a server becomes free, and queued low priority calls are served when the system has more than s_R available servers. Because the queue is infinitely large, backlogged calls are never lost as they are eventually served if the system is stable. If the cutoff is set to an extremely low value in a relatively congested system, the system may not be stable for the low priority calls, and we revisit this case later in a computational study.

Here we show the derivation for both systems. Throughout the paper we mostly assume the Loss system, since it is common to assume a zero-line queue in the EMS literature, and it is also consistent with real-world operations (Budge et al., 2009).

Table 2.1 summarizes the notation used in this paper. The Hypercube queueing model requires deployment and dispatch policies as inputs, which can be generated either externally (for example, send-the-closest) or by forming the preference lists from the optimal solutions of the MILP model in Section 2.3. Other parameters needed such as service times τ_{ij} and arrival rates λ_j^p can be collected from the data. We present derivation details for the results presented in this section in Appendix A.1.

The outputs of the approximate Hypercube model are the correction factors $q_k^p,\ k = 0, \cdots, s-1,\ p \in \{H, L\}$, server busy probabilities $r_i,\ i \in I^{open}$, and steady-state probabilities $P_i,\ i = 0, \cdots, s$. These outputs are obtained through an iterative procedure that we introduce later in this section.

The correction factors q_k^p adjust the errors from treating servers as independent. To be specific about what correction factors do, let B_k denote a event that jth selected server is busy and $F_k \equiv B_k^c$ denote a event that jth server selected is free. Then f_{jk}^p, the probability that a priority p call from node j is served by its kth preferred server, can be expressed as $P\{B_1 B_2 \cdots B_{k-1} F_k\}$. If the servers operate independently from one another, then the formulation for this term would be $\Pi_{l=1}^{k-1} r_{b_{jl}^p}(1 - r_{b_{jk}^p})$. However, the independence assumption is erroneous because the busy status of a server affects the probability of other servers being busy. Therefore, correction factors q_k^p, which we derive later, are needed to improve this approximation. Once we have the correction factors and server busy probabilities $r_i,\ i \in I^{open}$, the dispatch probability f_{jk}^p, which is the probability that a priority p call from node j is served by its kth preferred server, can be approximated as

$$f_{jk}^p = q_{k-1}^p \Pi_{l=1}^{k-1} r_{b_{jl}^p}(1 - r_{b_{jk}^p}), \qquad \forall j \in J,\ k = 1, \cdots, s,\ p \in \{H, L\}, \tag{2.1}$$

Table 2.1: Summary of the notation.

symbol	**description**
$j \in J$	set of call source nodes
$i \in I$	set of potential stations (used in the MILP)
$I^{open} \subset I$	set of open stations (used in the Hypercube model)
$p \in \{H, L\}$	set of call priorities
s	total number of servers
s_C	cutoff level for low priority calls
s_R	number of reserved servers, $s_R = s - s_C$
r_i	steady-state busy probability of server at station $i \in I$
r	steady-state system-wide average server busy probability, $r = \sum_i r_i/s$
P_i	steady-state probability that $i \in I$ servers are busy
q_k^p	correction factor for sending the kth preferred server, $k = 0, \cdots, s-1$, to priority $p \in \{H, L\}$ call
f_{jk}^p	probability of dispatching kth preferred server, $k = 1, \cdots, s$ to priority $p \in \{H, L\}$ call from node $j \in J$
f_k^p	system-wide average probability that the kth preferred server is dispatched to a priority p call
$a_{ij}^p \in \{1, \cdots, k\}$	preference order of server $i \in I$ by a priority $p \in \{H, L\}$ call from node $j \in J$
$b_{jk}^p \in I^{open}$	the station where the kth preferred server by priority $p \in \{H, L\}$ call from node $j \in J$ is located
λ_j^p	arrival rate of priority $p \in \{H, L\}$ call from node $j \in J$
$\lambda^H(\lambda^L)$	arrival rate of high(low) priority calls
λ	system-wide average call arrival rate, $\lambda = \lambda^H + \lambda^L = \sum_{j \in J} \sum_{p \in \{H,L\}} \lambda_j^p$
τ_{ij}	mean service time for calls from node $j \in J$ served by a server from station $i \in I$
τ	system-wide average mean service time. The initial value is given as $\tau^0 = \sum_{j \in J} \sum_{i \in I^{open}} \sum_{p \in \{H,L\}} \tau_{ij}(\lambda_j^p/\lambda^p)$, and this value is updated later in the algorithm
ρ	server utilization factor, $\rho = \lambda\tau/s$
w	relative weight of the low priority call coverage to high priority call coverage in the objective function
R_{ij}	probability that an ambulance located at station $i \in I$ successfully reaches the call from node $j \in J$ in a prespecified time limit
N_{ij}	set of call nodes that are neighbors to node j and are closer to station i than j
ϵ	server busy probability convergence threshold (used in the Hypercube model)
δ	imbalance measures that equals to standard deviation of server busy probabilities
γ	server workload imbalance threshold (used in the MILP)

for all $j \in J$ and $k = 1, ..., s$. Therefore, the correction factors restore the true dispatch probability f_{jk}^p when multiplied by the product of independent server busy probabilities.

We begin by providing the closed-form expression for the correction factors as

$$q_k^H = \frac{\sum_{i=k}^{s-1} \frac{(s-k-1)!(s-i)i!}{(i-k)!s!} P_i}{\nu^k \rho^k \big(1 - \nu\rho\big)}, \tag{2.2}$$

$$q_k^L = \frac{\sum_{i=k}^{s_C-1} \frac{(s-k-1)!(s-i)i!}{(i-k)!s!} P_i}{\nu^k \rho^k \big(1 - \nu\rho\big)}, \tag{2.3}$$

for every $k = 0, \cdots, s-1$, in which

$$\nu = \begin{cases} (1 - P_s \frac{\lambda^H}{\lambda} - \sum_{i=s_C}^{s} P_i \frac{\lambda^L}{\lambda}) & \text{for the Loss model} \\ 1 & \text{for the Queued model} \end{cases} \tag{2.4}$$

where ν refers to the probability that a call is immediately served. The derivation of q_k^p is detailed in Appendix A.1.

To compute these correction factors, we need to obtain the steady-state probabilities P_i, the probability that i servers are busy, $i = 0, \cdots, s$, which also are performance measures of interest in this Hypercube queueing model. In terms of describing the steady-state probabilities, the state is defined as the number of busy servers in the system. For the Loss model, we derive the closed-form expressions as follows.

$$P_i = \begin{cases} P_0 \frac{s^i \rho^i}{i!} & 0 \le i \le s_C - 1 \\ P_0 \frac{s^i \rho^i}{i!} (\frac{\lambda^H}{\lambda})^{i - s_C} & s_C \le i \le s \end{cases} \tag{2.5}$$

where

$$P_0 = \left(\sum_{i=0}^{s_C-1} \frac{s^i \rho^i}{i!} + \sum_{i=s_C}^{s} \frac{s^i \rho^i}{i!} (\frac{\lambda^H}{\lambda})^{i-s_C} \right)^{-1}. \tag{2.6}$$

For the Queued model, we refer the reader to Taylor and Templeton's work on cutoff

priority queues (Taylor and Templeton, 1980) for the detail of the derivation. They solve a series of balance equations to obtain the steady-state probabilities. Our Queued model refers to their SCQQ model so that both high and low priority calls are put in an infinite capacity queue when it is not possible to immediately dispatch a server. We denote $\xi_1 = \lambda^H \tau$, $\xi_2 = \lambda^L \tau$, and $\xi = \lambda \tau$, where τ is the system-wide mean service time. Then the steady-state probabilities are

$$P_i = \begin{cases} P_0 \frac{\xi^s}{s!} & 0 \le i \le s_C - 1 \\ P_0 (\xi^{s_C} \xi_1^{i-s_C} / i!) s_C / (s_C - \xi_2 S_Q(s_C, s)) & s_C \le i \le s - 1 \\ P_0 (\xi^{s_C} \xi_1^{i-s_C} / i!) [s_C / (s_C - \xi_2 S_Q(s_C, s))] s / (s - \xi_1) & i = s \end{cases} \tag{2.7}$$

where

$$P_0 = \left(\sum_{i=0}^{s_C - 1} \xi^i / i! + (\xi^{s_C} / (s_C - 1)!) S_Q(s_C, s) / [s_C - \xi_2 S_Q(s_C, s)] \right)^{-1}, \tag{2.8}$$

$$S_Q(k, s) = \xi_1^{-k} k! [\sum_{i=k}^{s-1} \xi_1^i / i! + (\xi_1^s / s!) s / (s - \xi_1)]. \tag{2.9}$$

In addition to the steady-state probabilities, we also evaluate the server busy probabilities r_i to obtain all the terms needed to compute the correction factors in (2.2) and (2.3). As in other approximate Hypercube models (Larson, 1975; Jarvis, 1985), we first obtain the expressions for server busy probabilities by applying Little's Law, as

$$\begin{aligned} r_i &= \sum_p \sum_j \lambda_j^p \tau_{ij} f^p_{j a^p_{ij}} \\ &= \sum_p \sum_j \lambda_j^p \tau_{ij} q^p_{a^p_{ij} - 1} \Pi_{l=1}^{a^p_{ij} - 1} r_{b^p_{jl}} (1 - r_{b^p_{j a_{ij}}}), \qquad \forall i \in I^{open}, \end{aligned} \tag{2.10}$$

where $a^p_{ij} = k$ refers that i is the kth preferred server by priority p call from node j. Then we rearrange the above equalities to write

$$r_i = \frac{V_i}{V_i + 1}, \qquad \forall i \in I^{open}, \tag{2.11}$$

where

$$V_i = \sum_{i \in I^{open}} \sum_{k \in K} \sum_{p \in \{H,L\}} \lambda_j^p \tau_{b_{jk}^p j} q_{k-1}^p (\Pi_{l=1}^{k-1} r_{b_{jl}^p}), \qquad \forall i \in I^{open}. \tag{2.12}$$

We can either compute the dispatch probabilities f_{jk}^p using individual server busy probabilities $r_i,\ i = 1, ..., s$ as in (2.1), or approximate the same quantity using the system-wide busy probability $r = \sum_i r_i/s$, as $f_{jk}^p \approx q_{k-1}^p (1-r) r^{k-1}$. This latter approximation becomes more precise when the individual server busy probabilities are close in magnitude, which occurs when the workload is balanced among the servers. In Section 2.4, we bound the server workload imbalance to be no more than 2% between ambulances in the computational study, which justifies the use of this approximation.

In a manner that is similar to Jarvis (1985), we use the above quantities to update system-wide service time τ as

$$\tau = \sum_{i \in I^{open}} \sum_{j \in J} \tau_{ij} \Big(\frac{\lambda_j^H}{\lambda} f_{ja_{ij}^H}^H (1 - P_s) + \frac{\lambda_j^L}{\lambda} f_{ja_{ij}^L}^L (1 - \sum_{i=s_C}^{s} P_i) \Big). \tag{2.13}$$

This update of τ is necessary as we have updated dispatch probabilities.

Finally, we summarize the iterative procedure for approximate Hypercube model with a cutoff priority queue, including the initialization step and termination criteria.

STEP 0 Given call arrival rates λ_j^p, service times τ_{ij} from inputs, open stations I^{open} and preference list b_{jk}, and the server busy probability convergence threshold ϵ, initialize system-wide service time τ and utilization factor ρ as

$$\tau^0 = \sum_{j \in J} \sum_{i \in I^{open}} \sum_{p \in \{H,L\}} \tau_{ij} \frac{\lambda_j^p}{\lambda^p}, \tag{2.14}$$

$$\rho = \frac{\lambda \tau}{s}. \tag{2.15}$$

STEP 1 Update steady-state probabilities P_i using (2.5) and (2.6) for the Loss model, or using (2.7), (2.8), and (2.9) for the Queued model.

STEP 2 Update the correction factors q_k^p using (2.2) and (2.3), where ν follows (2.4) for the Loss model or $\nu = 1$ for the Queued model.

STEP 3 Update the server busy probabilities r_i using (2.11) and (2.12).

STEP 4 Update the dispatch probabilities f_{jk}^p using (2.1), and normalize them by

$$f_{jk}^H \leftarrow \frac{f_{jk}^H(1-P_s)}{\sum_{k\in K} f_{jk}^H}, \qquad \forall j \in J, k = 1, \cdots, s, \tag{2.16}$$

$$f_{jk}^L \leftarrow \frac{f_{jk}^L(1-\sum_{i=sC}^{s} P_i)}{\sum_{k\in K} f_{jk}^L}, \qquad \forall j \in J, k = 1, \cdots, s. \tag{2.17}$$

Update the system-wide service time τ using (2.13) with the updated dispatch probabilities.

STEP 5 Update the utilization factor $\rho = \frac{\lambda\tau}{s}$ and system-wide busy probability $r = \nu\rho$, where ν has the same value as STEP 2. Normalize the busy probabilities by

$$r_i \leftarrow \frac{r_i}{\sum_{l\in I^{open}} r_l/s} r, \qquad \forall i \in I^{open} \tag{2.18}$$

Check if the maximum change in r_i between iterations is smaller than the server busy probability convergence threshold ϵ. If yes, terminate and return r_i for $i \in I^{open}$, q_k^p (and P_i for $i = 0, \cdots, s$, if the Queued model is used). If not, go back to STEP 1.

There are no proven theoretical results or bounds that guarantee the convergence of this procedure in general without further assumptions. However, our numerical experiments have demonstrated a fast convergence in a small number of iterations when the cutoff is not too low. Other Hypercube models also report fast convergence in their numerical studies (Larson, 1975; Jarvis, 1985).

Note that the model described in this section is an approximate model. Therefore, we demonstrate its accuracy in Section 2.4 by comparing the Hypercube model approximations to the results of a discrete event simulation in a computational study.

2.3 Maximum Expected Coverage Model

In this section we present a mixed integer linear program model that simultaneously locates servers at stations and dispatches servers to call nodes. The goal of the model is to maximize the expected coverage over the region. A call is considered to be covered if a server is dispatched to the call immediately upon the arrival and it reaches the call in the predetermined time limit (e.g., 9 minutes). The expected coverage is defined as the average proportion of calls successfully covered given the deployment and dispatching policy. The expected coverage is weighted over high priority and low priority calls, and it uses the busy probabilities and correction factors from the the approximate Hypercube model. The MILP model distinguishes between different call priorities in determining ambulance location and dispatch policies, and it allows for a cutoff priority scheme. The MILP model adapts the ambulance location and dispatch model of Ansari et al. (2015) and extended to involve cutoff priority scheme.

The inputs to the MILP model are busy probabilities and correction factors from the approximate Hypercube model. The MILP solution outputs are the set of open stations and the preference lists for each call priority, which then become inputs for the Hypercube model. Therefore, we alternatively solve the approximate Hypercube model described in the previous section and the MILP model, since each model's outputs provides the inputs to the other model. At the end of this section, we provide a Hypercube-MILP iterative procedure where the two models alternatively run, thus updating the inputs for each model until the server workloads are approximately balanced.

Both the Loss and the Queued models use the same MILP model illustrated in this section, except for one set of constraints that enforce load balancing among the servers. The Hypercube model returns different values of correction factors and server busy probabilities for various sizes of the queue, and these values are incorporated into the MILP model to generate deployment and dispatch decisions optimized for the type of the model.

We select s stations from a set of I potential stations for locating servers. The binary decision variables capture the public safety system design decisions. Let $y_i = 1(0)$ if a server

is (not) located at station $i \in I$. This variable set defines the set of open stations. Let $x_{ijk}^p = 1(0), i \in I, j \in J, k = 1, \cdots, s$ if station i is (not) the kth preferred station for a priority p call from node j. This variable set captures the dispatch policy for all types of calls. Note that the resulted optimal dispatch plan may deviate from the natural sending-the-closest-idle policy (see also McLay and Mayorga (2013a) and Jagtenberg et al. (2017)).

The mixed integer linear program is formally stated below.

$$\max_x \quad \sum_{i\in I}\sum_{j\in J}\sum_{k=1}^{s}\left(c_{ijk}^H x_{ijk}^H + w c_{ijk}^L x_{ijk}^L\right) \tag{2.19}$$

subject to

$$\sum_{i\in I} y_i = s, \tag{2.20}$$

$$\sum_{k=1}^{s} x_{ijk}^p = y_i, \qquad \forall i \in I, j \in J, p \in \{H, L\}, \tag{2.21}$$

$$\sum_{i\in I} x_{ijk}^p = 1, \qquad \forall j \in J, k = 1, \cdots, s, p \in \{H, L\}, \tag{2.22}$$

$$\sum_{j\in J}\sum_{k=1}^{s}\sum_{p\in\{H,L\}} \lambda_j^p q_{k-1}^p (1-r) r^{k-1} \tau_{ij} x_{ijk}^p \le (r+\delta) y_i, \qquad \forall i \in I, \tag{2.23}$$

$$\sum_{j\in J}\sum_{k=1}^{s}\sum_{p\in\{H,L\}} \lambda_j^p q_{k-1}^p (1-r) r^{k-1} \tau_{ij} x_{ijk}^p \ge (r-\delta) y_i, \qquad \forall i \in I, \tag{2.24}$$

$$x_{ij'1}^H \ge x_{ij1}^H, \qquad j \in J, i \in I, j' \in N_{ij}, \tag{2.25}$$

$$y_i \in \{0, 1\}, \qquad i \in I, \tag{2.26}$$

$$x_{ijk}^p \in \{0, 1\}, \qquad i \in I, j \in J, k = 1, \cdots, s, p \in \{H, L\}. \tag{2.27}$$

The objective function in (2.19) represents the weighted expected coverage over high priority calls and low priority calls. The weight parameter w reflects the relative significance of low priority calls in the decision maker's objective compared to high priority calls (which have an implicit weight of 1). Without loss of generality we assume $w \in [0, 1)$ as it is more

important to cover time-sensitive high priority calls. The coefficient c^p_{ijk} used in the objective function is computed as

$$c^p_{ijk} = q^p_{k-1}(1-r)r^{k-1}R_{ij}\frac{\lambda^p_j}{\lambda^H + w\lambda^L}, \qquad \forall i \in I, j \in J, k = 1, \cdots, s, p \in \{H, L\}. \tag{2.28}$$

This represents the weighted fraction of calls of priority p at node j that are reached in the prespecified time limit by server i if it is the kth preferred server. The objective coefficient (2.28) can be divided into three components. The first component $q^p_{k-1}r(1-r)^{k-1}$ is the probability that the kth preferred server is dispatched to a priority p call. This component captures ambulance unavailability and uses the system-wide server busy probability r and the correction factor q^p_{k-1}, which are the output from approximate Hypercube model. This yields an approximation to the queueing probabilities that compensates for the independence assumption between the servers. The second component R_{ij}, which captures travel time uncertainty, represents the probability that an ambulance located at station i reaches a call at node j within the specified time limit (Ingolfsson et al., 2008). The first two components together ($q^p_{k-1}(1-r)r^{k-1}R_{ij}$) capture the fraction of calls at node j covered by station i when i is the kth preferred server for calls at node j. The final component weighs calls by the demand (call volume) and the priority and normalizes the weights, thus yielding a weight of $\lambda^p_j/(\lambda^H + w\lambda^L)$ for priority $p \in \{H, L\}$ calls at node $j \in J$.

The cutoff priority queue scheme is captured in the MILP model through server busy probabilities and the correction factors. To capture calls being cut off, the low priority correction factor q^L_{k-1} is set to zero for all $k > s_C$. Therefore, the objective coefficient $c^L_{ijk} = 0$ when $k > s_C$, since the cutoff priority queue discipline restricts the Hypercube model to always return $q^L_{k-1} = 0$ for $k > s_C$. As a result, x^L_{ijk} for $k > s_C$ is "inactive" in practice, since the cutoff prevents dispatching to low priority calls when k or more ambulances are busy. We keep the variables x^L_{ijk} for $k > s_C$ as it simplifies notation and does not affect the objective value or the feasibility set. Note that $\{x^p_{ijk}\}$ is a binary representation of the preference list

$\{b^p_{jk}\}$. Once the MILP problem is solved, the preference lists for high and low-priority patients are set using the optimal MILP solution variables as $b^p_{jk} \leftarrow \{i|x^H_{ijk} = 1\},\ j \in J, k = 1, ..., s$ and $b^L_{jk} \leftarrow \{i|x^L_{ijk} = 1\},\ j \in J, k = 1, ..., s_C$.

We next describe the constraints in the model. A station is considered to be open if a server is located at it. Exactly s stations are open, which is guaranteed by (2.20). A server from station i is dispatched only when the station i is open, which is enforced by (2.21). Constraint set (2.22) requires every call node to have an ordered, non-overlapping preference list of available servers. Constraint sets (2.26) and (2.27) require the variables to be binary.

Constraint sets (2.23) and (2.24) are load balancing constraints that set upper and lower limits for server utilization. The server utilization for each open station is required to be within a tolerance δ of the average server utilization. These constraint sets have two implications. First, the program uses common busy probability r instead of $r_i, i \in I^{open}$ with an implicit assumption that the busy probability is the same for all servers. This assumption is reasonable, since (2.23) and (2.24) ensure that service providers spend approximately the same proportion of time serving calls, which is desirable for service providers maintaining their skill proficiency (McLay and Mayorga, 2013a).

Constraint sets (2.23) and (2.24) are valid for the Loss model. For the Queued model, we replace (2.23) and (2.24) with the following constraints:

$$\sum_{j\in J}\sum_{k=1}^{s}\sum_{p\in\{H,L\}} \lambda^p_j q_{k-1}(1-r)r^{k-1}\tau_{ij}x^p_{ijk} \leq (r+\delta-\frac{\lambda\tau P_D}{s})y_i, \qquad \forall i \in I, \tag{2.29}$$

$$\sum_{j\in J}\sum_{k=1}^{s}\sum_{p\in\{H,L\}} \lambda^p_j q_{k-1}(1-r)r^{k-1}\tau_{ij}x^p_{ijk} \geq (r-\delta-\frac{\lambda\tau P_D}{s})y_i, \qquad \forall i \in I, \tag{2.30}$$

where the parameter $P_D = P_s\frac{\lambda^H}{\lambda} + \sum_{i=s_C}^{s} P_i\frac{\lambda^L}{\lambda}$ captures the probability of a call being delayed and put in an infinite capacity queue. The additional term $\frac{\lambda\tau P_D}{s}$ represents the workload of serving delayed calls in the queue. The steady-state probabilities P_i for $i = 0, \cdots, s$ can be computed using the Hypercube model, along with correction factors and server busy

probabilities.

Finally, constraint set (2.25) requires the first priority districts to be contiguous so that the resulting dispatching policy yields contiguous first priority response districts for each open station, and it results in contiguity as follows. When the first preferred station of high priority calls from node j is station i, then constraint set (2.25) requires a node j' that is a neighbor of j and closer to station i ($j' \in N_{ij}$) to also have its first preferred station for high priority calls to be i ($x^H_{ij'1} = 1$). This requirement yields a contiguous district surrounding i.

The system-wide server busy probability r is treated as an exogenously provided input parameter in the objective coefficients as well as load balancing constraints. However, r is endogenous to the model in that it is a function of the decision of deployment and dispatching decisions. To maintain model tractability, a second iterative scheme is introduced. As mentioned in the previous section, the approximate Hypercube model generates the correction factors q^p_k and server busy probabilities r used as inputs for MILP model in the first iterative procedure. Then the solutions from the MILP model become new inputs for the Hypercube model that updates the value of correction factors and server busy probabilities. Again, those outputs are utilized as inputs for the MILP model. The second iterative procedure continues until the outputs converge, and server busy probabilities are approximately balanced. As this iterative procedure is an approximate algorithm, later we validate its accuracy through computational study in Section 2.4.

Below we detail the steps in iterative algorithm, which adopts the procedure introduced by Ansari et al. (2015) with a cutoff priority scheme. It is notable that this procedure is guaranteed to converge (termination criteria (1)) unless it leads to infeasibility (termination criteria (2)). In STEP 3, the imbalance parameter δ is updated with the standard deviation of $r_i,\ i \in I$, and it always decreases because its value from the previous iteration set upper and lower bounds for the server busy probabilities $r_i,\ i \in I$ through the load balancing constraint sets in MILP of STEP 1. Therefore, it is ensured that δ has shrinking margin every iteration, and the procedure terminates at the point that δ falls under the predetermined threshold

value γ or it imposes a tight load balancing constraint that is too strong to be feasible (see Ansari et al. (2015)).

STEP 0 Initialize:

Generate the initial preference lists b_{jk}^p, by always sending the closest available server. Set correction factors $q_k^p = 1$ for all $k = 1, \cdots, s$, server busy probabilities $r = r_i = \rho = \frac{\lambda\tau}{s}$ for all $i \in I^{open}$, where the system-wide mean service time τ is set to $\tau = \lambda^{-1} \sum_{i \in I} \sum_{j \in J} \sum_{p \in \{H,L\}} \lambda_j^p \tau_{ij}$. Set the imbalance tolerance to $\delta = 1.0$.

STEP 1 Solve the MILP model with inputs r, q_k^p (and P_D if the Queued model is used) and update preference lists b_{jk}^p and set of open stations I^{open}.

STEP 2 Solve the approximate Hypercube model with inputs b_{jk}^p and I^{open} to update server busy probabilities r and correction factors q_k^p.

STEP 3 At the 1st iteration, update $\delta = r^{max} - r^{min}$ and go back to STEP 1. Otherwise, update the imbalance parameter δ with the standard deviation of $r_i, i \in I^{open}$. Check termination criteria: stop if (1) the imbalance measure δ is less than the prespecified server workload imbalance threshold γ or if (2) MILP was infeasible. Else if both termination criteria are not satisfied, go back to STEP 1.

2.4 Computational Results

This section reports analytical and discrete event simulation results that validate the models and illustrate the effectiveness of introducing the cutoff priority queue scheme for emergency medical service systems. While we mainly consider the Loss model, we also report computational results for the Queued model for comparison. The validation for how accurately the proposed Hypercube model approximates system performance measures of the system with a cutoff is provided through comparison to simulation results. Throughout this section we use

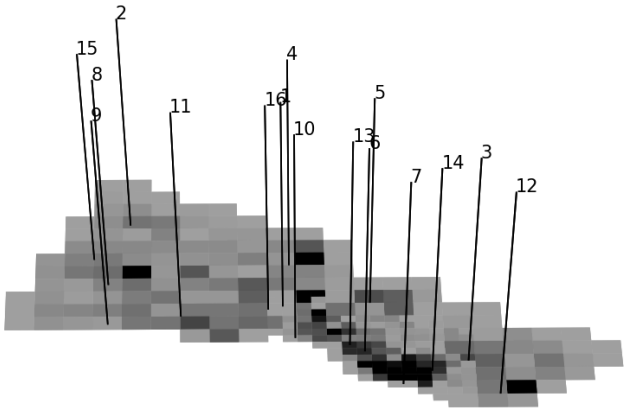
$s_R = s - s_C$, the number of servers reserved exclusively for high priority calls, in describing the cutoff priority scheme.

Computational Setting

The model is studied using real-world data set from Hanover County, Virginia (McLay and Moore, 2012). The data describes a geographical region with 16 potential stations, 270 aggregated demand nodes with 139 2×2 mile cells and 131 1×1 mile cells. The calls during the weekdays from 12PM to 6PM over 31 months is used to estimate the call arrival rates (demand). The call arrival rate is approximately constant over this time period. The demand over the region is visualized in Figure 2.2 as a colored cells on a grid, where a darker color represents a higher arrival rate. The arrival rate over the area is not uniform, as it is concentrated in the middle and south-east areas of the county while the arrival rates are low on the west side. The location of 16 potential stations are also marked in the map. The base case considers $s = 5$ servers.

To investigate the effect of reserving capacities in a large fleet system, we also consider a "scaled-up" version of the Hanover County dataset with a larger fleet. To generate the scaled-up dataset, the number of servers increases from 5 to 16, and the call arrival rates is scaled by a factor of 5.9. The demand scale parameter 5.9 is selected so that the system shows the similar coverage performance to the original 5-server case when there is no cutoff ($s_C = s$). Note that we do not choose the scale to fix the ratio between the demand volume and the number of servers. As a result, the server utilization factor of the large-scale dataset is 1.84 times greater than the original dataset. This large-scale dataset represents a higher traffic system with more resources compared to the original dataset. The effect of the reserved capacity on coverage appears different in low traffic and high traffic systems. By choosing the scale parameter that lines up the performance of baseline models where there is no cutoff, we enable a direct comparison of the coverage increment trend between the two systems. Importantly, this choice of a scale parameter results in a traffic intensity that employs the

Figure 2.2: Demand Distribution and Station Location on the Hanover County Map.



cutoff under a variety of settings and allows us to study a range of values for the cutoff.

The Hypercube model and the MILP model in the iterative procedure were implemented in Python 2.7.8 and solved using Gurobi 7.0.1. The running time of the Loss model for the iterative MILP-Hypercube procedure using Hanover County data is 27.13 seconds on average, and the time required for running the Hypercube models adds less than 0.11 seconds for all cases. The runtime for the scaled-up data with a large fleet is 40.87 seconds on average, and the Hypercube model runtime always adds less than 0.5 seconds. The load imbalance tolerance threshold γ in the termination criteria of the iterative procedure is set to 2% for the original dataset and 15% for the scaled dataset, and server busy probability convergence threshold ϵ is set to 1% for both. The computational time performance and parameter settings for the Queued model is analogous to the Loss model.

Model Validation

We compare analytical results obtained by the approximate Hypercube model to discrete-event simulation results using the Loss model to demonstrate the accuracy of the approximate Hypercube model. The simulation uses the same demand distribution and geographical data as the analytical model, and the resulting deployment and dispatch decisions from the analytical model are also used as policy inputs for the simulation. Note that unlike the Hypercube model and the MILP model that are iterative, the simulation only receives

Table 2.2: High Priority Dispatch Probabilities f_k^H and Loss Probabilities for High Priority Calls $P(\text{lost}|H)$ for Different Values of s_R.

	Hypercube					Simulation				
	$s_R=0$	$s_R=1$	$s_R=2$	$s_R=3$	$s_R=4$	$s_R=0$	$s_R=1$	$s_R=2$	$s_R=3$	$s_R=4$
$k=1$	0.6669	0.6770	0.6961	0.7326	0.7857	0.6594	0.6700	0.6903	0.7272	0.7813
$k=2$	0.2012	0.2013	0.2040	0.1978	0.1724	0.1980	0.1990	0.2007	0.1931	0.1664
$k=3$	0.0696	0.0703	0.0664	0.0520	0.0316	0.0732	0.0733	0.0695	0.0568	0.0379
$k=4$	0.0280	0.0279	0.0219	0.0116	0.0068	0.0316	0.0309	0.0253	0.0153	0.0099
$k=5$	0.0131	0.0135	0.0065	0.0035	0.0020	0.0150	0.0162	0.0088	0.0049	0.0029
$P(\text{lost}\|H)$	0.0212	0.0100	0.0050	0.0026	0.0016	0.0227	0.0106	0.0053	0.0027	0.0016

Table 2.3: Low Priority Dispatch Probabilities f_k^L and Loss Probabilities for Low Priority Calls $P(\text{lost}|L)$ for Different Values of s_R.

	Hypercube					Simulation				
	$s_R=0$	$s_R=1$	$s_R=2$	$s_R=3$	$s_R=4$	$s_R=0$	$s_R=1$	$s_R=2$	$s_R=3$	$s_R=4$
$k=1$	0.6709	0.6658	0.6293	0.5154	0.2798	0.6655	0.6583	0.6222	0.5072	0.2742
$k=2$	0.1987	0.1889	0.1487	0.0729	0.0000	0.1994	0.1896	0.1503	0.0732	0.0000
$k=3$	0.0689	0.0570	0.0281	0.0000	0.0000	0.0708	0.0582	0.0286	0.0000	0.0000
$k=4$	0.0275	0.0153	0.0000	0.0000	0.0000	0.0284	0.0162	0.0000	0.0000	0.0000
$k=5$	0.0127	0.0000	0.0000	0.0000	0.0000	0.0133	0.0000	0.0000	0.0000	0.0000
$P(\text{lost}\|L)$	0.0212	0.0731	0.1939	0.4117	0.7202	0.0226	0.0777	0.1990	0.4196	0.7258

Table 2.4: System-wide Server Busy Probability r for Different Values of s_R.

	Hypercube					Simulation				
	$s_R=0$	$s_R=1$	$s_R=2$	$s_R=3$	$s_R=4$	$s_R=0$	$s_R=1$	$s_R=2$	$s_R=3$	$s_R=4$
r	0.3303	0.3216	0.3026	0.2628	0.2085	0.3367	0.3281	0.3066	0.2668	0.2115

deployment and dispatch decisions from the MILP model after convergence, and the simulation output is compared to the Hypercube model output without further iteration. The program was implemented in Python 2.7.8, ran until 100,000 calls arrive, and repeated 30 times to generate summary statistics for dispatch probabilities, server busy probabilities, steady-state probabilities and coverage over the high priority calls. Since the system of concern is relatively low-traffic, and the duration of the simulation is long enough to ignore the transient behavior at the beginning of the simulation, we do not incorporate a warm-up in our simulation for the original dataset. For the scaled-up version with higher traffic, we use a warm-up period of 2,000 calls before starting to collect results.

Tables 2.2 and 2.3 display dispatch probabilities for the original Hanover County dataset with 5 servers, for different values of reserved servers s_R. System-wide dispatch probabilities f_k^p provided in the table are computed using the dispatch probabilities from the Hypercube

model as

$$f_k^p = \sum_{j \in J} \frac{\lambda_j^p}{\lambda^p} f_{jk}^p, \qquad \forall p \in \{H, L\}, k = 1, \cdots, s. \tag{2.31}$$

The left hand columns of Tables 2.2 and 2.3 report the analytical dispatch probabilities from the approximate Hypercube model, while the right hand columns report the average simulation dispatch probabilities. Table 2.4 reports the server busy probabilities for the original dataset in both the approximate Hypercube model and the simulation model. Dispatch probabilities are all zero for $k > s_C$ for low priority calls, since low priority correction factors q_{k-1}^L from (2.3) are all zero for $k > s_C$, and they are entered into (2.1) to yield dispatch probabilities of zero. This is the direct result of cutoff priority queue discipline. As we lower the cutoff s_C, the probability of dispatching the first preferred server to high priority calls increases.

Tables 2.2 – 2.4 enable us to compare various outputs from the approximate Hypercube model to their corresponding values from the simulation. The tables report that the discrepancy between the analytical Hypercube model results and simulation results are small. The absolute errors for the server busy probability r in Table 2.4 are less than 0.65% across all values of s_R. The dispatch probabilities in first five rows in Table 2.2 and Table 2.3 are also very close to the simulation results with the absolute error of at most 0.64%. Finally, the numbers in the last row of Table 2.2 refer to the loss probabilities, which are estimated with the absolute error of no more than 0.79%. The tables corresponding to Table 2.2 – 2.4 for the Scaled-up Dataset are presented in Appendix A.2.

The computational time for the simulation average 479.60 seconds for the original dataset and 525.49 seconds for the scaled-up dataset. Note that the runtime of the simulation is significantly longer than the runtime of Hypercube model, which was less than 0.11 seconds for the original dataset and 0.50 seconds for the scaled-up dataset, while they both serve the function of evaluating the system given preference lists. We note that some output statistics such as dispatch probabilities to less preferred ambulances can be relatively rare events and the simulation should be run for as long as $100,000$ calls in our example, and possibly longer when there are more ambulances in the system. This highlights a benefit of an analytical

Figure 2.3: Expected Coverage for Different Values of s_R in the Hanover County dataset with $s = 5$, $w = 0$.

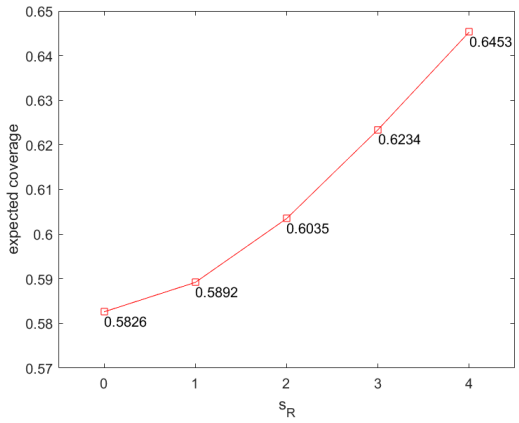
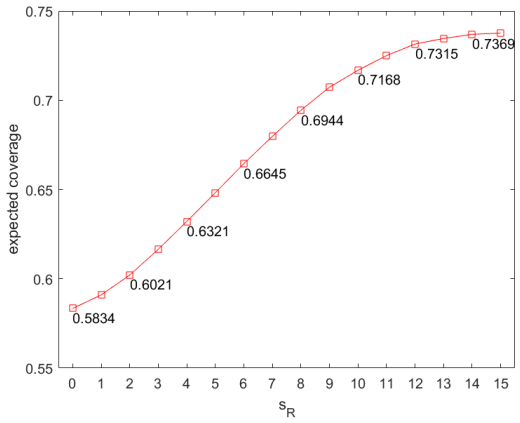


Figure 2.4: Expected Coverage for Different Values of s_R in the Scaled-up Hanover County dataset with $s = 16$, $w = 0$.



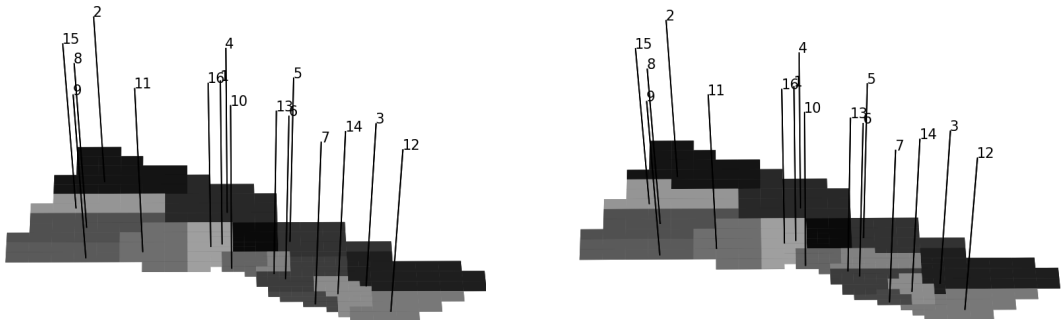
model such as the Hypercube model, whose computational time scales well in the number of servers.

System Design and Coverage Improvement

Here we consider the Loss model and set $w = 0$ so that the MILP objective is to maximize the expected coverage over high priority calls. This allows us to demonstrate the impact of the cutoff on high-priority calls. Figure 2.3 shows the expected coverage over high priority calls from the MILP model with the original dataset, and Figure 2.4 shows the expected coverage over high priority calls for the scaled-up version of Hanover County dataset with 16 servers.

The effect of the reserved capacity on high-priority call coverage is more pronounced in the scaled-up example, as illustrated in Figure 2.4. Specifically, the marginal improvement in expected coverage is 1.63% when the number of reserved servers s_R increases from five to six. Given that the maximum achievable coverage of this system is only 81.28% that occurs under the impractical assumption that every call is served immediately by the server located at the most reachable station, the improvement from 64.82% to 66.45% by reserving one additional server is substantial.

Figure 2.5: First-priority Districts for High Priority Calls with (a) $s_R = 8$ and (b) $s_R = 10$ in the Scaled-up Hanover County Dataset with $s = 16$, $w = 0$.



(a) $s_R = 8$ (b) $s_R = 10$

The expected coverage is monotonically increasing in the number of reserved servers s_R. The expected coverage improves as more servers are reserved based on a combination of two factors. First, as more servers are reserved for high priority call arrivals, servers are less busy (see Table 2.4). Second, as more servers are reserved for high priority call arrivals, the system is more likely to send highly preferred servers that are nearby and is therefore more likely to "cover" the calls. This can be seen in the left half columns of Table 2.2 and 2.3 for the original Hanover County dataset with $s = 5$ servers. The first and second choice servers are more likely to be sent—as shown by dispatch probabilities with $k = 1, 2$ for high priority calls—when s_R increases.

However, the improvement in expected coverage for high priority calls comes at the cost of losing more low priority calls. As seen in Table 2.3, in the base case of reserving no servers, the probability of losing a low priority call is 2.12%, and it increases to 7.31% when reserving one server, while resulting a coverage improvement of 0.66%. Therefore, there is a tradeoff between improving high priority coverage by reserving servers and staying responsive to low priority calls. It is up to decision makers to choose the level of cutoff with allowable loss for low priority calls.

The cutoff priority scheme leads to different design decisions in the MILP model. Figure 2.5 shows the optimal district designs for the scaled-up Hanover County dataset with sixteen servers. Each color block with a station number represents a first priority district covered

by the station. The districts associated with stations 6, 7, 13, and 14 that are the most populated areas in call arrivals, show the greatest changes when cutoff is modified. Therefore, the cutoff leads to different district designs for high priority calls, which contributes to a greater increase in the expected coverage than by solely reserving capacities. This suggests that planning decisions should be altered in response to the introduction of a cutoff, which is an important insight.

Identifying the Number of Reserved Servers

Thus far, we discuss how to improve coverage for high priority calls at the expense of losing more low priority calls by introducing a low priority cutoff level s_C and setting the low priority weight parameter $w = 0$. In this section, we consider $w > 0$ to explicitly consider the effect of losing low priority calls in the objective function. Setting $w = 0$ means we only consider high priority coverage in evaluating the performance of the system, while $w = 1$ implies that high and low priority calls are equally valued. Relatively smaller values such as $0.1 \leq w \leq 0.5$ are more realistic, since emergency calls with higher priorities are more time-sensitive and have a higher necessity to be served in a timely manner (i.e., covered).

Figure 2.6 and 2.7 show the expected weighted coverage for different cutoffs and weights in the original and scaled-up Hanover County dataset. These curves are no longer monotonically increasing in the number of reserved servers as earlier with $w = 0$, since the lost low-priority calls affect the expected weighted coverage here. Hence for each $w > 0$, there exist different values of s_R (with $s_R \leq s$) that maximize total expected coverage, and this analysis can inform decision makers regarding how to select the number of servers to reserve.

Consider the $w = 0.1$ case for the original dataset in Figure 2.6. From Figure 2.6 we can see that the expected coverage reaches its maximal value at $s_R = 3$, and starts to decrease when s_R increases beyond 3. Therefore, we can conclude that reserving $s_R = 3$ servers yields the highest expected weighted coverage and reserving more than three servers is suboptimal. As the weight w increases, the maximal expected weighted coverage for each value of w

Figure 2.6: Expected Coverage with Different Weight on Low Priority Calls in Hanover County Dataset with $s = 5$.

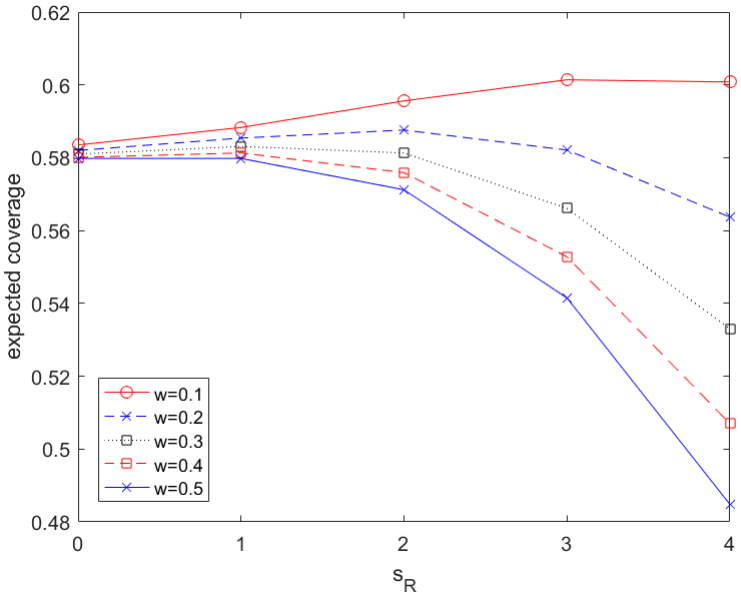


Figure 2.7: Expected Coverage with Different Weight on Low Priority Calls in the Scaled-up Hanover County Dataset with $s = 16$.

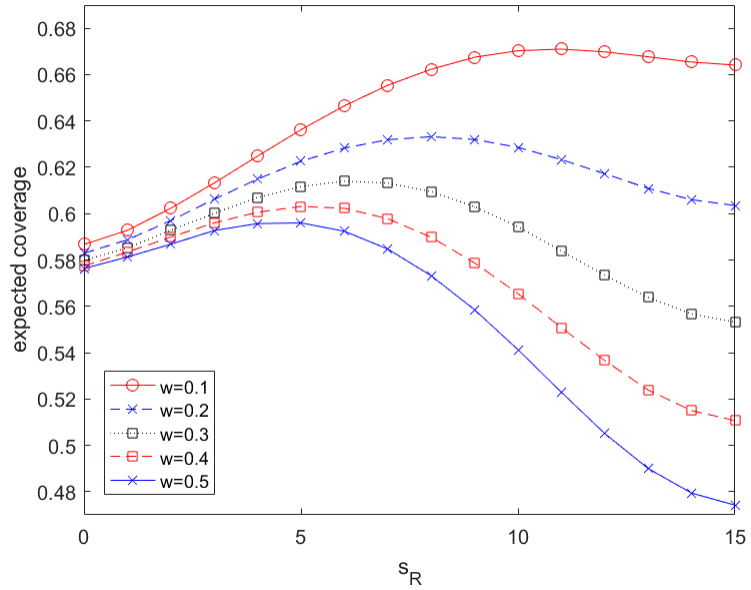


Table 2.5: Dispatch Probabilities for Different Values of s_R With Infinite-capacity Queue and $s = 5$.

	High Priority				Low Priority			
	$s_R = 0$	$s_R = 1$	$s_R = 2$	$s_R = 3$	$s_R = 0$	$s_R = 1$	$s_R = 2$	$s_R = 3$
k=1	0.6621	0.6615	0.6617	0.6553	0.6624	0.6451	0.5610	0.3198
k=2	0.1988	0.2018	0.2126	0.2368	0.1978	0.1838	0.1354	0.0457
k=3	0.0680	0.0719	0.0785	0.0807	0.0683	0.0560	0.0255	0.0000
k=4	0.0269	0.0318	0.0297	0.0174	0.0273	0.0147	0.0000	0.0000
k=5	0.0122	0.0171	0.0086	0.0052	0.0125	0.0000	0.0000	0.0000
P(delayed)	0.0317	0.0160	0.0084	0.0048	0.0317	0.1003	0.2768	0.6354

*Result for $s_R = 4$ is omitted, because if we solve the model with infinite capacity queue using this dataset, the system is unstable.

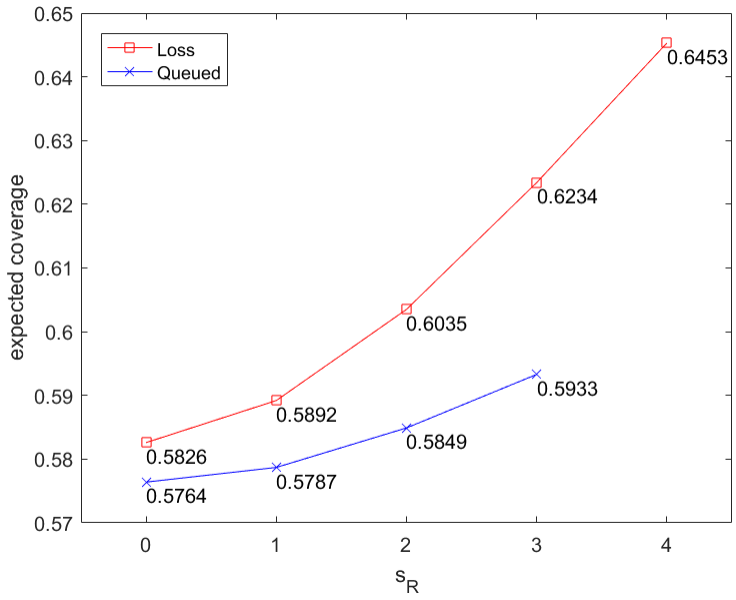
occurs by reserving fewer servers. Specifically when w is as large as 0.5 so that low priority calls are valued half as important as high priority calls in the objective function, expected coverage reaches its maximal values when no servers are reserved ($s_R = 0$). Similarly, for the Scaled-up dataset as seen in Figure 2.7, the value of s_R that yields the highest expected coverage decreases from 11 to 5 as we increase the weight on low priority calls from 0.1 to 0.5. Therefore, a cutoff priority scheme is more effective in improving the expected total coverage when high priority calls are weighed significantly more than low priority calls.

The System with the Infinite Capacity Queue

In this section we report results from the system with infinite-capacity queue, to understand how total expected coverage can be improved by reserving capacity when we maintain a queue and calls are not lost.

Figure 2.8 illustrates the expected coverage for the Queued and the Loss models as a function of s_R. The expected coverage in the Queued model is lower than in the Loss model. With an infinite capacity queue, servers ultimately serve every call, and therefore, the busy probabilities and dispatch probabilities are similar regardless of how many servers are reserved (see Table 2.5), since there is still a similar amount of "work" in the system. As servers become busy serving low priority calls that have been delayed service by the cutoff, there is

Figure 2.8: Expected Weighted Coverage with an Infinite Capacity Queue and $s = 5$, $w = 0$.



*Result for $s_R = 4$ is omitted due to infeasibility.

smaller benefit of improving the total expected coverage from reserving capacity.

By maintaining an infinite capacity queue, low priority calls are never abandoned. The Queued model can become unstable and the low priority call queue can become infinitely long if too many servers are reserved, which occurs for this example when $s_R = 4$ ($s_C = 1$) and the realized service rate for low-priority calls is as low as $P_0\mu$. Compared to the Loss model, the Queued model provides an useful tool for estimating system performances when calls are not abandoned or resorted to external services, as long as the resulting system is stable.

Lastly, a mixed situation where the calls are entered into a queue probabilistically can be considered. This implementation is applicable when external resources such as mutual aid is not always available. It is expected that the performance of a mixed system when the cutoff discipline is adapted to this case lies in between the two curves in Figure 2.8. Intuitively the result would be closer to the Loss model when external resources are available with higher probability.

2.5 Conclusion

This paper introduces models to evaluate the design of public safety systems with a cutoff priority queue, and it provides insights into the benefits that can be achieved from reserving some capacity for high priority calls. We introduce an approximate Hypercube model with cutoff priority queue, and we embed this model into a MILP model so we can identify how to locate and dispatch ambulances to respond to prioritized calls for service.

Our computational analysis shows that the use of iterative MILP-Hypercube procedure is a computationally efficient way to model and solve the problem. The analytical model provides a relatively efficient approximation that requires less computational time compared to discrete event simulation and is also accurate. The computational study demonstrates an improvement in coverage over high priority calls when introducing a cutoff for low priority calls using a real-world dataset. This improvement in coverage for high priority calls is accentuated when we consider a scaled-up dataset with more servers and a higher demand.

There is a tradeoff between improving coverage for high priority calls at the expense of "losing" more low priority calls when a cutoff is introduced. We study this tradeoff by evaluating the expected weighted coverage over both low and high priority calls. The analysis sheds light on how to identify the "right" number of servers to reserve, which can aid decision makers in setting the cutoff.

We also model a system where low priority calls are not lost due to cutoff but are rather served later when there are enough free servers. A system with an infinite capacity queue (Queued model) is implemented and compared to the Loss model. The system with an infinite capacity queue leads to a smaller improvement in coverage when more servers are reserved as compared to the system with a zero capacity queue. This is because low priority calls are delayed rather than lost when the server availability is below the cutoff, and therefore, ambulances are ultimately not "freed up" when there an infinite capacity queue. This suggests that a cutoff has the most potential to improve system performance when cutoff low-priority calls are ultimately serviced by external resources.

The model we provide can be extended. First, our framework can be applied to a system with more than two priority classes by imposing multiple cutoff numbers. This could be practical, since many emergency medical service systems are based on more than two patient priorities. Second, the relationship between the optimal cutoff and the system parameters such as call volume or fleet size can be further studied to understand how to reserve ambulances when there is a sudden increase in demand. The analysis could give insight into the development of emergency plans for mass casualty events and a surge in demand due to severe weather. Third, although 8- or 9-minutes are the most frequently used time thresholds when defining the call coverage, longer thresholds are often used for low priority patients or in rural settings. Our model can be applied to such EMS systems by replacing the reach parameters R_{ij} with priority-specific reach parameters, R_{ij}^{p}, $p \in \{H, L\}$. Finally, we use simulation to demonstrate the accuracy of the models. The framework developed in this paper could be considered a "Solution-Completion by Simulation" (SCS) model, in which the solution obtained by the optimization is applied in a simulation environment to discover more details of the model (Figueira and Almada-Lobo, 2014).

3 DYNAMIC PRIORITY ASSIGNMENT FOR EMERGENCY MEDICAL SERVICES WITH MULTIPLE TYPES OF SERVERS

3.1 Introduction

The problem of dynamically allocating resources to incoming demand is prevalent in service systems where an immediate response is required. Customers can have different service needs, and likewise, servers can vary in specialty or capability of service. Our problem is motivated by the triage process in emergency medical service (EMS) systems, although a number of other potential application areas exist, including telecommunications, logistics, and healthcare. EMS systems have two main goals when sending ambulances to patients: rapidly responding to patients, and responding with the right type of personnel based on the health needs of patients.

The first emergency response goal, the rapid response to emergency calls, is measured by the fraction of calls that is reached within a pre-specified response time threshold (RTT). RTTs play a central role in evaluating performance in nearly all EMS systems, since they are practical for measuring the promptness of service and are also easy to interpret (McLay, 2010). Response times are often critical to patient survival, especially for those with time-critical conditions that require immediate care (Larsen et al., 1993). In EMS systems, responses are often slower when more servers are busy serving patients (Alanis et al., 2013). When more servers are busy, it is likely that the servers located close to the emergency scene are occupied by other calls, resulting in longer response times. Therefore, there is a tradeoff between serving a current emergency patient and being prepared for future calls.

Another goal of EMS systems is a proper matching between patients and ambulances. Tiered EMS systems typically contain various types of medical units that are specialized to perform different jobs. A tiered system sends Advanced Life Support (ALS) vehicles staffed by paramedics, Basic Life Support (BLS) vehicles staffed by emergency medical technicians,

and/or other types of emergency vehicles to calls based on the information provided by the callers to the dispatchers. How to match calls and servers considering their types can be critical to patient outcomes. For example, ALS treatment, as opposed to BLS treatment, to non-traumatic cardiac arrest patients can improve patient survival by nearly 47% (Bakalos et al., 2011). While paramedics require more training and experience, it is argued that they are not necessary for a majority of calls (Kimmel and Persse, 2015). By reducing paramedic usage and only sending ALS units to those in need of ALS interventions, the system can respond to more urgent patients with shorter response times without sacrificing the quality of care for patients with less severe medical needs. However, the actual needs of patients are often not entirely disclosed when dispatch decisions are made. For instance, 911 operators prioritize each call based on a limited set of standard questions and answers provided by the caller (Reilly, 2006).

In this paper, we address these challenges by dynamically assigning servers to patients in real-time using a Markov decision process (MDP) model. The system of concern is non-stationary; the demand for ambulances varies depending on the time of the day and the day of the week (Channouf et al., 2007). This non-stationarity makes it challenging to derive a simple assignment rule that is consistently valid. The MDP model makes assignment decisions depending on imperfect information about patient needs, fluctuating resource levels, and anticipated demand volume. Under this framework, two identical patients who arrive at different times can be served by different types of servers if the system state when they arrive is different.

A large number of papers develop models for ambulance location and relocation problems (Brotcorne et al., 2003; Li et al., 2011; Aringhieri et al., 2017; Reuter-Oppermann et al., 2017). Although these decisions significantly influence EMS performance, deployment decisions are not the focus of this paper. Instead, our MDP model has a coverage-based objective function that inherits ideas from the ambulance location literature.

A cutoff priority queue, which is a simple dynamic resource assignment, is more directly

related to this paper than ambulance location problems. Under the cutoff service discipline, lower priority jobs are immediately served only if the number of busy servers does not exceed a pre-specified threshold. Otherwise, patients are placed in a queue and served later. In other words, when the system is congested, low priority jobs are reprioritized from 'serve immediately' to 'place in a queue.' Taylor and Templeton (1980), Schaack and Larson (1986), and Yoon and Albert (2018b) study EMS systems with a low priority cutoff with an exogenously given cutoff level. In contrast, our analysis provides a framework for adjusting the cutoff in an endogenous manner based on the demand, congestion, and the resource level of the system.

Patient prioritization problems, which are widely studied in the literature using MDP models, provide another framework for dynamically matching ambulances to patients. Argon and Ziya (2009) consider priority assignment decisions for a service system with imperfect information on customer type identities. Argon et al. (2011) provide a review of operations research approaches to improve patient triage and prioritization in the aftermath of Mass Casualty Incidents (MCIs). Jacobson et al. (2012) propose a model that determines patient priorities based on resource limitations and the scale of the event. MCIs triage seeks to manage patients in a long queue when the demand overwhelms supply in systems with high traffic intensity, while our research investigates ambulance assignment in EMS systems with lower traffic intensity. It is typical for such non-MCI settings to assume a loss system, *e.g.*, (Restrepo et al., 2009), or if patients are allowed to queue, that the queue maintains a reasonable length.

There have been previous papers that study the optimal dispatch of emergency vehicles to prioritized patients. McLay (2009) formulates a mixed integer linear program to study the optimal deployment of two types of ambulances to improve patient survivability. McLay and Mayorga (2013b) use an MDP model to explore how to dispatch servers to prioritized customers when there are classification errors in patient priorities. Chong et al. (2016) examine the value of a mixed fleet system by constructing an MDP model and an integer

program of a tiered EMS system with both ALS and BLS servers and high and low priority calls. To the best of our knowledge, our model is the first to solve an ambulance assignment problem in a tiered EMS system where patient information is partially observed.

Operational ambulance planning involves the management of multiple vehicles that cannot be treated independently, which quickly makes the problem intractable when the number of vehicles grows. Therefore, how the state space is defined is a critical issue when describing the operational ambulance planning with an MDP. Some papers provide spatial models that capture the server status and location in detail in the state space. Maxwell et al. (2010) and Rettke et al. (2016) use approximate dynamic programming to keep the problem computationally tractable, while Jagtenberg et al. (2017) derive a heuristic that can handle larger instances. Other papers choose to limit the state space to be compact, by describing the state of the system only with the number of available servers (Alanis et al., 2013; Chong et al., 2016, 2017). Our approach also uses the compact representation of the state space in pursuit of computational tractability. In order to compensate for the loss of explanatory power due to the compact state space, we present a reward function that represents decreasing marginal gains as the system becomes congested. Although the locations of the calls and the servers are not considered in our base model, we provide an approximate spatial model as an extension to consider the geographic information associated with the call.

This paper makes the following contributions:

1. We formulate an MDP model that matches multiple types of ambulances with multiple patient classes. Our model assigns ambulances to patients by considering the resource availability, the uncertainty in patient needs, and non-stationary demands.

2. We present structural properties of the optimal policy. We show conditions under which the optimal policy is a class separable, signal threshold type, and state control limit type policy.

3. We show how our findings regarding the structural properties of the optimal policies

can be extended to three model variations, including an infinite horizon model, a system with patient queues, and an approximate spatial model.

4. Our empirical study suggests that dynamically matching ambulances to patients using the optimal MDP policies significantly improves the system performance in realistic settings. The improvement is more pronounced when the system has a higher demand. With a dynamic resource assignment policy, calls with lower signals are served by BLS servers when the system becomes more congested. When locations of emergency calls are incorporated into the decision framework as call classes in the approximate spatial model, the optimal policy informs how to dynamically update response regions based on the number of available servers.

The remainder of this paper is organized as follows: Section 3.2 defines the notation and formally introduces the MDP modeling framework for the base model. Three variations of the base model are provided in Section 3.3. In Section 3.4, we introduce structural properties of our MDP models regarding the optimal resource assignment policy. The base model and the approximate spatial model are empirically studied with real-world datasets in Section 3.5. The paper is concluded in Section 3.6.

3.2 Dynamic Resource Assignment Model

We consider an EMS system with two types of ambulances—ALS servers and BLS servers—and multiple types of emergency calls (patients) that arrive in a sequential manner. We seek to assign servers to patients given the first-come-first-served non-preemptive service in a way that ensures short response times and matches the right type of ambulance to patients. Whether a call needs an ALS server (*advanced patient*) or not (*basic patient*) is not directly observed. Instead, based on the features that can inform their likelihood of being advanced patients, such as the primary complaint reported prior to dispatch, we group calls into *classes*. Each call class is associated with a signal parameter ω^i, which is the conditional probability

of requiring an ALS service given the call class. Therefore, when an emergency call arrives, its class is observable at dispatch with a signal of the patient's true service needs. We study whether to send an ALS or BLS server to the call with the MDP model. We require an ambulance to respond to every patient if one is available, even if it is not the preferred type. Emergency calls arrive according to a non-homogeneous Poisson distribution, and we assume that once an ALS (BLS) server is dispatched to a call, the service time follows an exponential distribution with rate μ^A (μ^B). The goal of the model is to maximize the total expected reward by matching ambulances to patients, where the reward captures both response times and patient-ambulance matching.

The Poisson arrival assumption of emergency calls is consistent with real-world datasets (Kim and Whitt, 2014; de Souza et al., 2015). Assuming exponential service time may deviate from what is observed in reality, but several papers validate the use of exponential service time through simulations and numerical experiments, showing that the performance of their models do not depend on the choice of service time distribution (Ansari et al., 2017; Jagtenberg et al., 2017). Lastly, we use a single-valued service rate for all servers of the same type that does not depend on the locations of emergency calls and servers. This is an assumption frequently made in the EMS literature including (Argon and Ziya, 2009; Jacobson et al., 2012; Chong et al., 2016).

A key simplification we make is that only the numbers of available ALS and BLS servers define the state. We intentionally keep our state space compact to avoid the curse of dimensionality and to derive structural properties of the optimal policies. As a result, we do not consider individual server locations when we describe the system state, and arriving calls are represented by their classes, not their locations. We lift this assumption later in Section 3.3 by including call locations into the state space representation. Therefore, the benefit of serving a call is measured based on the number of available servers and the class of the call, without considering their geographic locations. The core decision we make throughout the model is matching servers to calls that arrive sequentially. We note that the number of

available servers can be a good proxy for the probability of responding to an arbitrary call in a given RTT, especially with an appropriate redeployment policy, such as a compliance table (Alanis et al., 2013; Sudtachat et al., 2016).

Consider the 'boundary' cases where all servers of at least one type are busy. First, if all ALS (BLS) servers are busy and there is at least one BLS (ALS) server idle, arriving calls are forced to be served by a BLS (ALS) server regardless of its class. Second, if all ALS and BLS servers are unavailable and a call arrives, that call is either considered to be lost (*Loss system*) or put in one of two patient queues (*Queue system*). In the Loss system, lost calls are assumed to be served by external resources, such as neighboring EMS departments through a process called mutual aid. In the Queue system, two queues are maintained separately, one for calls to be served by ALS servers and another for calls to be served by BLS servers. Calls in an ALS (BLS) queue are served later on a first-come-first-served (FCFS) basis when ALS (BLS) servers become free. Throughout the paper, we primarily focus on properties of the Loss system, which is more common in the EMS literature and also consistent with many real-world operations. We provide the model description and structural properties of the Queue system in Subsection 3.3 and Subsection 3.4.

MDP Model Description

In this Subsection, we formally define our MDP model. A summary of the indices and notation used in the MDP model is presented in Table 3.1.

Time. The objective is to maximize the expected total reward by serving emergency calls over a given time period, for instance, a day or a week. For that, we assume a finite-horizon model without a time discount. We assume that the salvage value at the end of the time horizon is zero. In order to uniformize a continuous-time MDP into a discrete-time MDP, uniformization factor $\Lambda = \max_t \sum_i \lambda_t^i + N^A\mu^A + N^B\mu^B$ is used that defines the maximum rate of transition (Puterman, 1994). This uniformization is available when the value of Λ is finite. We note that since our model is a finite-horizon MDP, this uniformization is an approximation

Table 3.1: Summary of Notation

Symbol	Description
$N^A(N^B)$	Number of ALS (BLS) servers in the system
m	Number of call classes, $i \in \{1, \cdots, m\}$
T	Number of time epochs, $t \in \{1, \cdots, T\}$.
$p \in \{a, b\}$	Type of services calls need: a refers to advanced patient meaning that the call needs ALS service and b refers to basic patient meaning that the call needs basic support and can be served by either an ALS or BLS server
$q \in \{A, B\}$	Server types: A refers to ALS and B refers to BLS
λ_t^i	Arrival rate of class i calls at time t
$\mu^A(\mu^B)$	Service rate of a ALS (BLS) server
ω^i	Signal. Probability that class i calls need service from an ALS server
$s_t = (s^A, s^B)$	System state at time t
$d_t = (d^1, \cdots, d^m)$	Action (decision) at time t
$P_t(j\|s, d)$	Transition probability from state s to state j at time t given action d
$R_t(s, d)$	The expected reward at state s and time t given action d
U_{pq}	The expected utility gain of serving a type p call with a type q server
$f^p(s)$	Coverage function. Probability that a call in need of type p service is successfully covered in time threshold when there are s servers available
$V_t(s)$	Cost-to-go function (value function) at state s and time t

for which the approximation gap decreases to zero as the number of time epoch increases to infinity (Miller, 1968). We use a finite horizon to represent the non-stationary nature of the problem. When the parameters are stationary, the problem can be easily converted to infinite horizon MDP with long-run average reward criteria. In this case, the uniformization is exact, and it can be easily shown that all the structural properties presented in the paper and the managerial insights still hold.

States. We define the state of the system as $s_t = (s^A, s^B)$ for $s^A \in \{0, 1, ..., N^A\}$, $s_t^B \in \{0, 1, ..., N^B\}$, where we define s_t^A as the number of ALS servers available at time epoch t, and s_t^B as the number of BLS servers available at time epoch t. The state describes the ambulance availability of the system. The cardinality of the state space is $|(N^A+1)(N^B+1)|$.

Table 3.2: Coverage function

	Advanced patient (a)	Basic patient (b)
send ALS (A)	$f^a(s^A)$	$f^b(s^A)$
send BLS (B)	$f^a(s^B)$	$f^b(s^B)$

Table 3.3: Utility of serving a call

	Advanced patient (a)	Basic patient (b)
send ALS (A)	U_{aA}	U_{bA}
send BLS (B)	U_{aB}	U_{bB}

Actions. Actions are defined as $d_t = (d_t^1, \cdots, d_t^m)$, $d_t^i \in \{A, B\}$ for $i \in \{1, \cdots, m\}$, with

$$d_t^i = \begin{cases} A & \text{Dispatch an ALS server if a class } i \text{ call arrives at time epoch } t \\ B & \text{Dispatch a BLS server if a class } i \text{ call arrives at time epoch } t. \\ 0 & \text{Do not dispatch any server. (available only when all servers are busy)} \end{cases}$$

As previously mentioned, in the boundary cases where there are no available ALS (BLS) servers, we add a rule that we must dispatch a BLS (ALS) server. An arriving call is lost (or queued) only when all servers of both types are busy ($s^A = s^B = 0$).

Transition Probabilities. Exactly one of the following six transition events occur between two consecutive time epochs: (1) a call arrives and an ALS server is dispatched, (2) a call arrives and a BLS server is dispatched, (3) an ALS server finishes service, (4) a BLS server finishes service, (5) neither a call arrives nor a server completes service, or (6) a call arrives and is lost when all servers are busy.

Rewards. Recall that the objective is to maximize the expected number of covered calls weighed according to the matching between the patient and server assigned. Specifically, the reward is a function of (1) the signal, (2) the coverage function, and (3) the utility of serving a call given the signal and the action.

First, the signal represents the probability of a class i call requiring an ALS service and is endogenously given as ω^i. Next, the coverage function represents the probability that an ambulance can reach the scene in a pre-defined RTT (*e.g.*, 9 minutes) given the patient needs

and the action. Ambulances are spatially located, and therefore, only a subset of ambulances can respond to a specific location in an RTT. The probability that an arriving call is served in RTT is modeled as non-decreasing coverage functions f^a and f^b that are static over time, as defined in Table 3.2. Finally, the utility parameter reflects different payoffs depending on how the server type (ALS, BLS) matches to the actual needs of the call (advanced, basic). As seen in Table 3.3, the first subscript represents the true need of a call (a for advanced and b for basic), and the second subscript describes the type of server that serves the call (A for ALS and B for BLS).

The overall reward function is therefore

$$\begin{aligned} R_t(s^A, s^B, d) = &\frac{1}{\Lambda}\sum_i \lambda_t^i \mathbb{1}_{d^i=A}(\omega^i f^a(s^A)U_{aA} + (1-\omega^i)f^b(s^A)U_{bA}) \\ &+ \frac{1}{\Lambda}\sum_i \lambda_t^i \mathbb{1}_{d^i=B}(\omega^i f^a(s^B)U_{aB} + (1-\omega^i)f^b(s^B)U_{bB}) \end{aligned}$$

The reward is computed as a weighted sum of the product of coverage and utility, where the weight is determined by the signal. We multiply coverage and utility since service quality depends on both the response time and the type of service provided.

We define the coverage function $f^a(s)$ ($f^b(s)$) be the fraction of advanced patients (basic patients) covered in a given RTT when s ambulances are available. In practice, the fraction of calls that is covered also depends on the locations of calls. We can evaluate a valid upper bound of $f^a(s)$ and $f^b(s)$ by solving an instance of the Maximal Covering Location Problem (MCLP) with s servers (Church and ReVelle, 1974). Maxwell et al. (2014) prove that the solution from MCLP instances is a valid upper bound of the coverage function. If a redeployment policy based on the number of available servers is used, then the value of $f^a(s)$ and $f^b(s)$ can be computed directly from the objective value of the redeployment problem. A compliance table is an example of a redeployment policy that is commonly used in practice (Alanis et al., 2013). In this case, objective function values of MCLP instances can serve as estimates of the coverage functions instead of their upper bounds. This simplified coverage

function is an essential part of our modeling to prevent the curse of dimensionality due to the exponentially large state space. We also provide an approximate spatial model with this simplifying assumption lifted later in Section 3.3, where the coverage function is redefined as a function of the location of an emergency call as well as the number of available servers. As $f^a(s)$ and $f^b(s)$ are non-decreasing in s, the reward function decreases as more servers are busy. This incentivizes the model to reserve advanced servers to avoid visiting states with fewer advanced servers available, since those states produce lower rewards.

As far as the utility values are concerned, we assume $U_{aA} > U_{aB}$ to penalize the under-service of advanced patients. We do not assume relative orderings between any other pair of utility values. In particular, we do not explicitly penalize over-service of basic patient calls. Instead, the model discourages over-service through higher transition rates into lower states (*i.e.*, states with fewer advanced servers available). We additionally remark that other papers in the literature such as Chong et al. (2017) typically assume that $U_{bA} = U_{bB}$, since in practice a low priority call can be equally well served by an ALS server or a BLS server, and that $U_{aA} > \max\{U_{bA}, U_{bB}\}$, indicating that providing an ALS service to an advanced patient adds more value to the system than serving a basic patient that does not require advanced service. The utility values can be estimated using data: for instance, as far as non-trauma cardiac arrest patients are concerned, we can set $U_{aA} = 1.0$ and $U_{aB} = 0.68$ using the result of Bakalos et al. (2011).

Optimality Equations. The problem is discretized and has a finite horizon, and thus, we can solve the problem using backward induction. We maximize the total expected reward over a finite horizon without a time discount, and as the action space $D_{s_t} = \{A, B\}^m$ ($D_{(0,0)} = \{0\}$) is finite, the optimality equations (Bellman equation) are:

$$V_t(s^A, s^B) = \max_{d \in D_{s_t}} \left\{ R_t(s^A, s^B, d) + \sum_{(j^A, j^B) \in S} P_t(j^A, j^B | s^A, s^B, d) V_{t+1}(j^A, j^B) \right\},$$

for all time epoch $t \in \{1, ..., T\}$.

3.3 Model Variations

Thus far, we have investigated a finite horizon MDP in which the state space represents an Erlang Loss system, which is a commonly made assumption regarding EMS systems (Restrepo et al., 2009). Subsection 3.3 discusses how the base model presented in Section 3.2 can be extended to a Queue system. Subsection 3.3 explains how the base MDP model can be reformulated as an infinite horizon, average reward MDP model with infinite length patient queues. Lastly, Subsection 3.3 provides an approximate spatial model that considers the geographic locations of emergency calls.

Finite Horizon Model with Patient Queues

The first model variation is a Queue system with two infinite FCFS queues—an ALS queue and a BLS queue. Unlike the base model, emergency calls that arrive when all of the servers are busy are not abandoned. Instead, the model determines which queue the call enters: patients in the ALS (BLS) queue are served by ALS (BLS) servers when servers become available. Once a call enters a queue, it does not renege or relocate to the other queue. When all servers of one type are busy and at least one server of the other type is idle, a server is dispatched to the call regardless of the type, as in the Loss system. We make the following adjustments to the model defined in Section 3.2 for the Queue system.

- The domain for the each element in the state is extended to all integer values less than or equal to N^q ($s^q \in \{-T + N^q, ..., N^q - 1, N^q\}$) for $q \in \{A, B\}$, where negative values of s^q represent the number of calls waiting in the type q queue.

- Following the above, the domain for the coverage function $f^p(s)$ input s is extended to $s^q \in \{-T + N^q, ..., N^q - 1, N^q\}$ for $q = \{A, B\}$.

Note that the queue length is bounded above by $T - N^q$. Since we are solving the problem over a finite time horizon where at most one arrival events arises within a time epoch, the

maximum number of patient arrivals is bounded above by T. Therefore, the queue length $T - N^q$ is not infinite and allows all patients to either be served or queue. When the length of time horizon is exceptionally long, it is practical to impose smaller queue lengths such that the probability of patients balking due to full queues is sufficiently small.

Infinite Horizon Model Variation

The base MDP model can consider an infinite horizon, undiscounted, average reward MDP model when assuming a stationary arrival rate and making several modifications to the model. In an infinite horizon model, there is no transient behavior. We do not need to assign the initial server availability and end-of-time-horizon salvage values as in the finite horizon model. Moreover, the uniformization procedure to convert the continuous-time MDP into a discrete-time MDP, which is approximate for the base finite horizon model, is exact for the infinite horizon model. While the finite horizon model is useful in representing the non-stationary behavior of the system, the infinite horizon model helps in understanding the characteristics of the system when inputs are relatively stationary over time. We assume a Loss system as in the base model. However, it is straightforward to modify the model to have both of infinite horizon and patient queues.

Since the system cannot abandon patients unless all ambulances of both types are busy, the Markov Chain is positive recurrent for the Loss system under any feasible MDP policy. The MDP state space is finite and rewards are bounded. Therefore, the value iteration algorithm converges to an optimal solution (Puterman, 1994).

Approximate Spatial Model Variation

We present an approximate spatial model that determines which call locations to include in ALS (BLS) *response districts*—regions served by ALS (BLS) servers—and updates response districts when resource availability changes. The base model introduced in Section 3.2 assigns servers to calls based on their signal, which is their likelihood of being an advanced patient.

In the approximate spatial model, we embed spatial information in call classes in addition to the likelihood of being an advanced patient by considering the geographic location of the call in a similar manner as in Alanis et al. (2013). Let $j \in \{1, ...n\}$ denote the location of the call. A call class is represented by a tuple (i, j), where i represents the primary complaint reported at dispatch that gives imperfect information about the needs of the patients as in the previous sections, and j represents the location of the call. Although the approximate spatial model does not explicitly take into account the location of ambulances, the model provides insight regarding how to design response districts.

The base model uses coverage functions to represent the probability that a call is reached within an RTT. Since we include call locations in the approximate spatial model, the coverage functions are also updated to consider call locations. Instead of MCLP from Church and ReVelle (1974) as in the base model, we solve the Maximal Covering Location Problem with the Probabilistic Response Times (MCLP+PR) from Daskin (1987) and Erkut et al. (2009) to determine server locations when there are $s = \max\{N^A, N^B\}$ servers.

In the approximate spatial model, we do not assume a deployment policy that relocates servers whenever the number of idle servers changes. Instead, we assume that servers are located according to the optimal solution of MCLP+PR without redeployment and that all servers are equally likely to be busy. We consider all possible configurations of server status given the number of available servers $s \leq \max\{N^A, N^B\}$ to calculate $f^p(s, j)$, the probability that a server responds to a call in need of type p service at location j in a timely manner. Under this new definition, coverage functions are still monotone nondecreasing and concave. The modified reward functions are:

$$\begin{aligned} R_t(s^A, s^B, d) = &\sum_{i=1}^{m}\sum_{j=1}^{m} \frac{1}{\Lambda}\lambda_t^{ij} \mathbb{1}_{d^i=A}(\omega^i f^a(s^A, j)U_{aA} + (1-\omega^i) f^b(s^A, j)U_{bA}) \\ &+ \frac{1}{\Lambda}\sum_{i=1}^{m}\sum_{j=1}^{m} \lambda_t^{ij} \mathbb{1}_{d^i=B}(\omega^i f^a(s^B, j)U_{aB} + (1-\omega^i) f^b(s^B, j)U_{bB}) \end{aligned}$$

3.4 Structural Properties of the Optimal Policies

Structural Properties of the Optimal Policies for the Base Model

We describe a set of structural properties for the base MDP model presented in Section 3.2. We show conditions on the parameter values under which the following holds:

1. There is an optimal policy that is class separable, *i.e.*, assigns a server to a call regardless of the optimal assignment decisions for other classes.

2. There is an optimal policy that is more likely to send an ALS server to a call with a higher signal.

3. There is an optimal policy that is more likely to send an ALS server to a call when more ALS servers are available.

The findings in this section are valuable in two ways. First, the structural results allow us to substantially reduce the number of actions we need to evaluate to find the optimal policy. Second, under some restrictions on the parameter values, the optimal policy can be characterized as a signal threshold type policy and as a control limit policy. These results provide insights for finding relatively simple heuristics to solve ambulance-patient matching problems. The following theorems provide more detail regarding these results. All proofs can be found in Appendix B.1 – B.3.

Theorem 3.1 (Class Separability of the Optimal Policies)**.** *For any time epoch t and state s, the optimal action for class i does not depend on the actions for other class $i' \neq i$.*

Theorem 3.1 states that when we determine the optimal action for class i, the actions for other classes do not affect the decision. The number of call arrivals in a time epoch is at most one due to the uniformization, so the class separability result is intuitive. Class separability allows us to identify an optimal policy efficiently, since we can solve the value functions by comparing at most two actions for each call class (which involves comparing two actions m times each time epoch) instead of all possible combinations (which involves comparing 2^m

actions each time epoch).

Theorem 3.2 (The Optimality of Threshold Type Policies)**.** *For any time epoch t and state $s = (s^A, s^B)$, a threshold signal value $\bar{\omega}_t(s)$ can be specified such that it is optimal to send an ALS server to a class i call ($d_t^{i*}(s^A, s^B) = A$) if and only if $\omega^i > \bar{\omega}_t(s)$, if the following inequality is satisfied:*

$$U_{aA} f^a(s^A) - U_{bA} f^b(s^A) - U_{aB} f^a(s^B) + U_{bB} f^b(s^B) > 0. \tag{3.1}$$

The threshold value is defined as

$$\bar{\omega}_t(s) = \frac{U_{bB} f^b(s^B) - U_{bA} f^b(s^A) + V_{t+1}(s^A, s^B - 1) - V_{t+1}(s^A - 1, s^B)}{U_{aA} f^a(s^A) - U_{bA} f^b(s^A) - U_{aB} f^a(s^B) + U_{bB} f^b(s^B)}. \tag{3.2}$$

Theorem 3.2 further simplifies the search for an optimal policy, since it allows us to find a single-valued threshold $\bar{\omega}_t(s)$ for a given state $s = (s^A, s^B)$ at time epoch t, and use that threshold to determine actions for all call classes without any further evaluation on value functions. Unlike Theorem 3.1 that is unconditionally true, Theorem 3.2 holds under a specific choice of states and parameters that satisfy condition (3.1).

Condition (3.1) can be interpreted as that the marginal utility of serving an advanced patient with an ALS server, as opposed to a BLS server, is higher than the marginal utility of serving a basic patient with an ALS server instead of a BLS server. Condition (3.1) is more likely to be satisfied (and therefore the optimal policy is of threshold-type) if there are more ALS servers available (compared to BLS servers) and if U_{aA} is significantly larger than U_{aB}, *i.e.*, under-serving a call in need of ALS service is severely penalized.

Theorem 3.3 (Optimality of Control Limit Type Policies)**.** *For any time epoch t, call class i and the number of available BLS servers s^B, if $\mu^A = \mu^B$ and $f^a(s)$ is concave in s, then a*

control limit $\bar{s}_t(i, s^B)$ *can be specified such that*

$$d^{i^*}(s^A, s^B) = \begin{cases} A & s^A > \bar{s}_t(i, s^B) \\ B & s^A \leq \bar{s}_t(i, s^B) \\ 0 & s^A = s^B = 0 \end{cases}$$

Therefore, it is optimal to send an ALS server to a class i *call if and only if there are more than* $\bar{s}_t(i, s^B)$ *ALS servers available.*

Also, for any time epoch t*, call class* i *and the number of available ALS servers* s^A*, if* $\mu^A = \mu^B$ *and* $f^b(s)$ *is concave in* s*, then a control limit* $\bar{s}_t(i, s^A)$ *can be specified such that it is optimal to send a BLS server to class* i *call if and only if there are more than* $\bar{s}_t(i, s^A)$ *BLS servers available. These control limits construct a monotone switching curve which divides the state space into two connected regions, one in which sending an ALS server is optimal, and another in which sending a BLS server is optimal.*

Theorem 3.3 provides another characterization of the optimal policy in terms of the number of available servers. Our choice of the coverage functions $f^a(s)$ and $f^b(s)$, which are the objective function values of MCLP instances, satisfy the concavity conditions in Theorem 3.3. Since having more ambulances available has decreasing marginal gains in coverage, other choices of coverage functions also satisfy concavity conditions.

Structural Properties of the Optimal Policies for the Model Variations

The following two theorems state that the structural properties for the base model also hold for the Queue model and the infinite horizon model introduced in Section 3.3 under slight modifications. We provide sketches of the proofs for Theorems 3.4 – 3.5 in Appendix B.4.

Theorem 3.4 (Structural properties for the finite horizon MDP with Patient queues)**.** *Theorems 3.1, 3.2, and 3.3 are valid for the finite horizon MDP with patient queues.*

Theorem 3.5 (Structural properties for the infinite horizon, Loss system MDP)**.** *Theorems 3.1, 3.2, and 3.3 are valid for the infinite horizon MDP with a zero length queue.*

Finally, for the approximate spatial model, we note that the reward function remains separable by call class (i, j) where i represents the primary complaint and j represents the call location as in Subsection 3.3. Therefore, Theorem 3.1 also holds for the approximate spatial model. While Theorem 3.1 and Theorem 3.3 can be applied without any modification, Theorem 3.2 is true for the approximate spatial model if instead of applying the result directly to call class (i, j), we fix the call location j and compare signals between different complaints i and i'.

3.5 Computational Results

Setup

We carry out numerical experiments using the emergency call logs from Hanover County, Virginia from 2009 – 2011. From the call logs, we estimate non-stationary call arrival rates for each hour and call class, system-wide service rates, and the signal for each call class. The dynamic resource assignment problem is solved for a time horizon of 24 hours, which consists of 828 time epochs after uniformization based on the average weekday call arrival rates, for the system with 24 servers including $N^A = 12$ ALS and $N^B = 12$ BLS servers.

We divide calls into $m = 15$ classes and estimated signal value ω^i for each call class based on the call logs. The details of the class generation and the signal estimation are provided in Appendix B.5. The estimated signal value, which represents the fraction of advanced patients among all calls in each class, is presented in Figure 3.1 in decreasing order of ω^i.

Figure 3.1: Signal parameter ω^i for call class $i = 1, ..., m$

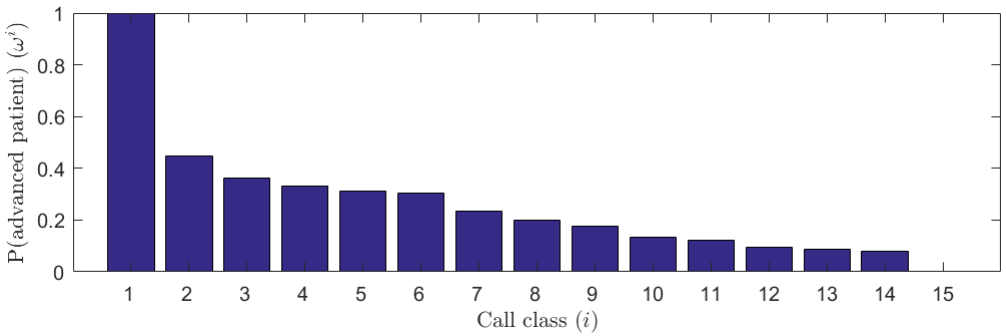
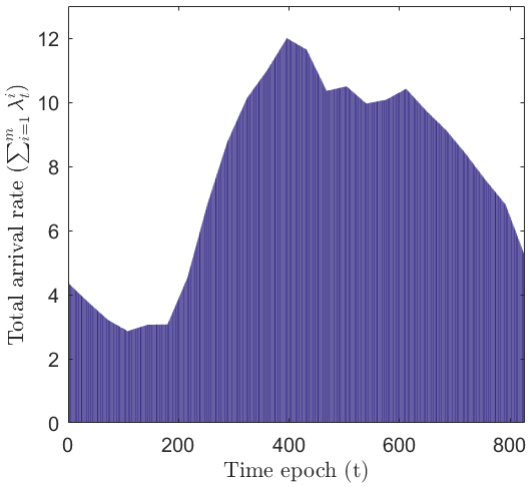


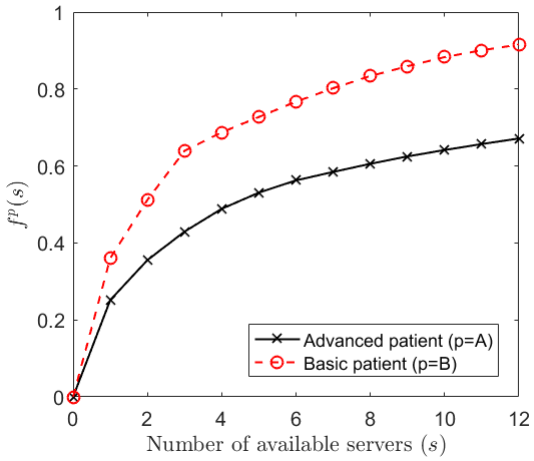
Figure 3.2: Non-stationary call arrival rates



We scale up arrival rates and service rates to demonstrate that our model is capable of solving the resource allocation problem for large-scale EMS systems. First, the service rate is normalized to 1 call per hour for both ALS and BLS servers. Then we scale up call arrival rates to keep the maximum utilization equal to 0.5. Here the utilization is defined as the ratio between total call arrival rates and total service rates ($\rho_t = \frac{\sum_i \lambda_t^i}{\mu^A N^A + \mu^B N^B}$). Since the call arrival rate is non-stationary, the maximum utilization is defined as the utilization at the time point with the highest arrival rate ($\rho = \max_t \rho_t$). The total call arrival rate is approximated by a continuous piecewise linear function as illustrated in Figure 3.2 with maximum utilization = 0.5. When testing the model performance, we conduct a sensitivity analysis on various values of maximum utilization ranging from 0.2 to 0.8. When we change

the utilization, we also update uniformization rate Λ to keep the assumption that at most one call arrival or service completion event happens in a time epoch.

Figure 3.3: Coverage functions $f^p(s)$ defined for $p = \{a, b\}$ and number of available servers $s = 1, ..., \max\{N^A, N^B\}$



We estimate coverage functions by solving the MCLP instances for all possible values of the number of available servers and are illustrated in Figure 3.3 based on RTTs. We use distance thresholds, instead of time thresholds, since the distance between the station and the call location is a close proxy of the response time. We assume that advanced patients are considered to be covered if there is at least one server located within four miles, while basic patients have a six-mile threshold. This makes it easier to cover a call that does not need ALS service, given the same number of available servers, which is why $f^b(s)$ is larger than $f^a(s)$ for all values of s. Figure 3.3 does not specify the server type in its x-axis, since we do not differentiate the service speed of ALS servers and BLS servers.

The utility values are set to $U_{aA} = 1.0$, $U_{aB} = U_{bA} = U_{bB} = 0.5$, assuming that dispatching an ALS server to an advanced patient produces twice as much value as under-serving such a patient or serving a basic patient. The final policy generated from backward induction can be interpreted as a series of resource assignment rules for each time epoch $t \in \{1, ..., T\}$ and call class $i \in \{1, ..., m\}$, given the number of idle ALS and BLS servers.

We relate our choice of parameters to structural properties introduced in Section 3.4.

First, we always have a class separable optimal policy by Theorem 3.1 independent of the choice of parameters. Second, utility values and coverage functions satisfy equation (3.1) for any value of s^A and s^B. Therefore, by Theorem 3.2, there exists an optimal threshold type policy. Finally, both coverage functions are monotone increasing and concave, and ALS servers and BLS servers have the same service rate. By Theorem 3.3 there exists an optimal control limit type policy. Therefore, we know that there exists a class separable, signal threshold type, and state control limit type optimal MDP policy under our choice of parameters, and we can restrict our action space to the set of actions that satisfy all three properties.

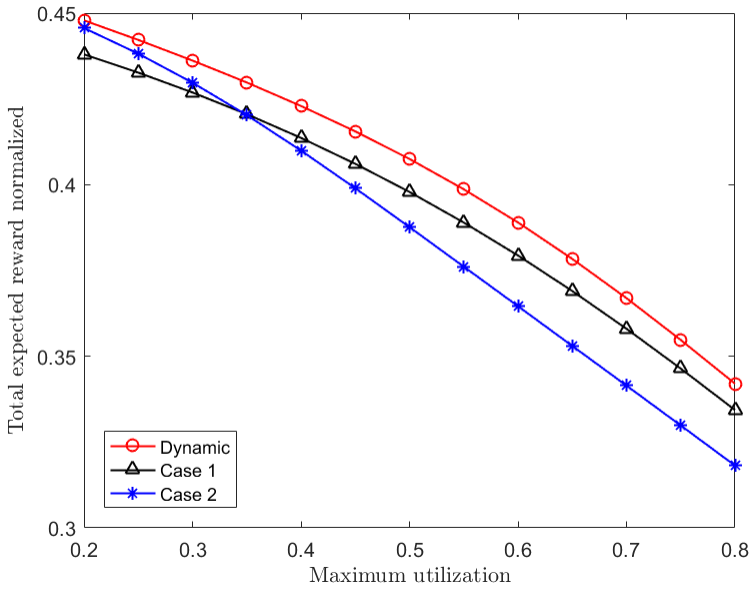
Data preprocessing and the implementation of the MDP model was performed in Python 2.7.13, and MCLP instances were solved using Gurobi 7.0.2. After reducing the optimal action space by applying structural results from Section 3.4, the entire runtime to solve the MCLP instances and MDP model together is less than 30 CPU seconds.

Reward Improvement using the Dynamic Policy

We define the total expected reward by the value of the MDP at time epoch 0 when all ALS and BLS servers are initially idle. The result of the optimal MDP policy—the dynamic policy—is compared to two other benchmark static policies presented by McLay and Mayorga (2013b). Under the *Case 1* policy, all classes except for class 1 are served by BLS servers. Under the *Case 2* policy, all classes except for class 15 are served by ALS servers. Both policies follow the same rule as the dynamic policy for boundary cases, *i.e.*, an arriving call is lost only when both types of servers are all busy ($s^A = s^B = 0$).

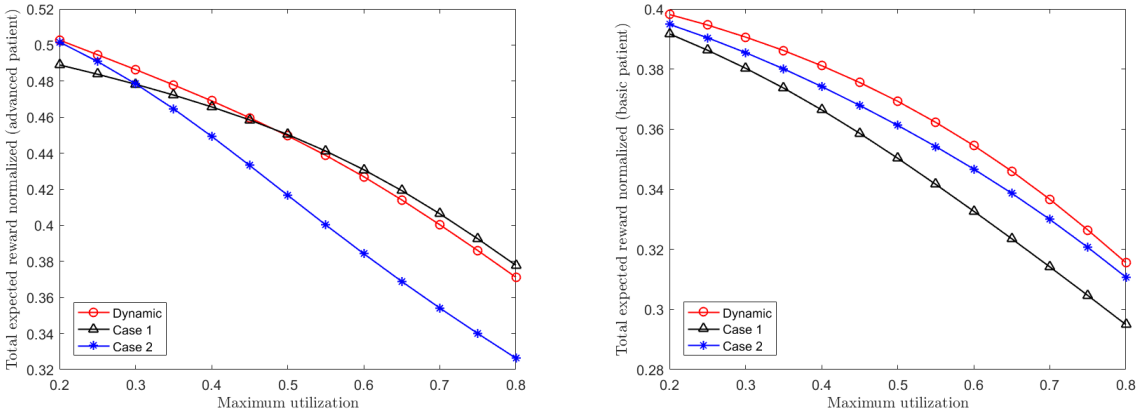
In this subsection, we compare the total expected reward from the base model's dynamic policy and benchmark static policies under various values of the utilization, which is illustrated in Figure 3.4. Since systems with different utilization factors have a different number of expected call arrivals, the total expected reward is normalized by the total number of calls. A higher utilization represents a system with more congestion compared to the resource

Figure 3.4: Total expected reward (normalized)



available. All three curves monotonically decrease when the utilization increases. The dynamic policy outperforms the two benchmark static policies for all values of the utilization. The increase in total expected reward by using the dynamic resource assignment policy is especially pronounced when the system is more congested, particularly, compared to the Case 2 policy.

Figure 3.5: Total expected reward (normalized) for (a) advanced patients and (b) basic patients



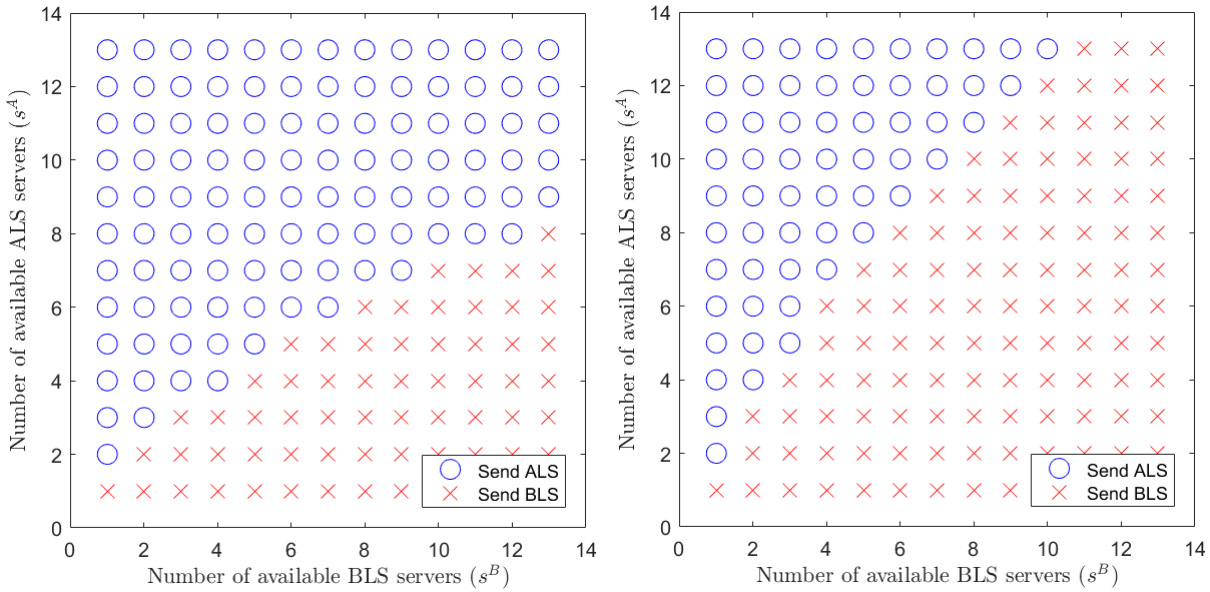
(a) Advanced patients

(b) Basic patients

An arriving patient is either an advanced or a basic patient. We recursively compute the total expected reward in each patient group based on the Bernoulli distribution with parameter ω^i for a patient in class i. Figure 3.5 shows the total expected reward accrued for advanced patients and basic patients separately. The dynamic policy refers to the optimal policy, the same dynamic policy illustrated in Figure 3.4. The *Case 1* and *Case 2* policies are defined likewise. By examining the total expected reward over each patient group separately, we obtain insight into how the optimal policy improves overall system performance. Figure 3.5a shows that for those in need of ALS service, the dynamic policy yields higher rewards than the Case 2 policy, and the difference is more pronounced when the utilization is higher. Meanwhile, the Case 1 policy yields higher rewards for advanced patients, since it rations ALS servers for class 1 patients. For basic patients, the trend is different. In Figure 3.5b, we observe that the dynamic policy consistently yields higher rewards than both benchmark static policies, and the gap is more significant when the utilization increases. The Case 1 policy yields the lowest rewards for basic patients. Hence, given the current problem instance, the improvement in the overall performance as shown in Figure 3.4 mainly comes from the improved service of basic patients.

Next, we discuss the optimal actions. Figure 3.6 depicts the optimal actions across all states at $t = \frac{T}{2} = 389$ for call class 2 (Figure 3.6a) and class 12 (Figure 3.6b). We choose $t = \frac{T}{2}$ to minimize the effect of transient behavior regarding warming up at the beginning of time horizon and salvaging at the end of time horizon. As previously proven in Theorems 3.2 and 3.3, a threshold, control limit type policy is optimal. For example, Figure 3.6a shows that when there are 6 available BLS servers in the system, the optimal control limit for the number of available ALS servers for a class 2 call is 6, *i.e.*, a class 2 call arriving at $t = \frac{T}{2}$ is served by an ALS server if and only if there are 6 or more ALS servers available in the system. It is also observable that the optimal policy is a threshold type by comparing Figure 3.6a and 3.6b. The signal for call class 2 is 0.4479, and the signal for call class 12 is 0.0955. If we pick $(s^A, s^B) = (6, 6)$, for instance, then the optimal action for a class 2 patient is to

Figure 3.6: Optimal actions at $t = \frac{T}{2}$ and (a) class 2 and (b) class 12 for different states



(a) Call class 2

(b) Call class 12

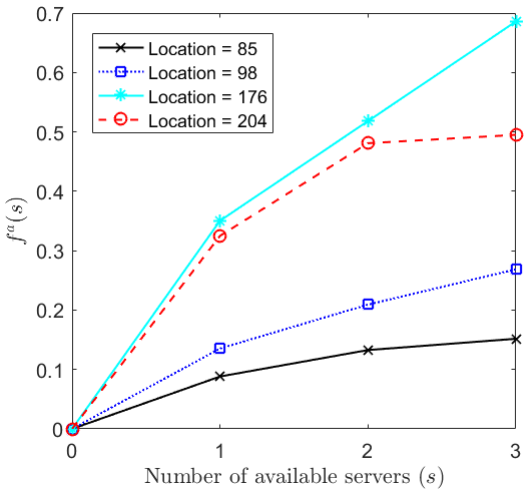
send an ALS server while the optimal action for a class 12 patient is to send a BLS server.

Computational Results for Approximate Spatial Model

We demonstrate through an empirical study that the base model can be extended to the approximate spatial model to inform how to design geographic response districts. Although our model does not describe server locations, the approximate spatial model can inform which call locations to response with ALS servers, and how to update that decision when the number of available servers changes.

We use the Hanover County dataset described in Subsection 3.5 to empirically demonstrate the approximate spatial model in Section 3.3. The system is equipped with $N^A = 3$ ALS servers and $N^B = 3$ BLS servers. The service rate is normalized to 1 call per hour and the maximum utilization is set to 0.5. The geographic region is divided and aggregated into 270 demand nodes consisting of 139 2×2 mile cells (districts in which demand is relatively low) and 131 1×1 mile cells (districts in which demand is relatively high). Each demand node

Figure 3.7: Advanced patient coverage functions for the approximate spatial model for call nodes 85, 98, 176, and 204

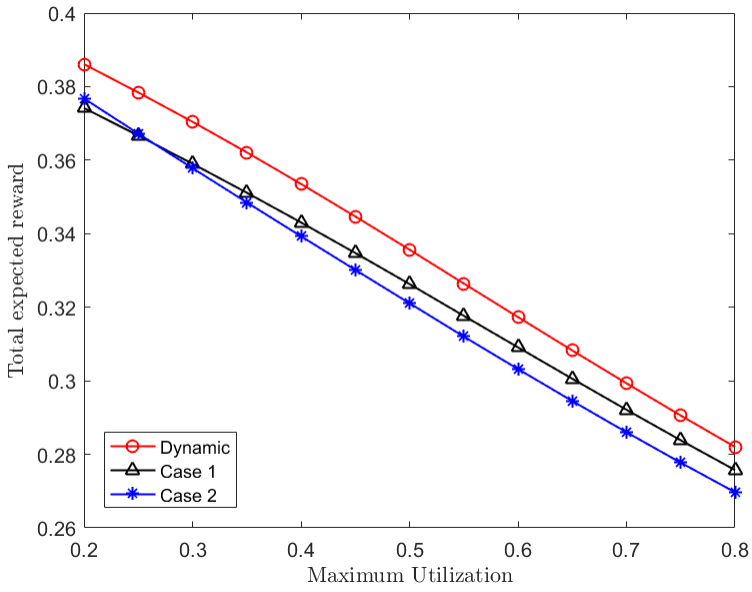


corresponds to one location class. As a result, the number of call classes increases from 15 to $15 \times 270 = 4320$.

We redefine two parameters for the approximate spatial model. The first parameter redefined is the call arrival rate, since we use spatio-temporal demand for each demand location j and time t. The second parameter is the location-dependent coverage function $f^p(s, j)$, computed by analyzing the MCLP+PR solution. For illustration purposes, we choose four different locations (call nodes 85, 98, 176, and 204) among 270 nodes and depict their advanced patient coverage functions in Figure 3.7. The magnitudes of the curves differ due to the distance to stations. For instance, the curve for location 85 is lower than other curves since the node 85 is relatively far from stations compared to other nodes. Advanced coverage functions for all nodes, including those that are not shown in Figure 3.7, are monotone increasing and concave. Basic patient coverage functions follow the same trend. With this dataset, the computational time to solve the MCLP+PR instances and the MDP instances for the extended model is less than 120 CPU seconds.

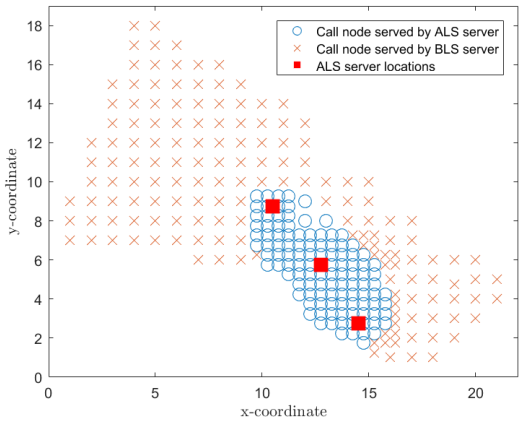
The total expected reward in the location-extended model using the dynamic policy is compared to two static benchmark policies in Figure 3.8. The dynamic policy yields a higher reward than Case 1 and Case 2 policies for all value of utilization.

Figure 3.8: Total expected reward for the approximate spatial model

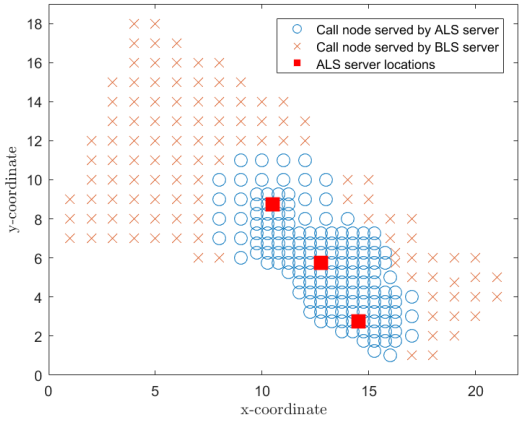


A summary of the optimal action for call class 2 at $t = \frac{T}{2}$ is depicted in Figure 3.9. Figures 3.9a, 3.9b, 3.9c illustrate the optimal actions for $s^A = 1, 2, 3$ and $s^B = 1$. We present three plots with different numbers of available ALS servers to demonstrate how the ALS/BLS response districts change as the resource availability varies. Note that Figure 3.9 differs from Figure 3.6, since the (x, y) coordinate of each call node represents the actual geographic location in the region. Each node is marked either by 'O' if it belongs to the ALS response district or by 'X' if it belongs to the BLS response district. The optimal locations of ALS ambulance bases are also illustrated in the plots. As the number of available ALS servers increases from Figures 3.9a, 3.9b to 3.9c, the geographic area served by ALS servers expands. In other words, when there are more ALS resources available, more call nodes can be included in the ALS response district. For instance, it is optimal to serve a class 2 call at $(10, 10)$ with a BLS server if there is 1 ALS server and 1 BLS server available (Figure 3.9a), and with an ALS server if there are at least 2 ALS servers and 1 BLS server available (Figures 3.9b and 3.9c).

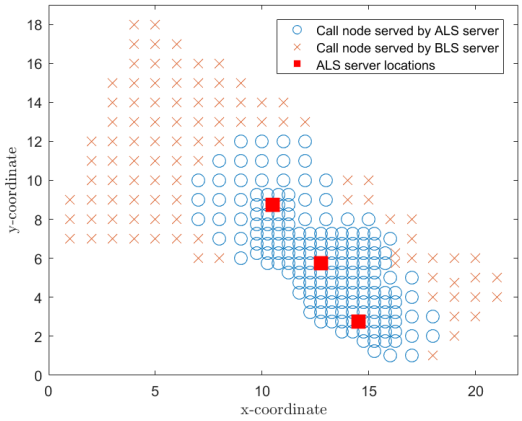
Figure 3.9: Optimal action summary at $t = \frac{T}{2}$ and call class 2



(a) s^A=1, s^B=1



(b) s^A=2, s^B=1



(c) s^A=3, s^B=1

3.6 Conclusion

This paper introduces MDP models that dynamically determine which type of ambulance to send to patients based on the resource availability instead of using a static assignment rule. The study considers an EMS system with multiple types of patients and servers. We report structural properties of the optimal resource assignment policies by showing the conditions under which there exists an optimal policy that is a class separable, signal threshold type, and state control limit type policy. These structural properties provide insight into the design of patient-ambulance matching. In addition, defining these characteristics yields a computationally tractable model. The MDP models can be solved to optimality in a computationally efficient manner for reasonably sized instances. Computational experiments on a real-world dataset show that the use of dynamic matching significantly improves patient coverage compared to benchmark static policies.

The base model considers a Loss system with a finite decision horizon. We also provide analyses of a Queue system and an infinite horizon average cost MDP. Moreover, we discuss an approximate spatial model extension to the base model that incorporates the location of emergency calls into the call class definition to study the spatial features of the EMS system. We provide the conditions through which the structural properties, identified in the base model, can be applied to these model variations.

Since the goal is to maximize the weighted expected coverage of patients, we use RTTs in this paper. The expected reward can reflect the minimum time response or maximum patient survivability objectives that may result in other insights (Zaffar et al., 2016). Future research could consider explicitly modeling server locations or implementing a discrete event simulation to investigate dynamic resource assignment policies in more practical settings. Simulation can lift assumptions, such as Poisson arrival and exponential service times, as well as describe the dispatching process more realistically. Finally, the model in this paper solves a resource assignment problem with the assumption that an ambulance deployment, dispatching, and redeployment policy already exists. It would be advantageous to consider an

integrated model that studies strategic ambulance planning and dynamic resource assignment in a single model.

4 DYNAMIC RESOURCE ASSIGNMENT FOR EMERGENCY RESPONSE WITH TWO SERVICE PHASES

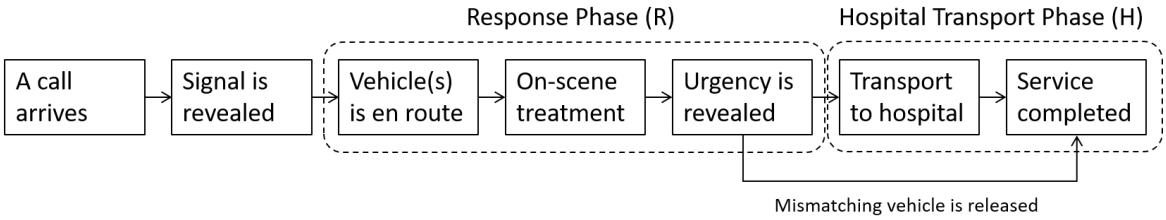
4.1 Introduction

Emergency medical service (EMS) systems pursue two goals when sending emergency vehicles to patients: promptly providing first-aid before a patient's condition deteriorates, and delivering the appropriate type of service based on the patient's health needs. Both objectives have a measurable effect on patient outcomes. For instance, for cardiac arrest patients, the chance of survival decreases by 24% for every additional minute of delay in treatment as compared to basic care (O'Keeffe et al., 2011). Furthermore, Bakalos et al. (2011) show that advanced care in non-trauma cardiac arrest patients increases the probability of patient survival almost by 47% as compared to basic care. To accomplish both goals, EMS systems equipped with multiple types of emergency vehicles (a *tiered system*) often dispatch multiple vehicles of different types to a single call (*multiple response*). Multiple response allows for a faster first-aid and ensures successfully matching vehicles to the patient's uncertain need, at the potential cost of making more vehicles unavailable compared to sending only one vehicle (*single response*).

When more than one ambulance is dispatched to the same call, the service process is different from that of the single response. If only one ambulance responds to an emergency patient, the ambulance must continue transporting the patient to a hospital after on-scene treatment. On the other hand, in multiple response, not all vehicles have to follow the entire service process. Because the patient's actual needs are typically evident once the responders arrive at the scene, only the vehicle that meets the patient's needs (the *matching vehicle*) stays after on-scene treatment to transport the patient to the hospital. The vehicle that does not meet the patient's needs (the *mismatching vehicle*) can be released to prepare for future calls. Therefore, it is important to model the service process as two phases, since unlike single

response vehicles, some multiple response vehicles are freed up during the service process.

Figure 4.1: Service Process with Two Phases



We divide the service process into the response phase and the hospital transport phase. In the response phase, an arriving call is prioritized, the probability that the call needs advanced care (the *signal*) is disclosed, vehicle(s) travels to the scene, and the vehicle(s) performs on-scene treatment. Then the true health needs (*urgency*) of the patient is revealed. In the hospital transport phase, one of the vehicles with transport ability transports the patient to a hospital. It is important in this process that the true urgency of some patients are disclosed during the response phase. If multiple transporting emergency vehicles are dispatched to the call, the vehicle that matches the patient's needs transports the patient to the hospital, while the other becomes available for other calls. This service process is presented in Figure 4.1.

To determine which vehicle to dispatch to the arriving patient, we propose a dynamic resource assignment based on the system status rather than following a static plan. When an emergency call arrives to the system, our model considers the system status, which includes the emergency call at hand, current resource availability of the system, and expected future demand volume, to determine which type of vehicle(s) to dispatch. There are three reasons we study this dynamic resource assignment decision. First, in EMS systems, responses are often slower when more vehicles are currently unavailable, since they are serving other patients (Alanis et al., 2013). Response times are critical for emergency patient survival. However, when more vehicles are occupied, it is likely that the vehicles must travel a longer distance to reach the emergency scene. Therefore, resource assignments should consider resource availability.

Second, EMS systems are non-stationary. The demand for ambulances varies temporally depending on the time of the day or the day of the week (Setzler et al., 2009). An ambulance-dispatching rule that is effective during times of peak demand may under-perform during a different demand pattern. Because there is a tradeoff between serving an emergency patient now and being prepared for future call arrivals, it is important for the decision system to reflect the temporal fluctuation in demand volume.

Lastly, emergency patients have different health needs, and likewise, emergency vehicles vary concerning the level of care they can provide. A tiered EMS system typically aims to provide advanced care by sending advanced life support (ALS) vehicles staffed by paramedics and basic care by sending basic life support (BLS) vehicles staffed by emergency medical technicians. Some EMS systems also deploy non-transport vehicles (NTVs) that may be ALS or BLS because of logistical reasons or budget constraints. The matching of emergency vehicle types (i.e., ALS or BLS) to patient needs is challenging, not only because emergency vehicles are limited but also due to the uncertainty in emergency patient's health needs at the moment of dispatch. For example, 911 operators prioritize each call based upon the limited set of standard questions and answers provided by the caller (Reilly, 2006). After a vehicle arrives at the scene, the call can be upgraded to a higher priority (ALS) or degraded to a lower priority (BLS) based on the patient's conditions observed by the responding personnel.

In this paper, we make the following contributions:

1. We provide a decision-making framework that dynamically assigns ambulance types to arriving emergency calls while allowing multiple response. We formulate a discrete-time, finite horizon Markov decision process (MDP) model with an undiscounted total expected reward criteria that describes the problem in a compact manner.

2. We demonstrate through an empirical study that enabling multiple response is significantly more effective in improving EMS system performance as compared to a system that only employs single response and a static heuristic policy. A sensitivity analysis on parameter values provides insight into the design of ambulance dispatching policies.

3. We show that the model can be extended to an approximate spatial model to consider call locations. An empirical study shows how the approximate spatial model can inform the design of dynamic response districts.

4. We extend the original MDP model to consider non-transport vehicles, which transforms the problem into the admission control of patients to advanced care by ALS NTVs. We conduct an empirical study which shows that the optimal policy sends ALS NTVs more frequently as the signal increases, and as often as possible after the signal exceeds the threshold.

The rest of the paper is organized as follows. Section 4.2 provides a literature review of models for EMS systems. We define the base MDP model that dynamically assigns vehicles to patients while allowing multiple response in Section 4.3. In Section 4.4, we report computational results using real-world datasets to study model insights. Section 4.5 introduces an approximate spatial extension of the original MDP model and provides empirical results. Section 4.6 introduces another extension to the original MDP model that considers non-transport emergency vehicles. We conclude the research in Section 4.7.

4.2 Related Literature

Previous papers consider multiple response in the context of deterministic maximal covering ambulance location problems. Schilling et al. (1979) present mathematical programming models (TEAM/FLEET) that maximize the fraction of demand covered by both primary and special equipment. Moore and ReVelle (1982) propose a location problem with a hierarchy of facilities and services. In their formulation, a demand point is covered when the system offers both lower and higher level services to the point. A lower level service is available if both lower and higher type facilities are present within an appropriate distance. Further, a higher level service is available if a higher type facility is accessible within an appropriate distance.

The aforementioned papers do not consider ambulance unavailability. Some papers propose probabilistic models that consider ambulance unavailability in locating multiple types of vehicles. ReVelle and Marianov (1991) and Marianov and ReVelle (1992) extend Schilling et al.'s FLEET model to locate two types of vehicles to maximize the number of patients who can be served by both types while ensuring individual and joint reliability requirements. Furthermore, Mandell (1998) proposes a probabilistic covering model with ALS vehicles and BLS vehicles that focuses on the ALS response times.

Some papers provide queuing models that consider multiple response in the emergency system context. McLay (2009) considers multiple response by non-transport ALS vehicles and BLS ambulances while focusing on the response time of the first responder. Further, van Barneveld et al. (2017) consider an EMS system with non-transport rapid responder units and transport units. They propose an integer program and a Hypercube model to construct a compliance table—a table indicating the desired location of ambulances based on their availability. Ansari et al. (2017) consider multiple response in an emergency system context by proposing an approximate Hypercube spatial queuing model that allows multiple vehicles to be dispatched to a single call. However, they focus on identical vehicles that complete service at the same time, while our system is a tiered system with multiple types of vehicles that allows early discharge of the mismatching vehicle.

We observe from EMS historical data that the initial classification of the patients—determined by the limited information from the caller—often does not match the patient's actual needs, which are implied by the type of vehicle transported the patient to the hospital when there were multiple vehicle types responding. Therefore, sending multiple types of vehicles to a patient is an effective way to guarantee that the EMS system always satisfies the patient's uncertain needs. Several papers, including Argon and Ziya (2009) and McLay and Mayorga (2013b), study ambulance assignment policies under imperfect information with the patients' needs. However, the literature has not yet considered multiple response as a solution to imperfect information about patients. To the best of our knowledge, our research

is the first to address the dynamics of multiple response in a tiered EMS system when a patient's needs are uncertain at dispatch.

There is a stream of research that models admission control in a queueing system (Stidham and Weber, 1993). In an EMS system context, Chong et al. (2017) is especially relevant as they consider forced-decision making in EMS systems when there are two patient priorities and two types of ambulances. The ideas are related to our consideration of NTVs in Section 4.6. While Chong et al.'s model addresses the routing of low priority patients into a queue while high priority patient service is required, we focus on the admission control of some calls by ALS NTVs while all calls receive BLS ambulance service.

Lastly, we note that the dynamic resource assignment scheme in this paper is inspired by Yoon and Albert (2018a), who present an MDP model that dynamically determines which type of ambulance to dispatch using single response. Our work extends their paper by adding two unique specifications that make the model more practical. First, we allow the model to send multiple vehicles of different types to a single call, regardless of the uncertainty in their health needs. Second, we divide the service process into two phases based on the level of information regarding patient's health needs and allow the early release of some vehicles following the first phase.

4.3 Problem Formulation

To construct the base MDP model, we specify our tiered, multiple-priority EMS system in greater detail. The EMS system contains two types of vehicles—ALS ambulances and BLS ambulances—and three call priority groups—priority 1, priority 2 and priority 3. There are N_A ALS ambulances and N_B BLS ambulances in the system. Priority $i \in \{1, 2, 3\}$ emergency calls at time t arrive according to a Poisson distribution with rate λ_{it} during a finite time horizon of length T. The priority is directly observed at the time of dispatch. Each call/patient type p is either an advanced patient (with $p = a$) who needs advanced care

from ALS ambulance or a basic patient (with $p = b$), and this information is not directly observed at dispatch. Every emergency call that is served, regardless of the priority, enters the response phase with a service time following an exponential distribution with rate μ_R. After the response phase, the call enters the hospital transport phase in which the service time follows an exponential distribution with rate μ_H.

We assume that a patient's health needs are partially identified through its *signal* at dispatch, and then fully disclosed after the emergency vehicles arrive at the scene. The signal ω captures the probability that a patient needs ALS service, and it is a function of the dispatch priority level (1, 2, or 3). A priority 1 call is assumed to have urgent health issues and always needs advanced care (i.e., its signal is 1). Therefore, a priority 1 call can be served either by an ALS ambulance or by co-responding ALS and BLS ambulances. A priority 2 call has signal ω and may need advanced care. Due to this uncertainty at the time of dispatch, the call can also be served by a BLS ambulance in addition to the earlier two options (an ALS ambulance or both an ALS and BLS ambulance). A priority 3 call is assumed to have non-urgent issues (i.e., its signal is 0) and can be served by a BLS ambulance. However, some of these service options are not available when all ALS or BLS vehicles are busy. Therefore, we additionally allow the model to send a BLS ambulance to a priority 1 call when all ALS vehicles are busy. Similarly, when all BLS vehicles are busy, the model can send an ALS vehicle to priority 3 calls. The call arrival rates, the service rates, and the signal can be estimated using historic call data.

We seek to dynamically determine which service option to provide to an arriving call. By doing so, we explicitly consider the tradeoff between serving current emergency patients and being prepared for future calls because vehicles are limited. Therefore, the model determines the optimal matching between service options and emergency patients, and it considers both the current vehicle availability and future call arrivals in this determination. We address this challenge by modeling the ambulance-patient matching problem using an MDP model. The system's state describes the number of busy vehicles in each service phase, and we compute

the rewards based on the promptness of the response as well as the matching of patient's needs and service options.

We assume certain specifications of the system. The first group of assumptions considers service procedures. We assume a non-preemptive service of emergency calls, in which the dispatch decision is made by only observing the signal instead of a patient's true health needs based on operations in practice. For instance, we do not allow the model to send one type of vehicle based on the signal first and another type later after observing the true needs. All patients are assumed to require hospital transport. The second group regards distributional assumptions. The Poisson call arrival assumption is consistent with observations from real-world datasets (Kim and Whitt, 2014). Through numerical experiments, several papers (Ansari et al., 2015; Jagtenberg et al., 2017) validate another distribution assumption—the exponential service time assumption. They show that the model performance does not depend on the choice of service time distribution.

We formally define our MDP model as follows; Table 4.1 provides a summary of the indices and notation used in the model.

Objective. The objective is to maximize the expected total reward by sending either an ALS ambulance, a BLS ambulance, or both of them, to patients over a given time horizon. We assume a finite horizon without a time discount to represent the non-stationary nature of the problem. We uniformize a continuous-time MDP to construct a discrete-time MDP, with uniformization factor $\Lambda = \max_t \sum_{i=1}^{3} \lambda_{it} + (N_A + N_B) \max\{\mu_R, \mu_H\}$ which defines the maximum rate of transition (Serfozo, 1979; Puterman, 1994; Alagoz and Ayvaci, 2010). Note that the uniformization is approximate for the finite horizon MDP, and the approximation gap decreases to zero as the number of time epoch increases to infinity (Miller, 1968).

States. We define the state of the system as a vector of the number of vehicles in each service phase: $s = (s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT})$, where

- s_{AR}=the number of ALS ambulances responding to a call in single response,
- s_{BR}=the number of BLS ambulances responding to a call in single response,

Table 4.1: Summary of Notation

Symbol	Description
$N_A(N_B)$	Number of ALS (BLS) vehicles in the system
T	Number of time epochs $t \in \{1, \cdots, T\}$
$i \in \{1, 2, 3\}$	Call priority groups
$p \in \{a, b\}$	Type of services calls need: a refers to advanced patient, which means that the call is in need of ALS service. b refers to basic patient, which means that the call needs basic support and can be served by either an ALS or BLS vehicle. This information is disclosed after the response phase.
$q' \in \{A, B\}$	Vehicle types: A refers to ALS vehicle and B refers to BLS vehicle
$q \in \{A, B, AB\}$	Service options: A refers to sending an ALS vehicle, B refers to sending a BLS vehicle, AB refers to sending both an ALS vehicle and a BLS vehicle together
λ_{it}	Arrival rate of priority i calls at time t
$\mu_R(\mu_H)$	Service rate in response (hospital transport) phase
ω	Signal, which refers to the probability that priority 2 calls need service from an ALS vehicle (a constant between 0 and 1)
s_t	System state at time t
$n_A(s_t)(n_B(s_t))$	The number of available ALS (BLS) vehicles in system state s_t
d_t	Action (decision) at time t
$P_t(s_{t+1}\|s_t, d)$	Transition probability from state s_t to state s_{t+1} at time $t = 1, ..., T$ given action d
$R_t(s_t, d)$	The expected reward at state s and at time t given action d
U_{pq}	The expected utility gain of serving a type p call with service option q
$f_p(n^q)$	Coverage function that captures the probability that a call in need of type p service is successfully covered in the time threshold when there are n type q vehicles available
$V_t(s)$	Value function (Cost-to-go function) at state s and at time t

- s_{ABR1}=the number of ALS and BLS ambulances co-responding to a priority 1 call,
- s_{ABR2}=the number of ALS and BLS ambulances co-responding to a priority 2 call,
- s_{AT}=the number of ALS ambulances transporting calls to hospitals,
- s_{BT}=the number of BLS ambulances transporting calls to hospitals.

We additionally denote the number of available ALS and BLS vehicles in state s as $n_A(s)$ and $n_B(s)$, where $n_A(s) = N_A - s_{AR} - s_{ABR1} - s_{ABR2} - s_{AT}$ and $n_B(s) = N_B - s_{BR} - s_{ABR1} - s_{ABR2} - s_{BT}$.

Actions. We define the action as $d_t = (d_{1A}, d_{1B}, d_{2A}, d_{2B}, d_{3A}, d_{3B})$, where $d_{iq'} \in \{0, 1\}$ denotes the number of type q' ambulance to dispatch if a priority i call arrives in time epoch t. Note that we only consider dispatch decisions as the action, because the decision regarding which vehicle should transport the patient is automatically determined based on the patient's true needs disclosed after the response phase. The action space $D_t(s_t) \subset \mathbb{B}^6$ is constructed based on the service options described earlier and depends on the state. Note that when all ambulances of at least one type are busy, some of the service options are unavailable. We assume that an arriving call can be lost (neither an ALS nor a BLS ambulance is dispatched) only if all ALS and BLS ambulances are busy. For instance, for priority 3 calls, the only possible action is $d_{3A} = 0$ and $d_{3B} = 1$ if there is at least one BLS vehicle available (state s such that $n_A(s) > 0, n_B(s) > 0$). However, because this action is not available if all BLS ambulances are serving calls, the only possible action choice for priority 3 calls in such states (state s such that $n_A(s) > 0, n_B(s) = 0$) is $d_{3A} = 1$ and $d_{3B} = 0$.

Transition Probabilities. Exactly one of the following transitions occurs between two consecutive time epochs:

- A priority 1 call arrives (with rate λ_{1t})
  - and is served by an ALS ambulance ($d_{1A} = 1$, $d_{1B} = 0$):
    $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \to (s_{AR} + 1, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT})$
  - and is served by co-responding ALS and BLS ambulances ($d_{1A} = 1$, $d_{1B} = 1$):
    $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \to (s_{AR}, s_{BR}, s_{ABR1} + 1, s_{ABR2}, s_{AT}, s_{BT})$
  - and is served by a BLS ambulance (if all ALS ambulances are busy, $d_{1A} = 0$, $d_{1B} = 1$):
    $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \to (s_{AR}, s_{BR} + 1, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT})$
- A priority 2 call arrives (with rate λ_{2t})
  - and is served by an ALS ambulance ($d_{2A} = 1$, $d_{2B} = 0$):
    $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \to (s_{AR} + 1, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT})$
  - and is served by co-responding ALS and BLS ambulances ($d_{2A} = 1$, $d_{2B} = 1$):
    $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \to (s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2} + 1, s_{AT}, s_{BT})$
  - and is served by a BLS ambulance ($d_{2A} = 0$, $d_{2B} = 1$):
    $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \to (s_{AR}, s_{BR} + 1, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT})$
- A priority 3 call arrives (with rate λ_{3t})

  - and is served by an ALS ambulance (if all BLS ambulances are busy, $d_{3A} = 1$, $d_{3B} = 0$):
    $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \rightarrow (s_{AR} + 1, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT})$
  - and is served by an BLS ambulance ($d_{3A} = 0$, $d_{3B} = 1$):
    $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \rightarrow (s_{AR}, s_{BR} + 1, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT})$

- A priority 1, 2 or 3 call arrives when all ALS and BLS ambulances are busy and is lost ($d_{iq} = 0$ for $i \in \{1, 2, 3\}$ and $q \in \{A, B\}$):
  $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \rightarrow (s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT})$

- A priority 1, 2 or 3 call in the response phase served by an ALS ambulance enters the hospital transport phase (with rate $\mu_R s_{AR}$):
  $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \rightarrow (s_{AR} - 1, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT} + 1, s_{BT})$

- A priority 1, 2 or 3 call in the response phase served by a BLS ambulance enters the hospital transport phase (with rate $\mu_R s_{BR}$):
  $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \rightarrow (s_{AR}, s_{BR} - 1, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT} + 1)$

- A priority 1 call in the response phase served by co-responding ALS and BLS ambulances enters the hospital transport phase with an ALS ambulance (with rate $\mu_R s_{ABR1}$):
  $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \rightarrow (s_{AR}, s_{BR}, s_{ABR1} - 1, s_{ABR2}, s_{AT} + 1, s_{BT})$

- A priority 2 call in the response phase served by co-responding ALS and BLS ambulances is an advanced patient and enters the hospital transport phase with an ALS ambulance (with rate $\omega\mu_R s_{ABR2}$):
  $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \rightarrow (s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2} - 1, s_{AT} + 1, s_{BT})$

- A priority 2 call in the response phase served by co-responding ALS and BLS ambulances is a basic patient and enters the hospital transport phase with a BLS ambulance (with rate $(1 - \omega)\mu_R s_{ABR2}$):
  $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \rightarrow (s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2} - 1, s_{AT}, s_{BT} + 1)$

- An ALS ambulance finishes transportation (with rate $\mu_H s_{AT}$):
  $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \rightarrow (s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT} - 1, s_{BT})$

- A BLS ambulance finishes transportation (with rate $\mu_H s_{BT}$):
  $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \rightarrow (s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT} - 1)$

- Neither a call arrives nor a ambulance(s) finishes response/hospital transport phase (dummy event in which nothing happens):
  $(s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \rightarrow (s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT})$

Rewards. The system achieves the reward upon the call arrival and the corresponding ambulance dispatch. The reward is obtained by multiplying the utility of matching a service option to an emergency call and the coverage function.

Table 4.2: Utility of serving a call

		Patient's needs:	
		Advanced patient (a)	Basic patient (b)
action:	send ALS (A)	U_{aA}	U_{bA}
	send BLS (B)	U_{aB}	U_{bB}
	send ALS and BLS (AB)	U_{aAB}	U_{bAB}

Table 4.3: Coverage function

		Patient's needs:	
		Advanced patient (a)	Basic patient (b)
action:	send ALS (A)	$f^a(n_A)$	$f^b(n_A)$
	send BLS (B)	$f^a(n_B)$	$f^b(n_B)$
	send ALS and BLS (AB)	$f^a(n_A, n_B)$	$f^b(n_A, n_B)$

The reward has two components. First, the utility parameter incentivizes the proper matching between the patient true need ($p \in \{a, b\}$) and the service option provided to the patient ($q \in \{A, B, AB\}$), as defined in Table 4.2. We assume that $U_{aA} > U_{aB}$ and $U_{aAB} > U_{aB}$ so that it disincentivizes the model from not providing ALS services to advanced patients. These utilities can be selected by decision-makers based on existing protocols. Typically, $U_{aA} = U_{aAB} = 1$, with U_{aB}, U_{bA}, and U_{bAB} set relative to 1 based on the relative value these responses represent relative to ALS vehicles responding to advanced patients. The utility values can be estimated using data: for instance, as far as non-trauma cardiac arrest patients are concerned, we can set $U_{aA} = 1.0$ and $U_{aB} = 0.68$ using the result of Bakalos et al. (2011).

Second, the coverage function represents the probability that the ambulance(s) can reach the patient in a pre-defined response time threshold (RTT). Due to the physical distance between ambulances bases and call locations, given an arbitrary call location, only a subset of ambulance bases can respond within the RTT. More specifically, the coverage function value $f^p(n_q)$ refers to the fraction of type p patients that can be covered within the RTT when n_q type q ambulances are available. Therefore, we implicitly model the location of calls through the coverage functions. We provide the full configuration of coverage functions in Table 4.3. All coverage functions should be monotone non-decreasing in n_A and n_B.

When both ALS and BLS ambulances respond to the same call, we consider the response time of the first responder to determine if the call is covered within the RTT; therefore, $f^p(n_A, n_B) \geq \max\{f^p(n_A), f^p(n_B)\}$ for $p \in \{a, b\}$.

One way to derive coverage functions is as follows. We can compute an upper bound to $f^a(n)$ by solving the Maximal Covering Location Problem (MCLP) with n ambulances and the RTT for type p emergency calls, as Maxwell et al. (2010) suggest. The basic patient multiple response coverage function, $f^b(n_A, n_B)$, can be easily estimated by $f^b(n_A, n_B) \geq \max\{f^b(n_A), f^b(n_B)\}$. To estimate the multiple response coverage function for advanced patients ($f^a(n_A, n_B)$), we suggest solving a variant of MCLP that locates two types of ambulances. To formulate such a variant, we selectively adapt the hierarchical service location model by Moore and ReVelle (1982) and Mandell (1998). A call node $j \in J$ can be covered by one of two options: (1) the node is within R^* from a station with a BLS ambulance and within R^{**} from a station with an ALS ambulance, and (2) the node is within S^* from a station with an ALS ambulance ($R^* \leq S^* \leq R^{**}$). Let x_k^q be a binary variable that takes on the value 1 if a type $q \in \{A, B\}$ ambulance is located at station $k \in K$; y_j be a binary variable that takes on the value 1 if node $j \in J$ is covered; and y_j^r be a binary variable that takes on the value 1 if node $j \in J$ is covered with option $r \in \{1, 2\}$. Our formulation for the Maximal Covering Location Problem with 2 types of ambulances (MCLP2) is as follows:

$$
\begin{aligned}
f^a(n_A, n_B) = \max \frac{1}{\sum_j \lambda_j} \sum_j \lambda_j y_j \\
\text{s.t.} \qquad & \sum_{k \in K} x_k^q \le n_q \qquad \forall q \in \{A, B\}, \\
& \sum_{i:d(k,j)<R^*} x_k^A \ge y_j^1 \qquad \forall j \in J, \\
& \sum_{i:d(k,j)<R^{**}} x_k^B \ge y_j^1 \qquad \forall j \in J, \\
& \sum_{i:d(k,j)<S^*} x_k^A \ge y_j^2 \qquad \forall j \in J, \\
& y_j^1 + y_j^2 \ge y_j \qquad \forall j \in J, \\
& x_k^A, x_k^B \in \{0,1\} \qquad \forall k \in K, \\
& y_j^1, y_j^2, y_j \in \{0,1\} \qquad \forall j \in J,
\end{aligned}
$$

where λ_j represents the demand at call node j.

The reward function for the state s and the action d is therefore:

$$
\begin{aligned}
& R_t(s,d) \\
& = \frac{\lambda_{1t}}{\Lambda}\big(d_{1A}\bar{d}_{1B} f_a(n_A(s))U_{HA} + \bar{d}_{1A} d_{1B} f_a(n_B(s))U_{HB} + d_{1A} d_{1B} f_a(n_A(s), n_B(s))U_{HAB}\big) \\
& + \frac{\lambda_{2t}}{\Lambda}\omega\big(d_{2A}\bar{d}_{2B} f_a(n_A(s))U_{HA} + \bar{d}_{2A} d_{2B} f_a(n_B(s))U_{HB} + d_{2A} d_{2B} f_a(n_A(s), n_B(s))U_{HAB}\big) \\
& + \frac{\lambda_{2t}}{\Lambda}(1-\omega)\big(d_{2A}\bar{d}_{2B} f_b(n_A(s))U_{LA} + \bar{d}_{2A} d_{2B} f_b(n_B(s))U_{LB} + d_{2A} d_{2B} f_b(n_A(s), n_B(s))U_{LAB}\big) \\
& + \frac{\lambda_{3t}}{\Lambda}\big(d_{3A}\bar{d}_{3B} f_b(n_A(s))U_{LA} + \bar{d}_{3A} d_{3B} f_b(n_B(s))U_{LB}\big),
\end{aligned}
$$

where $\bar{d}_{iq} = 1 - d_{iq}$. The first line in the equation captures the reward when a priority 1 call arrives; the second and the third line capture the reward when a priority 2 call arrives; and the last line captures the reward when a priority 3 call arrives. Note that our reward function is an expected reward at dispatch, because the reward depends on the true health needs

of the patient that is revealed later. Therefore, the reward reflects the partially observable information at the time of dispatch reflected by the signal.

Optimality Equations. We solve the discrete-time MDP using backward induction. We compute the value functions by solving the following optimality equations (Bellman equations):

$$V_t(s_t) = \max_{d_t \in D_{s_t}} \left\{ R_t(s_t, d_t) + \sum_{s_{t+1} \in S_{t+1}} P_t(s_{t+1}|s_t, d) V_{t+1}(s_{t+1}) \right\}, \qquad \forall t \in \{1, ..., T\}, \tag{4.1}$$

where $V_t(s_t)$ is the value function of state s_t at time epoch t, $R_t(s_t, d)$ is the reward of taking action d at state s_t and time epoch t, and $P_t(s_{t+1}|s_t, d)$ is the probability of the transition from state s_t to state s_{t+1} at time t with action d. At the last time epoch, we assume zero salvage values ($V_T(s_T) = 0$ for all s_T).

The optimal policies are class separable. Due to the class separability structural property of the optimal policy, we can reduce the number of actions we need to evaluate to solve the optimality equations. When we determine the optimal action for priority p patient, the optimal action for other priorities does not affect the decision. This idea can be generalized to the setting in which there are more than three patient classes. Suppose there are P different patient classes instead of 3 as in our model. Then we can compute the value functions by comparing at most three actions for each call priority (which involves comparing three actions P times each time epoch, instead of considering all possible combinations (which involves comparing 3^P actions each time epoch). This class separability property is especially useful when the number of patient classes P is large; for instance, in Section 4.5, we provide an extension of the original MDP model that has $P = 810$ patient classes. More details of the class separability and the proof can be found in Appendix C.1.

Note that an additive reward function that adds the matching utility to the coverage function could be considered as an alternative to the multiplicative reward function. Numerical experiments yield the same insights as those from the multiplicative reward model. For simplicity, we only consider a multiplicative reward function in the remainder of this paper.

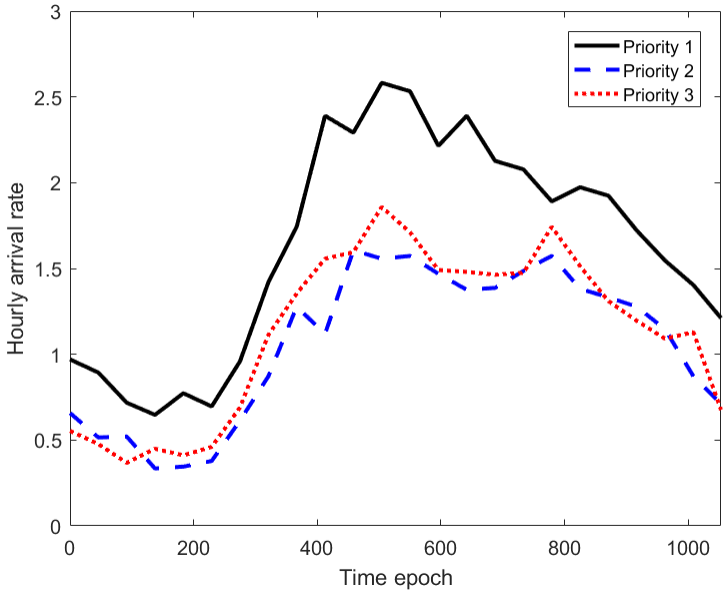
4.4 Computational Results

To illustrate the optimal policies, we conduct a numerical study using real emergency call logs from Hanover County, Virginia. The system contains $N_A = 6$ ALS ambulances and $N_B = 6$ BLS ambulances. The utility values are set to $U_{aA} = U_{aAB} = 1.0$, $U_{aB} = 0.5$, and $U_{bA} = U_{bB} = U_{bAB} = 0.5$, assuming that providing advanced care to an advanced patient produces twice as much value as under-serving such a patient or serving a basic patient. The signal ω, which is the probability that a priority 2 patient needs ALS services, is estimated from the data as 0.174. Considering this value as a base case, we also vary the value of the signal from 0.0 to 1.0 in a sensitivity analysis. We estimate the value of service rates μ_R and μ_H from the call log as 3.32 and 1.43 calls per hour (hr^{-1}), respectively. We normalize the call arrival rates so that the value of the instantaneous maximum utilization ($\frac{\max_t\{\lambda_{1t}+\lambda_{2t}+\lambda_{3t}\}}{(N_A+N_B)(\mu_R^{-1}+\mu_H^{-1})^{-1}}$) equals 0.5 in the base case. Further, we uniformly scale the call arrival rates to test the model for systems with various values of the maximum utilization ranging from 0.3 to 0.9 in a sensitivity analysis. Figure 4.2 shows the base case non-stationary arrival rates of priority 1, 2 and 3 emergency calls that span 24 hours, which are converted into $T = 1,024$ discrete time epochs after the uniformization.

We compare the performance of the optimal policy from our MDP model that allows multiple response (denoted as *optimal policy*) to two other benchmark policies. The first benchmark policy (denoted as *single response policy*) is an optimal MDP policy that always sends up to one vehicle to a call and does not allow multiple response. The other benchmark policy (denoted as *static heuristic policy*) sends ALS vehicles to priority 1 and 2 calls and BLS vehicles to priority 3 calls. This policy can identified using the base MDP model with a restricted action set.

Figure 4.3 illustrates the total expected reward of the optimal policy and two benchmark policies while varying the maximum utilization (Figure 4.3a) and the signal (Figure 4.3b). The difference between the optimal policy and the single response policy represents the value of multiple response option. The difference between the single response policy and the static

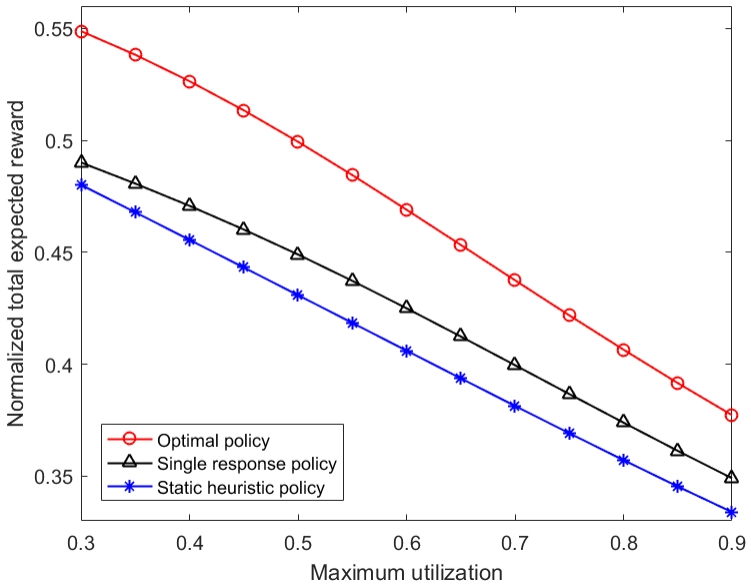
Figure 4.2: Non-stationary arrival rates



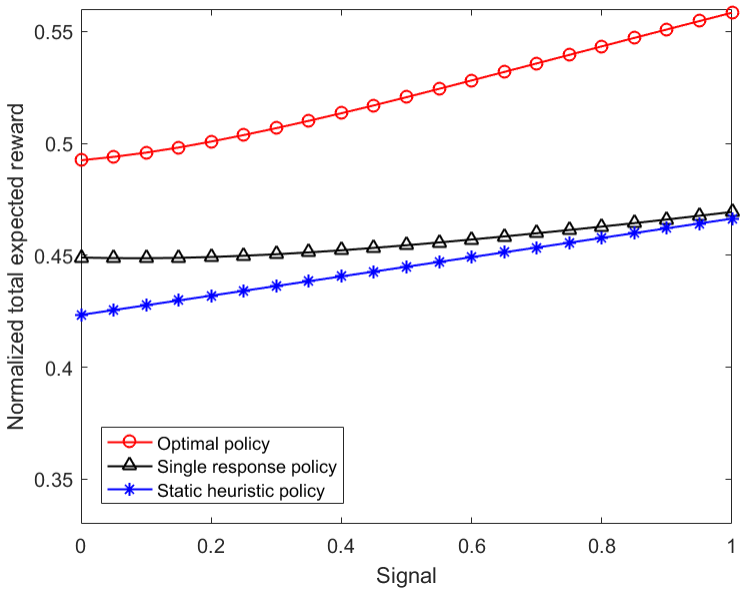
heuristic policy shows the value of dynamic ambulance-patient matching dependent on the system's status. Figure 4.3a shows that a multiple response option results in a significant increase in the total expected reward for any value of the utilization, while the improvement's magnitude is more pronounced across lower values of the utilization. We compare the total expected reward under various values of the signal in Figure 4.3b. In this figure, the maximum utilization is 0.5, and we vary the signal values from 0.0 to 1.0. The optimal policy always yields significantly higher rewards than the two benchmark policies across all signal values.

It is notable that the improvement in total expected reward by adding the multiple response option (the difference between the optimal policy and the single response policy) is significantly higher than the improvement by implementing dynamic assignment scheme (the difference between the single response policy and the static heuristic policy). The multiple response option adds a small amount of complexity to the dynamic assignment policy. Therefore, if the system can implement dynamic ambulance-patient matching, allowing for multiple response is an effective way to improve system performance with little to no additional cost.

Figure 4.3: Normalized total expected reward

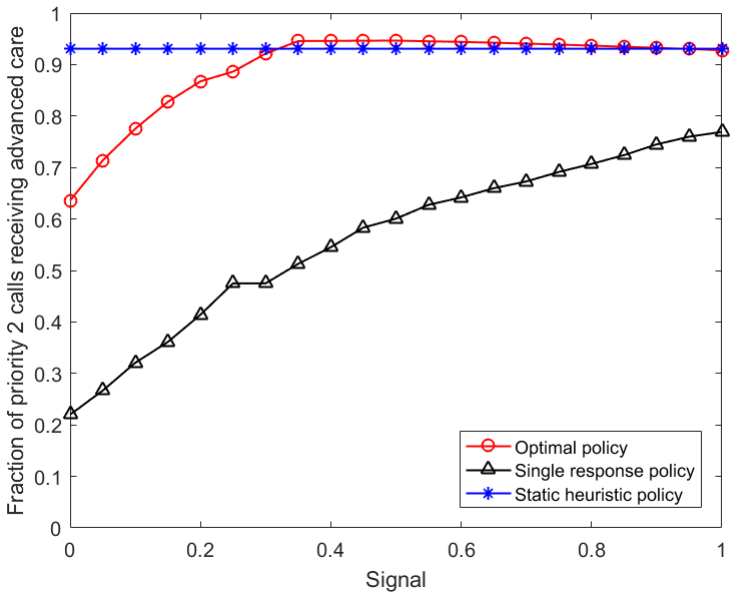


(a) Maximum Utilization



(b) Signal

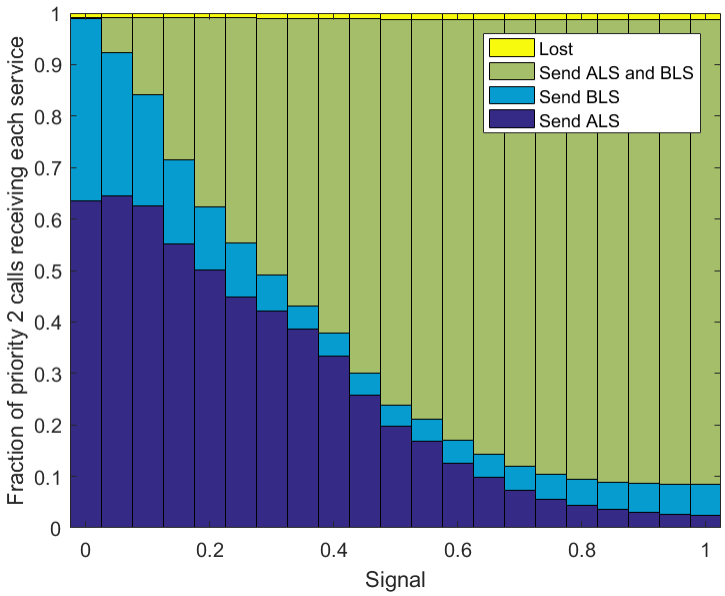
Figure 4.4: Fraction of priority 2 patients receiving advanced care



To understand what causes the difference in the total expected reward between policies, we compare how often priority 2 patients receive advanced care under the the optimal multiple response policy versus the single response policy. This is illustrated in Figure 4.4. Under the optimal policy, a priority 2 patient receives advanced care if an ALS ambulance alone responds or if both ALS and BLS ambulances respond. In contrast, under the single response policy, the curve simply represents how frequently an ALS ambulance is dispatched to a priority 2 call instead of a BLS ambulance. Figure 4.4 indicates that if the system allows multiple response, then more priority 2 patients can receive advanced care for any value of the signal. This means the system can successfully match health needs of advanced patients more often, which contributes to the improvement in the total expected reward shown in Figure 4.3.

We further investigate how the optimal policy serves priority 2 patients in Figure 4.5, which shows the fraction of priority 2 patients that receive each service option for various values of the signal. Interestingly, the optimal policy does not send ALS ambulances alone more often for higher vales of the signal when priority 2 calls are more likely to require ALS

Figure 4.5: Fraction of priority 2 patients receiving each service under the optimal policy allowing multiple response



service. Instead, the optimal policy chooses multiple response more frequently for higher signal values. As the signal increases, the system expects more advanced patient arrivals during the time horizon, which means ALS ambulances also become more valuable. This motivates the system to prevent ALS ambulances transporting priority 2 calls to hospitals when they do not need ALS service. As a result, the optimal policy sends both vehicles to priority 2 calls more often when the signal is higher.

4.5 Approximate Spatial Model Allowing Multiple Response

In real EMS systems, the dispatcher considers the location of the emergency patient in addition to the call priority when determining which types of vehicles to send. However, the original MDP model in Section 4.3 does not explicitly consider call locations, although call locations are implicitly modeled by the reward functions through the coverage functions. One way to consider call locations is to design pre-determined ALS (BLS) response districts—regions

served by ALS (BLS) ambulances—and update these regions when resource availability changes. In this subsection, we provide an extension of the original model in Section 4.3 that enables such decision-making.

We present the *approximate spatial model* extension of the original MDP model by incorporating incident locations in the action space definition. Let $j \in J$ denote call nodes that capture the possible locations of emergency calls. Then, an arriving call is represented by a tuple (i, j), where $i \in \{1, 2, 3\}$ represents call priority, and j represents the call node. We allow the model to decide the optimal action based not only on the call priority but also on the call location. Therefore, the approximate spatial model has an extended action $d = (d^j_{iq}, \forall i \in \{1, 2, 3\}, q \in \{A, B\}, j \in J)$, where $d^j_{iq} \in \{0, 1\}$ is the number of type q ambulances dispatched if a priority i call occurs at node j.

More information is available for an arriving call under this setup, and therefore, we redefine the coverage functions so that they become location-specific. Let $f^j_{pq}(n)$ denote the coverage function that represents the probability that an emergency call at node j in need of type p service is successfully covered in the given RTT by a type q ambulance when there are n type q ambulances available. Assuming that all ambulances are equally busy and then considering all possible configurations of the ambulances' busy statuses given the number of available ambulances, we can compute $f^j_{pq}(n)$ as the probability that a type q ambulance responds to a type p call at node j within a pre-determined RTT.

Suppose that the optimal location of N_q ambulances of type $q \in \{A, B\}$ from the MCLP2 solution is $\mathbb{I}$. Let $\mathbb{K}$ denote the set of all possible combinations of n locations out of $\mathbb{I}$. Each element in $\mathbb{K}$ is a list of locations of length-n and represents a possible scenario of n ambulances' availability statuses given that s ambulances are available. Because we assume that all ambulances are equally busy, all scenarios have the same probability. We can calculate the single response coverage as follows: $f^j_{pq}(n) = \frac{1}{|\mathbb{K}|} \sum_{k \in \mathbb{K}} \max_{e \in k} R^p_{ej}$, where R^p_{ej} is the probability that a type q ambulance at station e responds to a type p call within the RTT. For multiple response ($q = AB$), we use $f^j_{pAB}(n) = \max\{f^j_{pA}(n), f^j_{pB}(n)\}$, since we focus on

the response time of the first responder. However, alternatives are acceptable.

Based on this newly defined coverage function $f^j_{pAB}(n)$, the reward function for state s and the action d is also redefined as:

$$
\begin{aligned}
&R_t(s,d)\\
&=\sum_{j\in J}\frac{\lambda^j_{1t}}{\Lambda}\big(d_{1A}\bar{d}_{1B}f^j_{aA}(n_A(s))+\bar{d}_{1A}d_{1B}f^j_{aB}(n_B(s))+d_{1A}d_{1B}f^j_{aAB}(n_A(s),n_B(s))\big)\\
&+\sum_{j\in J}\frac{\lambda^j_{2t}}{\Lambda}\omega\big(d_{2A}\bar{d}_{2B}f^j_{aA}(n_A(s))+\bar{d}_{2A}d_{2B}f^j_{aB}(n_B(s))+d_{2A}d_{2B}f^j_{aAB}(n_A(s),n_B(s))\big)\\
&+\sum_{j\in J}\frac{\lambda^j_{2t}}{\Lambda}(1-\omega)\big(d_{2A}\bar{d}_{2B}f^j_{bA}(n_A(s))+\bar{d}_{2A}d_{2B}f^j_{bB}(n_B(s))+d_{2A}d_{2B}f^j_{bAB}(n_A(s),n_B(s))\big)\\
&+\sum_{j\in J}\frac{\lambda^j_{3t}}{\Lambda}\big(d_{3A}\bar{d}_{3B}f^j_{bA}(n_A(s))+\bar{d}_{3A}d_{3B}f^j_{bB}(n_B(s))\big),
\end{aligned}
\tag{4.2}
$$

where $\bar{d}_{iq}=1-d_{iq}$ and λ^j_{it} refers to the spatiotemporal hourly arrival rate of priority i patients at demand node j during time epoch t. The first line in (4.2) captures the reward when a priority 1 call arrives; the second and the third lines capture the reward when a priority 2 call arrives; and the last line captures the reward when a priority 3 call arrives.

We conduct numerical experiments using the Hanover County emergency call logs from Section 4.4 to demonstrate the approximate spatial model empirically. We consider a system containing $N_A=3$ ALS ambulances and $N_B=3$ BLS ambulances. The service rate is normalized to 1 call per hour, and the arrival rates are also normalized to meet the target maximum utilization values ranging from 0.3 to 0.9. We divide the geographic region into $|J|=270$ demand nodes that consist of 139 2×2 mile cells that represent regions with lower demand and 131 1×1 mile cells that represent regions with higher demand. The spatiotemporal arrival rates λ^j_{it} are estimated from the call logs. Figure 4.6 provides arrival rates for each nodes (darker cell represents higher demand). Figure 4.6 also shows 16 candidate station locations in Hanover County. Finally, the location-specific coverage functions, $f^j_{pq}(n)$,

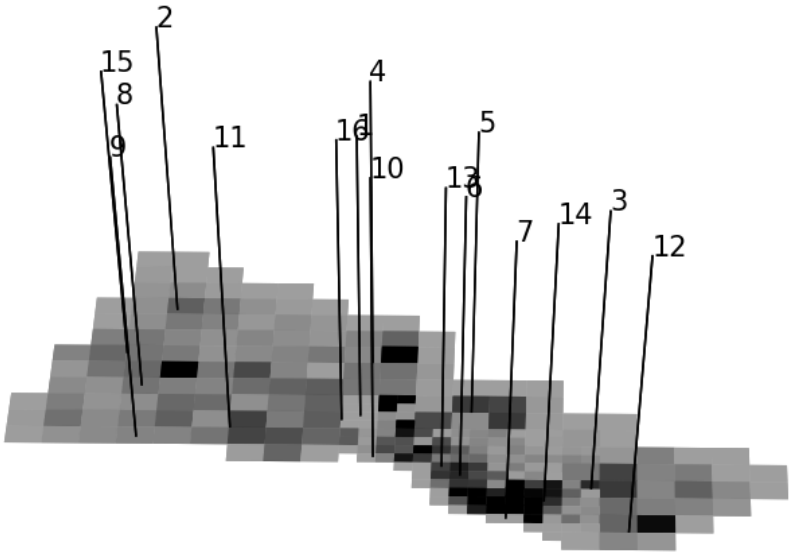


Figure 4.6: Arrival rates for each demand node and positions of ambulance stations

are obtained from MCLP2 solutions. Based on the MCLP2 solution, ALS ambulances are located at stations 7, 13, and 16, and BLS ambulances are located at stations 3, 4, and 7.

Figure 4.7: Normalized total expected reward for the approximate spatial model

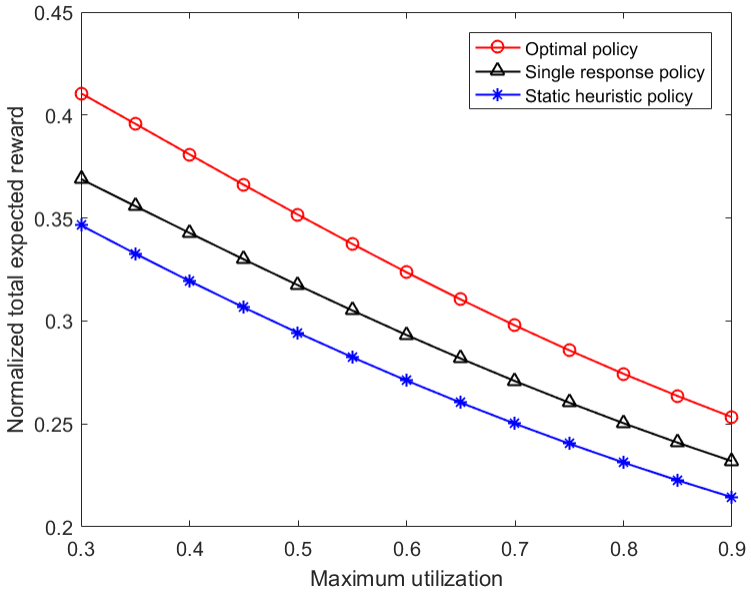
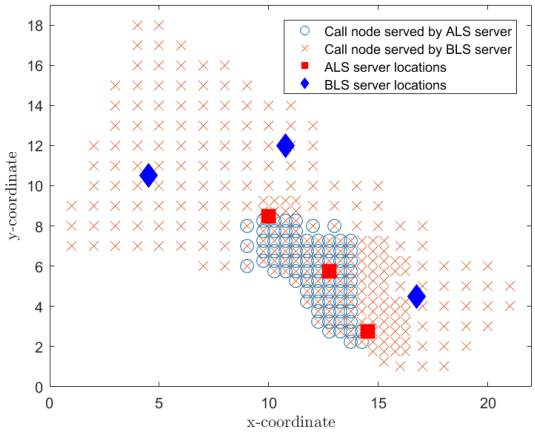


Figure 4.7 shows the normalized expected reward of the optimal policy and the benchmark policies for the approximate spatial model under various values of the maximum utilization factor. This figure exhibits the same trends as the original model results in Figure 4.3. First,

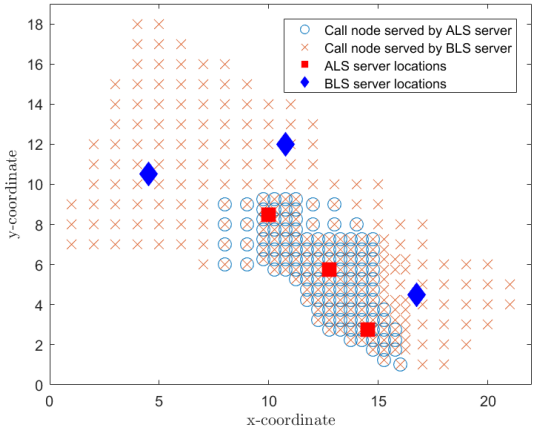
allowing multiple response results in a significant increase in the total expected reward; and second, the improvement is more significant for lower values of the maximum utilization.

Figure 4.8 shows the optimal action for priority 2 patients at $t = \frac{T}{2}$. We choose to check the optimal action at the middle of the time horizon ($t = \frac{T}{2}$) to minimize transient effects at the beginning or the end of the time horizon. Each demand node j has (x, y) coordinates that relate to a geographic location in the city. A demand node is marked with $\bigcirc$, $\times$, or $\otimes$, depending on which service option is provided to a priority 2 call that arrive during time epoch t: an ALS ambulance responds if the patient is at a node marked with $\bigcirc$; a BLS ambulance responds if the patient location is marked with $\times$; and an ALS ambulance and a BLS ambulance co-respond if the patient location is marked by $\otimes$. Figures 4.8a – 4.8c also display the optimal location of ambulances obtained by solving an MCLP2 instance, by marking ALS ambulance stations with red $\square$ and BLS ambulance stations with blue $\lozenge$. However, the ambulance locations are not explicitly accounted for in the model, so Figure 4.8 illustrates average response districts across different ambulance availabilities. Comparing Figure 4.8a, 4.8b, and 4.8c demonstrates how response districts change as the ambulance availability varies. When more ALS ambulances are busy, only the nodes that are closest to ALS stations receive advanced care. As an additional ALS ambulance becomes available—from Figure 4.8a to Figure 4.8b and then Figure 4.8c—the ALS response district continuously expands so that more call nodes receive advanced care. For example, a priority 2 patient at node $(10, 10)$ is served by a BLS ambulance if at least one ALS ambulance is busy responding to other calls (Figure 4.8a and 4.8b). Node $(10, 10)$ receives advanced care by co-responding ALS and BLS ambulances only if all ALS ambulances are available (Figure 4.8c).

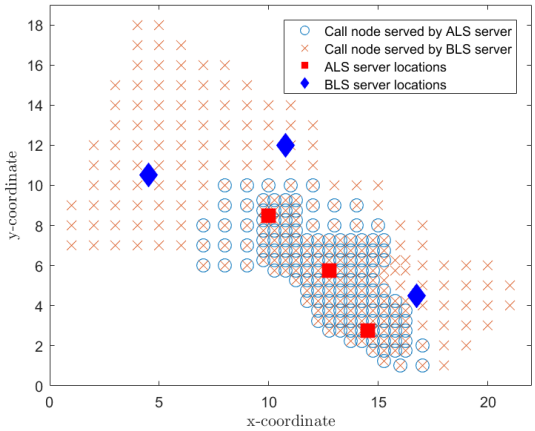
Figure 4.8: Optimal action summary at $t = \frac{T}{2}$ for priority 2 calls



(a) s=(2,1,0,0,0,0): two ALS ambulances and one BLS ambulance are in response phase



(b) s=(1,1,0,0,0,0): one ALS ambulances and one BLS ambulance are in response phase



(c) s=(0,1,0,0,0,0): one BLS ambulance is in response phase

4.6 Model with Non-Transport Units

Many EMS systems utilize NTVs that can easily access incident locations and improve response times. However, few papers consider the optimal planning of EMS systems with NTVs, with some notable exception. Both McLay (2009) and van Barneveld et al. (2017) study ambulance location problems for EMS systems equipped with NTVs and transport vehicles, and both propose Hypercube models and integer programs. However, both focus on the EMS systems' steady-state performances rather than their non-stationary nature, and neither study dispatch decisions.

To fill the gap in the literature regarding the dispatch of NTVs, we extend the original MDP model in Section 4.3 to consider a different type of tiered EMS system that consists of ALS NTVs and BLS ambulances. Unlike ambulances, NTVs cannot transport patients to a hospital. If a NTV responds to an emergency call, then a BLS ambulance must also be sent to transport the patient to a hospital. Therefore, we update the list of our service options as follows: a priority 1 call can be served by a co-responding ALS NTV and a BLS ambulance; a priority 2 call can be served by either both of them or just a BLS ambulance; and a priority 3 call can be served by a BLS ambulance alone. In this section, we denote this extension as the *NTV model*, and we describe how we modify the original model to consider the system with NTVs. To accommodate this change, we update the base MDP model as follows:

States. We redefine the state of the system with the number of vehicles in each service phase as $s = (s_{BR}, s_{ABR1}, s_{ABR2}, s_{ABT}, s_{BT})$, where s_{BR} is the number of BLS ambulances responding to a call; s_{ABR1} is the number of ALS NTVs and BLS ambulances co-responding to a priority 1 call; s_{ABR2} is the number of ALS NTVs and BLS ambulances co-responding to a priority 2 call; s_{ABT} is the number of ALS NTVs and BLS ambulances co-transporting calls to hospitals; and s_{BT} is the number of BLS ambulances transporting calls to hospitals.

Actions. As before, the actions are $d_t = (d_{1A}, d_{1B}, d_{2A}, d_{2B}, d_{3A}, d_{3B})$, but here we restrict the model to send a BLS ambulance to every call ($d_{pB} = 1$ for $p \in \{1, 2, 3\}$) as long as there is at least one BLS vehicle available. Consequently, the problem can be viewed as an admission

control of priority 2 patients (Chong et al., 2017) to ALS services, because we always send a BLS ambulance and only determine whether to also send an ALS NTV. If a call arrives when all BLS ambulances are busy, we do not dispatch any vehicle and consider that call lost.

Transition Probabilities Exactly one of the following transitions occurs between two consecutive time epochs:

- a priority 1 call arrives and is responded by an ALS NTV and a BLS ambulance,
- a priority 1 call arrives and is responded by a BLS ambulance (only when all ALS NTVs are busy),
- a priority 2 call arrives and is responded by an ALS NTV and a BLS ambulance,
- a priority 2 call arrives and is responded by a BLS ambulance,
- a priority 3 call arrives and is responded by a BLS ambulance,
- a priority 1, 2 or 3 call arrives and is lost (only when all BLS ambulances are busy),
- a priority 1 call in the response phase with an ALS NTV and a BLS ambulance enters the hospital transport phase,
- a priority 2 call in the response phase with an ALS NTV and a BLS ambulance is an urgent patient and enters the hospital transport phase with both vehicles,
- a priority 2 call in the response phase with an ALS NTV and a BLS ambulance is a non-urgent patient and enters the hospital transport phase with the BLS ambulance, while the ALS NTV is released,
- a priority 3 call in the response phase with a BLS ambulance enters the hospital transport phase,
- an ALS QRV and a BLS ambulance finishes the hospital transport phase,
- a BLS ambulance finishes the hospital transport phase,
- nether a call arrives nor a vehicle(s) finishes a service phase (dummy event in which nothing happens).

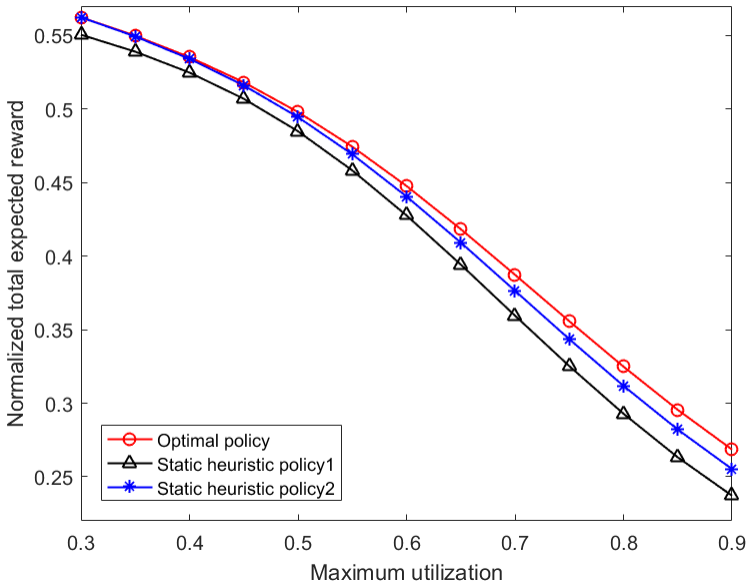
Finally, the definition of the reward is analogous to the base model and thus we omit the details. This model can be extended to consider call locations using techniques introduced in Section 4.5.

We conduct an empirical study using the same data as in Section 4.4. However, regarding NTVs, it is reasonable to consider a tiered EMS system in which the majority of vehicles are BLS transport units rather than a system with an equal number of ALS and BLS vehicles, because sending transport vehicles to emergency calls is always forced in this model. In this context, we consider a system equipped with $N_A = 3$ ALS NTVs and $N_B = 9$ BLS ambulances.

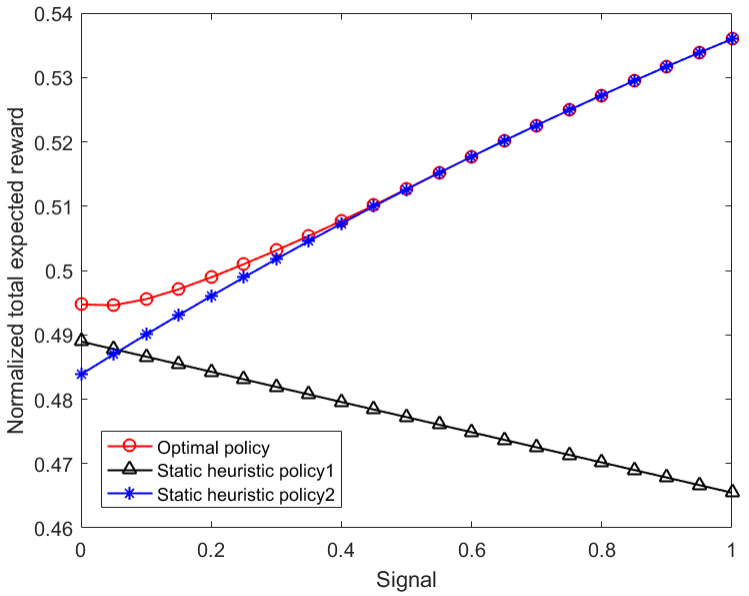
We compare the optimal MDP policy to two benchmark static policies: static heuristic policy 1, which sends a BLS ambulance to priority 2 patients; and static heuristic policy 2, which sends both an ALS NTV and a BLS ambulance to priority 2 patients when available. Figure 4.9 compares the performance of the optimal policy and two benchmark policies. Figure 4.9a compares the performances across various values of the maximum utilization. The differences between the optimal policy and benchmark static policies are greater when the maximum utilization is higher; however, the magnitude of the improvement over benchmark static policies is less pronounced in contrast to the original model shown in Figure 4.3. Figure 4.9b compares the total expected reward for the policies across various values of the signal, while holding the maximum utilization at 0.5. In this case study, the critical threshold of the signal is 0.5: when the priority 2 patient's signal is higher than 0.5, the optimal policy is identical to the static heuristic policy 2. In other words, if a priority 2 patient is more likely to be an advanced patient than a basic patient, it is optimal to send an ALS NTV to the patient.

Figure 4.10 illustrates how the optimal policy serves priority 2 patients. When the signal exceeds 0.5, the fraction of patients that receive ALS service slowly decreases as the signal increases. This appears to be counter-intuitive, because we already observed that the optimal policy is identical to the static heuristic policy when the signal exceeds 0.5, and hence, the optimal policy sends ALS NTVs as frequently as possible. One proper explanation is that when the signal is higher, ALS NTVs responding to priority 2 calls enter the hospital transport phase more frequently, and therefore, ALS NTVs are occupied longer. As a result, when the

Figure 4.9: Normalized total expected reward for the NTV model

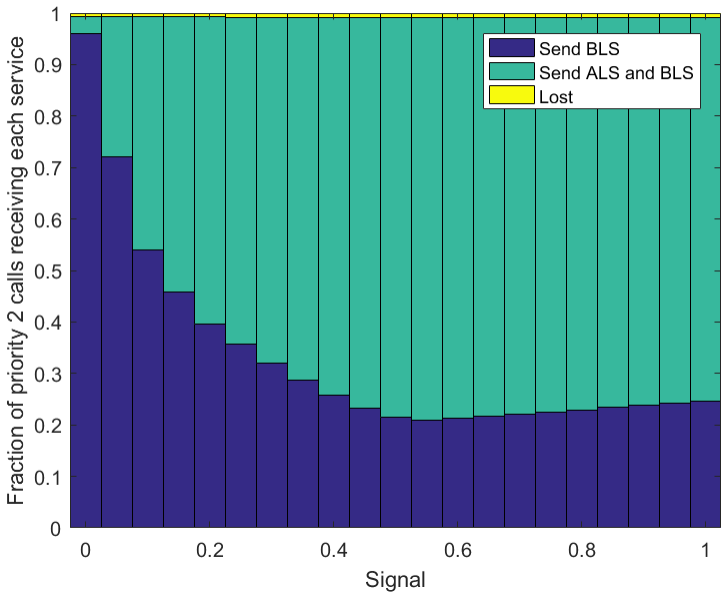


(a) Maximum Utilization



(b) Signal

Figure 4.10: Fraction of priority 2 patients receiving each service under the optimal NTV model policy



signal exceeds 0.5, fewer priority 2 patients receive ALS service when they are more likely to be advanced patients.

4.7 Conclusion

In this paper, we investigate the value of multiple response by formulating and implementing an MDP model that assigns ALS and BLS ambulances based on resource availability and future demand volume. Numerical experiments demonstrate that multiple response can significantly improve system performance . We provide two model extensions. First, we discuss an approximate spatial analogue of the base model and show how this model updates the dynamic response districts in real-time. A second extension considers an EMS system with ALS NTVs that must be accompanied by BLS ambulances for patient transport. We demonstrate through a computational study that the optimal policy approaches a static heuristic policy as priority 2 patients' signal increases.

Future research could consider providing the characteristics of the optimal policy, especially

for the model with NTVs. We propose identifying conditions on parameter values, including coverage functions and matching utilities, under which the optimal policy can be characterized as a threshold-type, control limit policy. Another direction for future research is to study a model in which some patients are not transported to the hospital. Specifically, it would be interesting to consider the setting in which the probability that a patient is transported to the hospital is not a fixed value, but instead depends on the information available to the dispatcher prior to dispatch—including patient priorities or primary symptoms reported by the caller.

5 A STOCHASTIC PROGRAMMING APPROACH FOR LOCATING AND ROUTING TWO TYPES OF AMBULANCES

5.1 Introduction

Emergency medical services (EMS) systems provide pre-hospital care to those who are in need of urgent medical treatment and transport these patients to hospitals for more definitive care. Emergency vehicle planning, or “ambulance planning" as it is more commonly denoted in the literature, is challenging because of the uncertainty in ambulance demand, which includes emergency calls’ arrival times, the incident locations, and the patients’ health needs. The ambulance planning problem has two levels of decision making: a strategic level that locates ambulances at bases and an operational level that routes ambulances to calls. These two levels of decisions are based on different levels of uncertainty in ambulance demand. The strategic level addresses the long-term planning of emergency vehicle deployment and occurs before the call arrival information is revealed. The operational level concerns how emergency vehicles are dispatched to calls in real-time.

We propose a new two-stage stochastic program to this ambulance planning problem. The stochastic program deploys two types of emergency vehicles in the first stage, and in the second stage, it routes these vehicles to patients as call arrival scenarios are disclosed. Typically, the strategic level and the operational level decisions are made separately in the literature, although the server locations inform how ambulances are routed to calls (Batta et al., 1989) and vice versa. To fill the gap, this paper integrates both strategic decisions and operational decisions in a single stochastic programming model. We place emphasis on the first-stage decisions, which are critical for informing deployment decisions. Dispatch decisions in the second stage are informative since we assess the value of the first-stage deployment solutions through the second-stage objective function values.

To construct the parameters for the second stage, we generate call arrival scenarios by

sampling directly from the emergency call log, instead of constructing interarrival and service time distributions. This data-driven approach is beneficial for the following two reasons. First, the data-driven approach does not make distributional assumptions nor does it assume vehicle independence. Optimization, queueing, and simulation models used to describe and predict EMS system performances often require assumptions about the behavior of vehicles or the distribution of events, such as exponentially distributed inter-arrival times or service times. However, such assumptions can be inconsistent with real-world observations. Moreover, when the system of concern involves multiple types of vehicles and uncertain demand—like tiered EMS systems—these assumptions, unless they are carefully justified, can be restrictive and limit the model's applicability. Second, the model allows for spatiotemporal correlation between emergency calls. The demand for EMS services fluctuates depending on the day of the week and the time of the day (Setzler et al., 2009). Further, Zhou et al. (2015) shows that ambulance demand often exhibits location-specific serial dependence and seasonality. Moreover, Lord and Mannering (2010) provide a review of statistical models to describe spatial and temporal elements associated with motor vehicle crash-frequency data. Our model does not lose information in the data concerning both (1) inter-dependency between consecutive call locations and (2) temporal auto-correlation between call arrival times, by sampling call arrival scenarios directly from the call logs.

A third novel feature of our model is that we consider a system with multiple types of emergency vehicles. Because patients have different health needs, many EMS systems employ a *tiered system* that are equipped with multiple types of emergency vehicles with different service capabilities. A tiered system sends advanced life support (ALS) vehicles staffed by paramedics, basic life support (BLS) vehicles staffed by emergency medical technicians, and/or other types of emergency vehicles to calls. In tiered EMS systems, the response time, as well as the type of emergency vehicle that responds to a patient, are critical to the patient's outcomes. For example, for cardiac arrest patients, the chance of survival decreases by 24% for every additional minute of delay in treatment (O'Keeffe et al., 2011); furthermore,

Bakalos et al. (2011) show that ALS care in non-trauma cardiac arrest patients increases the probability of patient survival almost by 47% compared to BLS care. To achieve both goals, emergency vehicles must be strategically deployed and routed so that they can appropriately respond to arriving emergency calls.

Tiered EMS systems often employ *multiple response* by dispatching multiple vehicles of different types to one emergency call. Multiple response allows providing faster first aid to emergency patients, since the vehicle that arrives at the scene first, regardless of its type, can start providing first aid. The vehicle that arrives later can provide additional services that the first responder is not capable of providing, such as advanced care or hospital transportation. However, multiple response has a potential cost of making more vehicles unavailable compared to sending only one vehicle (*single response*). This exhibits the trade-off between serving a current emergency call and being prepared for future calls. Therefore, the decision maker must understand the characteristics of the system and select multiple response when weighing its benefits against cost. Our model provides a decision-making framework for a tiered EMS system with consideration of multiple response which is differentiated from the literature that considers single response.

In this paper, we make the following contributions:

1. We formulate and implement an ambulance location model that locates and dispatches two types of vehicles at the same time under uncertainty in ambulance demand. Furthermore, our model allows the dispatcher to send multiple types of vehicles to a call (multiple response). We do not make distributional assumptions for call inter-arrival times and service times with a data-driven scenario-based approach.

2. We provide two extensions to the original model to consider different EMS system specifications including stochastic ambulance travel times, patient-ambulance matching utilities, and non-transport quick response vehicles.

3. We demonstrate the value of the scenario-based stochastic programming approach with case studies. We first investigate the number of scenario samples needed to obtain a stable

solution, and then we evaluate the value of the stochastic approach with a simulation study. We also present a solution scheme based on Benders cuts to solve large-scale instances in a reasonable amount of time.

The remainder of this paper is organized as follows. Section 5.2 provides a review of the related ambulance planning literature. We provide the base model formulation in Section 5.3. Two model variations are introduced in Section 5.4. We conduct a case study of the base model and provide the numerical results in Section 5.5. Section 5.6 introduces and tests a solution scheme based on Benders cuts. The paper is concluded in Section 5.7.

5.2 Related Literature

Ambulance planning problems have been widely studied in the literature, and we focus on models considering the stochastic nature of the ambulance location problem since they are the most relevant to our stochastic programming approach. Daskin (1983) introduces one of the earliest probabilistic location models which maximizes expected coverage when considering ambulance unavailability. Batta et al. (1989) extend Daskin's approach to investigate the underlying spatial queueing dynamics of the ambulances with a Hypercube queueing model. Restrepo et al. (2009) propose two models based on the Erlang loss formula that can be used to screen potential ambulance allocations. We refer the readers to the review articles by Brotcorne et al. (2003), Aringhieri et al. (2017) and Reuter-Oppermann et al. (2017) for a more extensive discussion of ambulance location models.

Relatively few papers propose ambulance location models with multiple types of vehicles. Schilling et al. (1979) develop mathematical programming models (TEAM/FLEET) that deploy equipment to maximize the fraction of demand covered by both primary and special equipment. Marianov and ReVelle (1992) extend Schilling et al.'s model to add individual and joint reliability requirements. Mandell (1998) proposes a probabilistic covering model with ALS and BLS vehicles and focuses on the ALS response time. In contrast, McLay (2009)

introduces an expected coverage model with ambulances and non-transport vehicles that serves multiple patient types, allows multiple response, and focuses on the first responder's response time in multiple response. Grannan et al. (2015) propose a binary linear programming model that locates two types of air ambulances and construct response districts in military medical evacuation systems.

Although the stochastic nature of the ambulance planning problem has been emphasized by a large number of EMS studies, it was not until recently that a scenario-based stochastic programming approach was used. A few papers develop stochastic programming models based on call arrival scenarios; their two-stage stochastic programs deploy emergency vehicles in the first stage and operate these vehicles to respond to demand in the second stage. Beraldi and Bruni (2009) and Noyan (2010) present a reliability approach by embedding probabilistic constraints. Nickel et al. (2016) investigate the ambulance location problem with the goal of minimizing the total cost while assuring a minimum coverage level. All of these papers perceive the call arrival scenario as a *bundle* of calls, which is the total number of emergency calls in each demand node during a given time period. It is assumed that a vehicle can be assigned only once during a scenario.

An alternative way to model a call arrival scenario is a consecutive *series* of calls associated with individual arrival time and service time. In this paper, we choose this approach because the series structure reserves the precedence relationship between calls and thus allows the model to send the same vehicle multiple times during a scenario without double-assigning the vehicle. While most ambulance location papers with stochastic programming models consider the bundle of calls, two notable exceptions define the scenarios as a series of calls. Naoum-Sawaya and Elhedhli (2013) formulate a two-stage stochastic optimization model that redeploys and dispatches ambulances to minimize relocation while maintaining an acceptable coverage level. Enayati et al. (2018) provide a stochastic ambulance redeployment model with workload limitations.

A two-stage stochastic program is useful for solving the ambulance planning problem

for tiered systems because of the uncertain ambulance demand and the interaction between vehicles in tiered EMS systems. However, to the best of our knowledge, the existing literature does not consider both topics simultaneously: the only exception is Boujemaa et al. (2018), who propose a two-stage stochastic programming location-allocation model for a two-tiered EMS system. There are two notable differences between their approach and ours. First, our model allows multiple response. Incorporating multiple response not only is a difference in how the service is specified, but it also results in a different integer programming model that is not totally unimodular, making the problem more difficult to solve. Second, while they use bundle-type call arrival scenarios, our model is based on series-type call arrival scenarios that capture more details of the EMS practice.

As far as the multiple types of ambulances are concerned, the optimal matching of the ambulance type to patient's health needs has been studied by several papers using Markov Decision Process (MDP) models. Chong et al. (2016) examine the value of a mixaed fleet system by constructing an MDP model that dispatches ALS and BLS ambulances to high and low priority calls. Yoon and Albert (2018a) present an MDP model that dynamically determines which types of ambulances to dispatch with only partial information on patients' health needs. Our research bridges the gap in this stream of literature by integrating the patient-ambulance matching problem with the ambulance deployment problem.

5.3 Problem Formulation

We consider an EMS system with two types of vehicle—ALS ambulances and BLS ambulances—and two call priority groups—high and low. High priority calls are assumed to be life-threatening and require ALS care, while low priority calls are considered non-life-threatening and either ALS or BLS care is sufficient. The EMS system serves a city or county network that consists of demand nodes and candidate ambulance stations. The ambulance planning problem concerns two types of interdependent decisions. At a strategic level, ALS and BLS

ambulances are deployed to ambulance stations. Then at an operational level, when an emergency call arrives, the dispatcher determines the type(s) of vehicles that respond, based on the call priority, the demand node in which the call is located, and the current availability of all vehicles. As far as the vehicle type is concerned, the dispatcher can choose to send an ALS ambulance, a BLS ambulance, or both (a BLS ambulance as a first responder, followed by an ALS ambulance as a backup responder).

The goal of the deployment and dispatch decision is to provide prompt and proper service to arriving emergency calls. The service is prompt if the first responder arrives at the incident scene within a pre-defined response time threshold (RTT) from the dispatch. The service is proper for high priority calls if the system provides advanced care, by either sending an ALS ambulance or both an ALS ambulance and a BLS ambulance. In contrast, any type of vehicle can provide proper service to low priority calls.

When sending the vehicle to an emergency call, we consider three levels of service based on the response time and the type of service. First, a call is *covered* if it is served by a proper type of vehicle within the RTT, e.g., nine minutes. We assume deterministic travel times between the ambulance station and incident location. Specifically, given a call c in scenario s, we define a fixed set of ambulance stations I_c^s that are close enough to call c's location to respond within the RTT. Second, a call is *served* but not covered if at least one vehicle is dispatched to the call, but either the type of service is improper for the call's priority or the response time exceeds the RTT. Finally, a call is *lost* if no vehicle is dispatched to the call. These lost calls are assumed to be served by external resources, such as neighboring EMS departments or private ambulance companies, through a process called "mutual aid." We add the reward to the system's objective function value when a call is covered and subtract the penalty from the objective function value when a call is lost. Serving a call does not add or subtract from the objective value. We enforce a penalty cost to the objective function value for lost calls to restrict the risk of not satisfying the ambulance demand, as an alternative to a chance-constrained model as in Noyan (2010) and Beraldi and Bruni (2009).

We construct a two-stage stochastic programming model that deploys vehicles to stations in the first stage and dispatches them to calls in the second stage, with the goal of maximizing the objective function value. Parameters and decision variables used for the model are summarized in Tables 5.1 and 5.2. The mathematical formulation for the scenario-based ambulance location for two types of vehicles (SALT) model is presented below:

Table 5.1: Parameters

Indices and Parameters	
I	set of stations.
$\{A, B\}$	set of vehicle types, where A is ALS ambulance and B is BLS ambulance.
S	set of scenarios.
C^s	list of calls in scenario s.
$C_H^s(C_L^s)$	set of high(low) priority calls in scenario s.
B_{ic}^s	set of calls in scenario s such that if an ambulance from station i serves a call $c' \in B_{ic}^s$, then it is still serving c' when c arrives and hence it cannot be dispatched to c.
I_c^s	set of stations that are close enough to cover call c in scenario s.
$w_H(w_L)$	gain from covering high (low) priority calls.
ρ	penalty from "lost" calls.
n_p	fleet size of type $p \in \{A, B\}$ vehicles.

Table 5.2: Decision Variables

First stage variables (Integer)	
x_{ip}	= the number of type $p \in \{A, B\}$ ambulances located at station $i \in I$.
Second stage variables (Binary)	
y_{ipc}^s	= 1 if a type p ambulance is dispatched from station i to serve call c in scenario s.
z_c^s	= 1 if call c is covered in scenario s.
z_{cA}^s	= 1 if a high priority call c is covered by a nearby ALS ambulance in scenario s.
z_{cB}^s	= 1 if a high priority call c is covered by a nearby BLS and an ALS ambulance in scenario s.
γ_c^s	= 1 if no ambulance is dispatched to a call c in scenario s.

$$z = \max_x \mathbb{E}_s[Q_s(x)] \tag{5.1}$$

$$\text{s.t.} \quad \sum_{i \in I} x_{ip} \leq n_p, \quad \forall p \in \{A, B\}, \tag{5.2}$$

where $Q_s(x) = \max_{y,z,\gamma} \sum_{c\in C_H^s} w_H z_c + \sum_{c\in C_L^s} w_L z_c - \rho \sum_{c\in C^s} \gamma_c$,

$$\text{s.t.} \quad y_{ipc} + \sum_{c'\in B_{ic}^s} y_{ipc'} \le x_{ip}, \quad \forall p \in \{A,B\}, i \in I, c \in C^s, \tag{5.3}$$

$$\sum_{p\in\{A,B\}} \sum_{i\in I} y_{ipc} + \gamma_c \ge 1, \quad \forall c \in C^s, \tag{5.4}$$

$$z_c \le \sum_{i\in I_c^s} \sum_{p\in\{A,B\}} y_{ipc}, \quad c \in C_L^s, \tag{5.5}$$

$$z_c \le z_{Ac} + z_{Bc}, \quad \forall c \in C^s, \tag{5.6}$$

$$z_{Ac} \le \sum_{i\in I_c^s} y_{iAc}, \quad \forall c \in C_H^s, \tag{5.7}$$

$$z_{Bc} \le \sum_{i\in I_c^s} y_{iBc}, \quad \forall c \in C_H^s, \tag{5.8}$$

$$z_{Bc} \le \sum_{i\in I\setminus I_c^s} y_{iAc}, \quad \forall c \in C_H^s. \tag{5.9}$$

The objective is to maximize the weighted number of covered calls and is penalized by the number of lost calls as stated in the equation (5.1). In the first-stage problem (FSP), constraint set (5.2) states that the total number of deployed vehicles for each type does not exceed the fleet size. Constraints (5.3) – (5.9) describe the second-stage problem (SSP). At any time, the total number of dispatched type p vehicles from station i cannot exceed the number of type p vehicles located at station i, which is enforced by (5.3). Constraint set (5.4) ensures that if no ambulance is dispatched to an arriving call, it is counted as a lost call and the objective is penalized. The service requirement differs by call type. Low priority calls can be covered by a nearby vehicle of any type as described in (5.5). Constraint set (5.6) states that high priority calls can be covered by either the first or the second option: first, as enforced by (5.7), high priority calls can be served by an ALS ambulance that is located nearby; second, as described in constraint set (5.8) and (5.9), high priority calls can be served by a nearby BLS ambulance and another ALS ambulance that is not necessarily

close. Note that in (5.9), we sum over $I \setminus I_c^s$ instead of I. We do so to improve the formulation; specifically, if there is an optimal solution with $y_{iAc} = 1$ for $i \in I_c^s$, then the value of z_{Bc} does not affect the objective value because $z_{Ac} = 1$ is guaranteed.

The proposed SSP formulation assumes an omniscient dispatcher who possesses perfect information about when and where emergency calls will occur during a scenario. Therefore, the second stage provides an upper estimate of the coverage that would be realized in practice. To evaluate the system performance of the first-stage solution without over-informing the dispatcher, we conduct a simulation study in Section 5.5.

The number of the possible realization of call scenarios is unbounded, and therefore, solving (5.1) – (5.9) is computationally intractable. We accordingly solve an approximate problem obtained through sampling based on the sample average approximation (SAA) (Robbins and Monro, 1951). In the SAA deterministic equivalent formulation (DEF) for the SALT model, the objective function (5.1) is replaced with the following sampling-based objective function:

$$z \approx \max_{x,y,z,\gamma} \frac{1}{|S|} \sum_{s \in S} \Big(\sum_{c \in C_H^s} w_H z_c^s + \sum_{c \in C_L^s} w_L z_c^s - \rho \sum_{c \in C^s} \gamma_c^s \Big), \tag{5.10}$$

where S describes a finite set of scenarios sampled from the data with an equal probability of occurrence. We discuss the choice of $|S|$ in Section 5.5 with a case study. We propose a solution scheme to solve large-scale problem instances in Section 5.6.

5.4 Model Extensions

This section provides two extensions of the SALT model to consider different service specifications. The first extension considers stochastic travel times from ambulance stations to incident scenes and patient-ambulance matching utility based on the appropriateness of server types to patients' health needs. The second extension considers non-transport vehicles that can improve response times but cannot transport patients to hospitals.

The SALT model With Stochastic Response Time and Matching Utility

The SALT model introduced earlier assumes a deterministic travel time when an ambulance responds to patient, under which only a subset of ambulances that are close to the incident location (I_c^s) can cover a call. In the same vein, only the ambulance of the type that exactly matches the patient's health needs can cover the patient with the reward of 1.0.

As an alternative, this extension model considers stochastic ambulance travel times and patient-ambulance matching utilities between zero and one. First, explicitly modeling uncertainty in ambulance travel times makes the model more realistic (Ingolfsson et al., 2008). Instead of a fixed set of stations that can respond to a call c within the RTT (I_c^s), we implement probabilistic coverage as a monotone decreasing function of the distance between the ambulance station and patient location (and possibly other factors). For a given station i and call c, the promptness of the service is evaluated by the probability that a server located at station i can respond to a call c in priority h RTT (R_{ic}^h). We refer the reader to Goldberg and Paz (1991) and Ingolfsson et al. (2008) for more discussion on uncertain ambulance travel times. Second, rather than providing a positive reward only when a proper service is provided, this extension considers additional patient-ambulance matching utility parameter U_{hp}, which is the reward from matching a type p ambulance to a priority h patient.

This stochastic extension is useful, since it encourages decisions that are disincentivized in the original SALT model although they are beneficial to the patient outcomes in practice. While the original model's objective function only considers whether a call is served by an ambulance located within the RTT or not, this stochastic extension imposes incentives to send a vehicle that can respond as promptly as possible. The same principle applies to patient-ambulance matching. The stochastic extension is more flexible in determining how appropriate a vehicle type is to the patient's health needs.

Under this new perception of coverage, any ambulance, even the ones that are distant from the incident location or are not the type of vehicle call requests can cover the call and

contribute to the objective function value. Instead of fixing a set of ambulances that can cover a call as in the SALT model, the new model should add the according reward to the objective function value based on the travel time of the ambulance that covers the call and its matching utility to the patient's health needs. Therefore, this necessitates a significant change in the model by including new parameters, variables, and constraints. The following equations (5.11) – (5.21) present the formulation of the extension model. The model is a two-stage stochastic programming model solving deployment and dispatching problem like the original SALT model. However, it maximizes expected coverage over all emergency calls instead of maximizing the number of promptly and properly covered patients. Table 5.3 defines the parameter and variable sets used in the model in addition to Table 5.1 and 5.2.

Table 5.3: Additional Parameters and Variables

Indices and Parameters	
R_{ic}^{h}	Probability that a server from station i can reach call c in priority h RTT
U_{hp}	Utility of responding to a priority h call with a type p vehicle.
Second Stage Variables (Binary)	
z_{iAc}	=1 if an ALS vehicle responses to call c from station i.
z_{iBc}	=1 if a BLS vehicle responses to call c from station i.
z_{iABc}	=1 if double response (a BLS first responder from station i and an ALS backup responder) is provided to call c

$$\max_{x,y,z} \sum_{s} \frac{1}{|S|} \Big(\sum_{c \in C_H^s} w_H \sum_{i \in I} R_{ic}^{H} (U_{HA} z_{iAc}^{s} + U_{HB} z_{iBc}^{s} + U_{HA} z_{iABc}^{s}) + \sum_{c \in C_L^s} w_L \sum_{i \in I} R_{ic}^{L} (U_{LA} y_{iAc}^{s} + U_{LB} y_{iBc}^{s}) - \rho \sum_{c \in C^s} \gamma_c^s \Big) \tag{5.11}$$

$$\text{s.t.} \quad \sum_{i \in I} x_{ip} \leq n_p, \quad \forall p \in \{A, B\}, \tag{5.12}$$

$$y_{ipc}^{s} + \sum_{c' \in B_{ic}^{s}} y_{ipc'}^{s} \leq x_{ip}, \quad \forall p \in \{A, B\}, i \in I, c \in C^s, s \in S, \tag{5.13}$$

$$\sum_{p\in\{A,B\}}\sum_{i\in I} y^s_{ipc} + \gamma^s_c \geq 1, \quad \forall c \in C^s, s \in S, \tag{5.14}$$

$$\sum_{p\in\{A,B\}}\sum_{i\in I} y^s_{ipc} \leq 1, \quad \forall c \in C^s_H, s \in S, \tag{5.15}$$

$$\sum_{i\in I} y^s_{ipc} \leq 1, \quad \forall p \in \{A,B\}, c \in C^s_L, s \in S \tag{5.16}$$

$$\sum_{i\in I}(z^s_{iAc} + z^s_{iBc} + z^s_{iABc}) \leq 1, \quad \forall p \in \{A,B\}, c \in C^s_H, s \in S, \tag{5.17}$$

$$z^s_{iAc} \leq y^s_{iAc}, \quad \forall i \in I, c \in C^s_H, s \in S. \tag{5.18}$$

$$z^s_{iBc} \leq y^s_{iBc}, \quad \forall i \in I, c \in C^s_H, s \in S. \tag{5.19}$$

$$z^s_{iABc} \leq \sum_{i\in I} y^s_{iAc}, \quad \forall i \in I, c \in C^s_H, s \in S, \tag{5.20}$$

$$z^s_{iABc} \leq y^s_{iBc}, \quad \forall i \in I, c \in C^s_H, s \in S. \tag{5.21}$$

The objective function (5.11) describes total expected coverage weighted by the call priority, the matching utility, and the probability of reaching to the call within the RTT. Constraint set (5.12) is the FSP, and the rest of the constraints describe the SSPs. Constraint sets (5.12) – (5.14) are analogous to the original SALT formulation. At most one server of each type can be dispatched to a high priority call as captured in (5.15); in contrast, at most one server can be dispatched to a low priority call, as captured in (5.16). Constraint set (5.17) ensures that a high priority call is served by one of the three options: by an ALS ambulance, which is captured by (5.18); a BLS ambulance which is captured by (5.19); or both an ALS ambulance and a BLS ambulance, as described by (5.20) and (5.21). Note that in the last case, we determine the matching utility based on the ALS ambulance because advanced care is provided ultimately, although the BLS ambulance that provides the first-aid and determines the response time. Note that if we set the parameters R^p_{ic} and U_{hp} as follows, then the result from this extension model simplifies to the original SALT model: $R^p_{ic} = 1$ if the station $i \in I^s_c$ when call c is located at j, and 0 otherwise; and $U_{HA} = U_{LA} = U_{LB} = 1$ and $U_{HB} = 0$.

The SALT model With NTVs

Many EMS systems are equipped with NTVs that can reach incident location faster than ambulances. However, NTVs cannot transport patients to hospitals. Therefore, if a NTV responds to an emergency call, a BLS also must be sent to transport the patient to the hospital. This difference in the dispatch protocol motivates us to present an extension of the SALT model that can accommodate NTVs.

We consider an EMS system equipped with n_A ALS NTVs and n_B BLS ambulances. Because NTVs can access to incident locations more rapidly than ambulances, we allow a longer distance threshold for NTVs. In this respect, we additionally define a parameter $I_c^{s\prime}$, which is a set of station that is within the NTV's distance threshold (longer than the ambulance's distance threshold) from the incident location of call $c \in C^s$. Therefore, $I_c^s \subset I_c^{s\prime}$. Then, we replace (5.4) – (5.9) with constraints (5.24) – (5.27) to construct the following NTV model:

$$\max_{x,y,z} \frac{1}{|S|} \sum_{s\in S} \Big(\sum_{c\in C_H^s} w_H z_c^s + \sum_{c\in C_L^s} w_L z_c^s - \rho \sum_{c\in C^s} \gamma_c^s \Big), \tag{5.22}$$

$$\text{s.t.} \qquad (5.29), \tag{5.23}$$

$$\sum_{i\in I} y_{iBc} + \gamma_c \geq 1, \quad \forall c \in C^s, \tag{5.24}$$

$$z_c \leq \sum_{i\in I_c^s} y_{iBc}, \quad c \in C_L^s, \tag{5.25}$$

$$z_c \leq \sum_{i\in I} y_{iBc}, \quad \forall c \in C_H^s, \tag{5.26}$$

$$z_c \leq \sum_{i\in I_c^{s\prime}} y_{iAc}, \quad \forall c \in C_H^s. \tag{5.27}$$

The objective function (5.22) and the first-stage constraint set (5.23) are identical to the SALT model. A call is served if there is a BLS ambulance responding to a call, as described in constraint set (5.24); constraint set (5.25) means that a low priority call can be covered

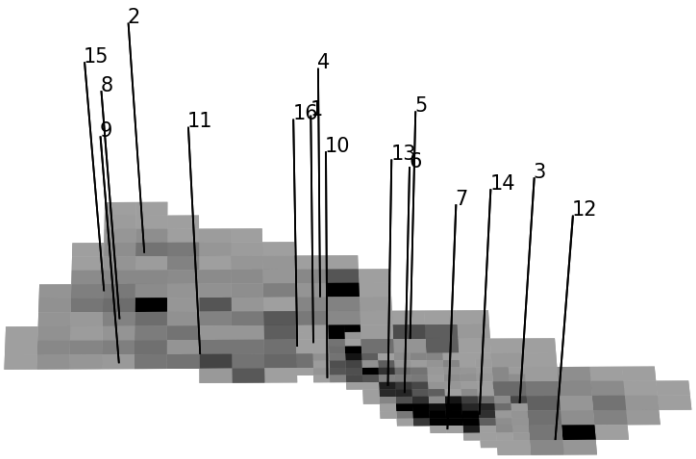


Figure 5.1: Caption

by a BLS ambulance located within the ambulance distance threshold; and constraint sets (5.26) and (5.27) indicate that a high priority call can be covered by an ALS NTV located within the NTV's distance threshold and a BLS ambulance.

5.5 Case Study

Setup

We conduct a case study for the SALT model in Section 5.3 using 911 emergency call logs from Hanover County, Virginia. The dataset includes 33,810 call records collected over 31 months during 2009 – 2011. Each record contains information regarding the call's arrival time, incident location, priority, and service times (response times, on-scene times, and hospital transport times). We divide and aggregate the geographic region into $|J| = 270$ nodes that consists of 139 2×2 mile cells (regions with lower demand) and 131 1×1 mile cells (regions with higher demand). There are $|I| = 16$ nodes that are potential candidates for ambulance stations. Figure 5.1 shows arrival rates for each demand nodes (darker cell colors represent higher demand) and the locations of 16 candidate stations.

We solve the SALT model for a shift on weekdays, from 12 PM to 6 PM, with $n_A = 3$ ALS ambulances and $n_B = 3$ BLS ambulances. Our call arrival scenario represents a consecutive

series of emergency calls. We sample a scenario from 674 weekdays according to a discrete uniform distribution. Once a random day is selected, all calls arrived between 12 PM and 6 PM on that day are considered as a consecutive series of calls in the scenario. Because the calls are sampled from the data based on a fixed length of time horizon (6 hours), the number of consecutive calls in scenarios are different. On average, the number of consecutive calls in a scenario is 9.25.

In addition to sampling calls from weekdays 12 PM – 6 PM, we also sample calls from weekdays 6 AM – 12 PM to make our implementation of the dispatching problem even more realistic with a warm-start. These "initialization" calls are considered as if they arrive exactly at the beginning of the time horizon (12 PM) with reduced service times. Specifically, if a call c arrives at t_1 has t_2 response time, t_3 on-scene time, and t_4 transport time, and is not completed until T =12 PM, then the call c need to be handed over to a server in the next shift which starts from 12 PM, to finish the remaining service. Therefore, we consider the call c as if it arrives at T with the remaining service time as $t_i = \max\{t_{ic}, T - t_1 - t_2 - t_3\} + t_4$ if served by an ambulance from station i where t_{ic} is the travel time from station i to call c's location. As a result of this initialization procedure, the SALT model dispatches some vehicles immediately at the start of the time horizon to calls already in progress, which is more realistic because in practice the EMS system does not always start a shift with all vehicles idle. Upon this warm-start procedure, the number of initialization calls is 0.76 on average. The SALT model was implemented in Python 2.7.14 and solved using Gurobi 7.5.2.

Lastly, we note that the aforementioned data-driven sampling is not a requirement for the proposed SALT formulation. For instance, one may only know summary statistics such as hourly arrival rate without the access to the raw call log data; otherwise, the dataset might only represent the history and does not predict future call arrivals precisely. In these cases, one can choose to fit distributions using summary statistics and sample from the constructed distributions to generate call arrival scenarios as an alternative to the direct data-driven sampling approach we present and solve the SALT model without further modification.

Determining the number of scenarios

When we solve the SALT model based on the SAA, the solution quality depends on the number of scenario samples. However, stochastic programming models are computationally intractable when the number of scenarios is extremely large. The choice of the number of scenarios is not trivial due to this tradeoff between the precision of the approximation and the computational tractability. Therefore, we conduct a statistical analysis of the bounds of the objective value across various values of the number of scenarios. Algorithm 1 describes a standard procedure to estimate confidence intervals (CIs) for the lower bound, upper bound, and SAA optimality gap of a given solution (Kleywegt et al., 2002). We use $M = 15$ and $K = 500$.

$\|S\|$	upper bound		lower bound		optimality gap	
	point est	CI	point est	CI	point est	CI
50	0.8093	(0.8051, 0.8135)	0.7905	(0.7642, 0.8169)	0.0205	(0, 0.0410)
100	0.8008	(0.7979, 0.8036)	0.7911	(0.7648, 0.8174)	0.0113	(0, 0.0225)
200	0.8001	(0.7979, 0.8023)	0.7875	(0.7606, 0.8143)	0.0020	(0, 0.0079)

Table 5.4: SAA results (95% CIs)

Table 5.4 presents point estimates and 95% CIs for the objective value's upper bound, lower bound, and SAA optimality gap, for different number of scenarios. As the number of scenarios increases from 50 to 200, the point estimate for the optimality gap decreases as the number of scenario increases. When $|S|$ is as large as 200, the upper bound and lower bound CIs overlap and the CI for the optimality gap falls within 1%. Based on this statistical results, for the rest of Section 5.5, we use $|S| = 200$ scenarios for all computational experiments.

Value of Stochastic Solution

We conduct a simulation study to evaluate the value of stochastic programming approach. Using simulation has two advantages over directly comparing the objective function values. First, the SALT model assumes an omniscient dispatcher who determines dispatch decisions based on the knowledge of the arrival information for all calls in a scenario, which deviates

Algorithm 1 SAA algorithm for a given value of $|S|$

Estimating the Upper Bound:
for $m = 1, ..., M$ **do**
Create a set of new scenarios S.
Solve the SALT model with S, denote its objective value as ζ_m, and first-stage solution as χ_m.
end for
Let $U_M := \frac{1}{M}\sum_{m=1}^{M}\zeta_m$, and $s_U(M) := \frac{1}{M-1}\sum_{m=1}^{M}(\zeta_m - U_M)^2$. Then an approximate $(1-\alpha)$-CI for the upper bound is

$$\left[U_M - \frac{t_{M-1,\frac{\alpha}{2}}s_U(M)}{\sqrt{M}}, U_M + \frac{t_{M-1,\frac{\alpha}{2}}s_U(M)}{\sqrt{M}}\right].$$

Estimating the Lower Bound:
for $k = 1, ..., K$ **do**
Create a scenario s.
Solve a SSP with the first-stage solution from the last batch of the upper bound estimating procedure χ_M: and denote its objective value as $\max Q_s(\chi_M) := q_k$.
end for
Let $L_K := \frac{1}{K}\sum_{k=1}^{K}q_k$, and $s_L(K) := \frac{1}{K-1}\sum_{k=1}^{K}(q_k - L_K)^2$. Then an approximate $(1-\alpha)$-CI for the lower bound is

$$\left[L_K - \frac{t_{K-1,\frac{\alpha}{2}}s_L(K)}{\sqrt{K}}, L_K + \frac{t_{K-1,\frac{\alpha}{2}}s_L(K)}{\sqrt{K}}.\right]$$

Estimating the Optimality Gap:
Create a set of new scenarios S.
Solve the SALT model with S, denote its first-stage solution as χ^*.
for $m = 1, ..., M$ **do**
Create a set of $|S|$ new scenarios S.
Solve the SALT model, denote its objective as ζ_m^*.
for $s \in S$ **do**
Solve a SSP with the first-stage solution χ^*: $\max Q_s(\chi^*)$, denote its objective value as q_s.
end for
Denote g_m be the gap between ζ_m^* and $\sum_s q_s$.
end for
Estimate the optimality gap CI from $g_1, ..., g_M$. Let $G_M = \frac{1}{M}\sum_{m=1}^{M}g_m$ and $s_G(M) = \frac{1}{M-1}\sum_{m=1}^{M}(g_m - G_M)^2$. Then an one-sided $(1-\alpha)$ CI for the SAA optimality gap is

$$\left[0, G_M + \frac{t_{M-1,\alpha}s_G(M)}{\sqrt{M}}\right].$$

from reality. The SALT objective function value overestimates and has to be reevaluated to provide the expected coverage observed in practice based on the given deployment solution. Second, although the SALT objective function value converges with probability one when we use SAA and increase the number of scenarios (Kleywegt et al., 2002), the deployment solution is not guaranteed to converge. Therefore, by solving the SALT model multiple times with different random scenario sets, we obtain multiple optimal solutions that must be further evaluated.

Typically, the value of the stochastic solution (VSS) is evaluated by constructing a "mean value" problem, obtaining a mean value solution, and then comparing the performance of mean value solution and the stochastic solution under the same set of scenarios (Birge, 1982). However, in our problem setting, the mean value problem is not easily defined, since there exists no average call arrival scenario. Therefore, we construct a deterministic optimization model that locates two types of vehicles as a benchmark, and then we compare the performance of the optimal solutions from the SALT model to the performance of the benchmark solution.

We construct a deterministic ambulance location problem to create a benchmark solution. Moore and ReVelle (1982) and Mandell (1998) provide hierarchical service location models that are relevant to our problem setting. In particular, two service types are defined in their model: a lower level service is available if both lower and higher type facilities are present within an appropriate distance; and a higher level service is available if a higher type facility is accessible within an appropriate distance. Considering these papers as a baseline, we define the maximal covering location problem for prioritized calls and two types of vehicles (MCLPP2). In MCLPP2, λ_j^p is the hourly priority p call arrival rate at demand node j estimated from the same call log we sample call arrival scenarios for the SALT model, I_j is the set of stations located within the RTT from node j. The decision variables are as follows: x_i^q is the number of type q vehicles deployed to station i; h_j (ℓ_j) is a binary variable that equals 1 if high (low) priority calls in node j are covered; h_j^1 (h_j^2) is a binary variable that

equals 1 if high priority calls in node j are covered by the first (second) service option.

$$\max \quad \sum_j w_H \lambda_j^h h_j + w_L \sum_j \lambda_j^l \ell_j \tag{5.28}$$

$$\text{s.t.} \quad \sum_{i \in I} x_i^q \leq n_p \qquad \forall q \in \{A, B\}, \tag{5.29}$$

$$\sum_{i \in N_j} x_i^A \geq h_j^2 \qquad \forall j \in J, \tag{5.30}$$

$$\sum_{i \in I} x_i^A \geq h_j^1 \qquad \forall j \in J, \tag{5.31}$$

$$\sum_{i \in N_j} x_i^B \geq h_j^1 \qquad \forall j \in J, \tag{5.32}$$

$$h_j^1 + h_j^2 \geq h_j \qquad \forall j \in J, \tag{5.33}$$

$$\sum_{i \in I_j} \sum_{q \in \{A,B\}} x_i^q \geq \ell_j \qquad \forall j \in J, \tag{5.34}$$

$$x_i^A, x_i^B \in \{0, 1\} \qquad \forall i \in I, \tag{5.35}$$

$$h_j^1, h_j^2, h_j, \ell_j \in \{0, 1\} \qquad \forall j \in J, \tag{5.36}$$

The objective is to maximize the demand-weighted number of covered calls as defined in (5.28). The number of vehicles deployed to the station cannot exceed the fleet size as stated in (5.29). Constraint set (5.30) describes one option to cover a high priority call, which is to have an ALS ambulance located close to the call. Another option, which is described in (5.31) and (5.32), is a combination of a BLS ambulance located close to the call and an ALS ambulance in any location. Constraint set (5.33) states that either the first or the second option is sufficient to cover a high priority call. For low priority calls, any type of ambulance located nearby can cover the low priority call, which is enforced by (5.34).

To obtain stochastic solutions, we repeatedly solve the SALT model with different random sets of scenarios until we pick six distinct first-stage solutions. We denote the resulting six distinct first-stage solutions as SALT1 — SALT6. These stochastic solutions and a benchmark

deterministic solution can be found in Table 5.5.

Model solution	ALS stations	BLS stations
MCLPP2	1, 6, 11	12, 14, 15
SALT1	1, 6 ,14	6, 11, 16
SALT2	1, 6, 14	6, 8, 16
SALT3	1, 6, 14	4, 6, 16
SALT4	1, 6, 8	6, 14, 16
SALT5	7, 10, 14	6, 8, 16
SALT6	1, 6, 14	6, 15, 16

Table 5.5: Optimal deployment solutions for a deterministic benchmark and six stochastic solutions

We run a discrete event simulation with each of seven deployment solutions (a deterministic benchmark solution and six stochastic solutions) in Table 5.5. The simulation was implemented in Python 2.7.14. Each run of the simulation contains 200 scenarios as in the SALT model, and the simulation is repeated 30 times to construct the 95% confidence interval of the weighted total expected patient coverage. We reiterate that one of the purposes of the simulation study is to assess the first-stage deployment decisions without assuming an omniscient dispatcher who possesses the call arrival information of the entire scenario, as in the SALT model's SSP. This is beneficial since the simulation results reflect the practical performance of the deployment decisions. In this respect, the simulation is based on a myopic dispatcher who sends the closest appropriate vehicle. Algorithm 2 formalizes the simulation's dispatching logic.

Figure 5.2 presents 95% confidence intervals for the weighted total expected coverage obtained from simulations with seven different deployment solutions. It is notable that all six stochastic solutions achieve significantly improved expected coverage compared to the benchmark static solution. Although the SALT model returns different first-stage solutions based on the choice of random samples even when the scenario size is large enough ($|S|$=200), our simulation study exhibits that all of these deployment solutions perform well in terms of the expected patient coverage.

Algorithm 2 Dispatching algorithm in the simulation

```
if all vehicles are busy then
  return NULL
end if
if c.priority=H then
  if there is an idle ALS ambulance a within RTT from c then
    a ← the closest idle ALS ambulance
    return a
  else if there is an idle BLS ambulance a within RTT from c and an idle ALS ambulance
  a′ then
    a ← the closest idle BLS ambulance, a′ ← the closest idle ALS ambulance
    return a, a′
  else if there is an idle ALS ambulance a then
    a ← the closest idle ALS ambulance
    return a
  else if can find an idle BLS ambulance a then
    a ← the closest idle BLS ambulance
    return a
  end if
else
  a ← closest idle ambulance
  return a
end if
```

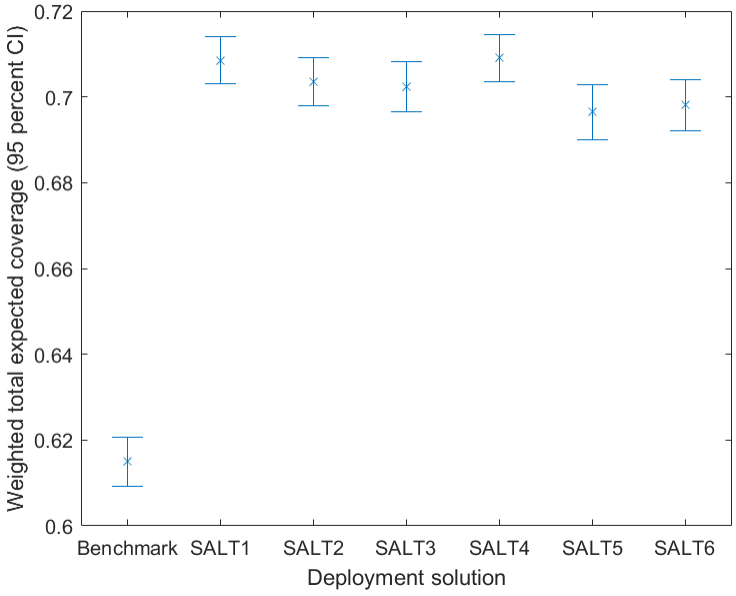


Figure 5.2: Simulation results for a deterministic benchmark and six stochastic solutions

5.6 Solution Method for Large-scale Problem Instances

Stochastic programming models are computationally challenging when the size of the instance is large. As the number of scenarios, the number of consecutive calls in a scenario, the fleet size, or the size of the network increases, the SALT model becomes computationally intensive to solve. The numerical experiments in Section 5.5 do not incur any computational challenge, however, the SALT model is more useful when it can be implemented for a broader class of real-world datasets. This motivates us to present a solution scheme for large-scale problem instances.

In this section, we propose a solution method that is based on Benders cuts to improve solution times for the SALT model. We linearize the SSP and introduce a Benders cut generating procedure that is integrated into the branch-and-cut procedure to solve the integer FSP. Then, we demonstrate the computational performance of this solution method with a case study.

Branch-and-Benders-cut

We linearize second-stage variables as a simple way to resolve computational intractability, recognizing that the SALT model informs deployment solutions (first-stage variables). However, unlike similar problems in the literature such as Peng et al. (2018) and Boujemaa et al. (2018), the matrix defining the feasible region of our SSP is not totally unimodular. Therefore, relaxing the integrality constraints in the SSP comes at a cost, which should be compared to the benefit from the computational time reduction. In this respect, we additionally propose to implement Benders decomposition to take advantage of the linear SSP Benders (1962).

Benders decomposition has been used in a large number of applications to improve computational tractability. The core idea is to decompose the deterministic equivalent problem into $|S|$ subproblems (the SSPs with linear SSP) and a master problem (the FSP with some integer variables). However, as far as the integer master problem is concerned,

repeatedly solving the integer master problem whenever a Benders cut is added can become very time-consuming, especially after more cuts are added. As an alternative, the branch-and-Benders-cut (BBC) scheme adds Benders cuts only as necessary during a single branch-and-cut process.

To make the cut generating procedure more effective, we add a time-limited phase 0 procedure at the root node. In phase 0, we consider a linear master problem where all integer first-stage variables are relaxed and repeatedly add Benders cuts until either no additional Benders cuts are generated or a preset time limit is reached. This phase 0 procedure is essential for the following two reasons. First, because generating Benders cuts by solving the LP relaxation of the master problem at the root node yields cuts faster than adding Benders cuts with integer master problem later at the node with incumbent solutions. Second, by including a number of Benders cuts as a part of the initial formulation, we can take advantage of commercial integer programming solvers' abilities, since they derive valid cuts as needed to strengthen the LP relaxation, such as mixed-integer rounding cuts (Bodur and Luedtke, 2016) or split cuts (Bodur et al., 2017). We set a time limit on phase 0 since we want to ensure that there is enough time left after phase 0 for the remaining branch-and-cut procedure.

Computational experiments

All numerical experiments in this section are based on the emergency call logs from Mecklenburg County, North Carolina. The call logs provide the calls' arrival times, travel times, service times, and geographic locations. The call logs do not provide the call's priority information, and therefore, we randomly generated priorities (high or low) according to a Bernoulli distribution. We estimate the parameter value for the Bernoulli distribution based on the Hanover County dataset (described in Section 5.5) as $P(high) = 0.2917$. Call arrival scenarios are sampled as a consecutive series of calls following a procedure identical to Section 5.5. Focusing on testing the solution method, we omit the initialization procedure for simplicity. We solve the SALT model for a shift on weekdays, 12 PM – 6 PM. On average,

the number of consecutive calls in a scenario is 60. The city network represents a service area of approximately 540 square miles that is discretized into $|J| = 168$ demand nodes. Each demand node is a 2×2-mile cell. The potential station locations are not given in the data, and therefore, we select $|I| = 30$ demand nodes as potential station locations. Considering the demand volume and the size of the network, we determine the fleet size as $n_A = 6$ ALS ambulances and $n_B = 6$ ambulances. Overall, we construct a larger problem instance with the Mecklenburg County dataset than the Hanover County dataset in Section 5.5.

$\|S\|$	original DEF	DEFL with linear SSP		BBC with linear SSP	
	opt gap (%)	opt gap (%)	lin gap (%)	opt gap (%)	lin gap (%)
10	0.35	0.00	1.10	0.41	1.10
50	12.6*	1.69	1.14	1.36	1.13
100	*	7.52	0.90	2.73	1.35
150	*	228.41	0.31	3.25	1.59
200	*	*	*	3.84	1.26

Table 5.6: Performance of the original DEF, the DEFL with linear SSP, and the proposed BBC procedure after 7200-second runtime
Note: "opt gap" is the optimality gap after 7200-second run, "lin gap" is the linear relaxation gap based on the difference between the objective value based on the linear SSP and integer SSP, "*" means that the Gurobi solver was terminated before the time limit is reached because of out-of-memory error.

Table 5.6 presents the optimality gaps and linear relaxation gaps from three solution methods: the original DEF for the SALT model (denoted as DEF), the original DEF with linear SSP (denoted as DEFL), and the BBC with linear SSP (denoted as BBC), for different number of scenarios. For all three methods, we evaluate the optimality gaps ("opt gap" in Table 5.6) achieved after 7200 CPU seconds. In particular, for the BBC, we set an additional time limit of 2500 CPU seconds for the phase 0 procedure, and assign the remaining time to branch-and-cut, so that the total runtime (both phase 0 and branch-and-cut) is bounded by 7200 CPU seconds. The optimality gap is defined as follows:

$$opt\_gap = \frac{obj\_bd - obj\_val}{obj\_val},$$

where obj_bd is the best objective bound and obj_val is the best incumbent solution found before the time limit is reached. For the DEFL and BBC which are based on the linearly relaxed SSP, we also evaluate their linear relaxation gaps ("lin gap" in Table 5.6) as follows:

$$lin\_gap = \frac{obj\_lin\_SSP - obj\_int\_SSP}{obj\_int\_SSP},$$

where obj_int_SSP (obj_lin_SSP) is the objective function of the integer (linear) SSPs when the first-stage variable values are determined by the optimal deployment solutions from the given solution method.

Table 5.6 shows that the BBC is an effective method to solve the SALT model when the size of the instance is large, in both time and memory. Relatively simple implementations (DEF, DEFL) are more effective than the BBC to find the optimal solutions only when the number of scenarios is as small as $|S| = 10$. As the number of scenarios $|S|$ increases, both the DEF and DEFL quickly become computationally intractable: DEF fails to find a feasible solution due to the out-of-memory issue after $|S|$ exceeds 50; for the same reason, the DEFL fails to find a feasible solution when $|S| = 200$. Although the DEFL performs better than the DEF by finding a feasible solution when $|S|$=50, 100, 150, its optimality gaps are significantly larger than what the BBC achieves. Additionally, we measured the optimality gaps after 3600 CPU seconds for a more time-limited setting. The BBC found feasible solutions for all instances ($|S| = 10, 50, 100, 150, 200$) within 3600 CPU seconds, whereas both DEF and DEFL only found feasible solutions for $|S| < 50$. Lastly, for all sizes of $|S|$ and for both of DEFL and BBC, the linear relaxation gaps are small enough to justify the usage of linear SSP.

5.7 Conclusion

This paper introduces the SALT model, which is a two-stage stochastic programming model that deploys and dispatches two types of emergency vehicles. We propose a data-driven

approach of sampling call arrival scenarios directly from the call logs to lift distributional assumptions and capture spatiotemporal correlation between emergency calls. We demonstrate the value of the stochastic solutions through a case study with a discrete event simulation. To enhance the generality of the model, we suggest a solution method based on Benders cuts and test the solution method on large-scale problem instances. We additionally provide two extensions to the SALT model: the first extension considers stochastic ambulance travel times and patient-ambulance matching utilities; the second extension considers NTVs.

There are several possible extensions of the SALT model. First, future research could consider reconstructing the SSP by lifting the omniscient dispatching. One direction is implementing a dispatching scheme that more closely mimics practice, for instance, sending the closest vehicle. Another direction is creating a dispatching scheme that constructs an SSP with totally unimodularity.

Moreover, it would be advantageous to study a similar model with imperfect information about the patient's health needs. This specification of the system is especially interesting regarding multiple response. Once the patient's health needs are fully disclosed after multiple response vehicles arrive at the scene, a vehicle that does not meet the patient's actual health needs can be discharged earlier and be available for other calls. Therefore, incorporating the uncertainty in patient's health needs in the SALT model will incentivize multiple response.

A APPENDIX FOR CHAPTER 2

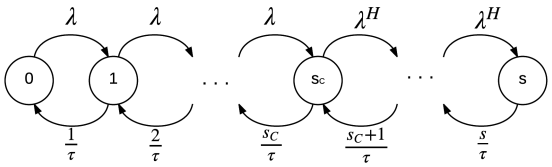
A.1 Derivation of Approximate Hypercube Model with Cutoff Priority Queue

In this appendix, we present derivation details for the approximate Hypercube model for a cutoff priority queue presented in Section 2. First, we derive closed-form expressions for the steady-state probabilities in a cutoff priority queue. The steady-state probability formulations in (2.5) and (2.6) are easily obtained from balance equations in the standard birth-death process. Figure A.1 shows the corresponding transition diagram for a cutoff priority queue systems with a zero-line queue.

To derive the steady-state probabilities formulation for the infinite capacity queue in (2.7), (2.8) and (2.9), we adapt the SCQQ model in Taylor and Templeton (1980), where both high and low priority calls are put in an infinite capacity queue when the number of busy servers is equal to or greater than the cutoff (s for high and s_C for low priority calls). In their paper they also provide the formula for the case where high priority customers are lost while only low priority customers are queued (SCQL model), which we do not consider. Therefore if one wants to apply the idea of reserving capacity only with a low priority queue, that model can be directly used in our procedure too, with updates for the steady-state probabilities P_i in STEP 1 and by setting $\nu = 1 - P_s\frac{\lambda^H}{\lambda}$ in STEP 2 (of the iterative Hypercube procedure in Section 2.2).

Next, we derive correction factors for low priority calls in the cutoff priority queue. This

Figure A.1: Transition Diagram for a Zero Capacity Queue with a Cutoff s_C.



follows the procedure in Larson (1975) closely, with some modifications due to the cutoff assumption. Suppose, for now, instead of having non-identical servers and preference lists, when a call arrives we randomly sample servers until the first free server is found, and then we dispatch that server unless it does not violate the cutoff priority scheme. Let B_k denote the event that the kth selected server is busy (not available), $F_j \equiv B_k^c$ denote the event that the kth server selected is free (available), S_i denote the state of the queueing system where exactly i servers are busy (hence $P_i = P(S_i)$), and $P\{B_1B_2\cdots B_kF_{k+1}\}$ denote the probability that the first free server is the $k+1$st sampled server.

Under the cutoff scheme, if a low priority call arrives, the probability of dispatching the $k+1^{st}$ selected server is

$$P\{B_1B_2\cdots B_kF_{k+1}\} = P\{B_1B_2\cdots B_kF_{k+1}|S_i, i < s_C\},$$

since we do not select a server to dispatch to a low priority call if the number of busy servers is equal to or greater than the cutoff. Therefore we wish to derive an expression for $P\{B_1B_2\cdots B_kF_{k+1}|S_i, i < s_C\}$ that motivates an approximation procedure for the model in which servers are not identical. Recall that in the Hypercube model under a fixed-preference dispatching policy, the dispatcher always assigns the most preferred available server. Therefore the desired probability is the probability that the first k preferred servers are busy and the $k+1^{st}$ server is free. By conditioning on the number of busy servers, we obtain

$$P\{B_1B_2\cdots B_kF_{k+1}|S_i, i < s_C\} = \sum_{i=0}^{s_C-1} P\{B_1B_2\cdots B_kF_{k+1}|S_i\}P_i. \tag{A.1}$$

where

$$\begin{aligned} &P\{B_1B_2\cdots B_kF_{k+1}|S_i\} \\ &= P\{F_{k+1}|B_1B_2\cdots B_kS_i\}P\{B_k|B_1B_2\cdots B_{k-1}S_i\}\cdots P\{B_1|S_i\}. \end{aligned} \tag{A.2}$$

Then, as in Larson (1975), we have

$$P\{B_i|B_1B_2\cdots B_{j-1}S_i\} = \frac{i-(j-1)}{s-(j-1)}, \qquad j = 1, 2, \cdots, i+1,$$
$$P\{F_{k+1}|B_1B_2\cdots B_kS_i\} = \frac{s-i}{s-k}, \qquad k = 0, 1, \cdots, i. \tag{A.3}$$

By substituting (A.2) and (A.3) into (A.1) we get the expression for desired dispatch probability as

$$\begin{aligned} P\{B_1B_2\cdots B_kF_{k+1}|S_i, i < s_C\} &= \sum_{j=k}^{s_C-1} \frac{i}{s}\frac{i-1}{s-1}\cdots\frac{i-(k-1)}{s-(k-1)}\frac{s-i}{s-k}P_i \\ &= \sum_{i=j}^{s_C-1} \frac{(s-k-1)!(s-i)i!}{(i-k)!s!}P_i \end{aligned} \tag{A.4}$$

with P_i in (2.5) and (2.6) (for the Loss model) or in (2.7), (2.8) and (2.9) (for the Queued model). We can also write another expression for (A.4) using the correction factors:

$$P\{B_1B_2\cdots B_kF_{k+1}|S_i, i < s_C\} = q_k^L r^k(1-r),$$

where

$$q_k^L = \frac{\sum_{i=k}^{s_C-1} \frac{(s-k-1)!(s-i)i!}{(i-k)!s!}P_i}{(1-P_s\frac{\lambda^H}{\lambda} - \sum_{i=s_C}^{s} P_i\frac{\lambda^L}{\lambda})^k\rho^k\big(1-(1-P_s\frac{\lambda^H}{\lambda} - \sum_{i=s_C}^{s} P_i\frac{\lambda^L}{\lambda})\rho\big)}. \tag{A.5}$$

Lastly, the derivation of q_j^H is analogous to the above derivation with s_C replaced by s.

A.2 Computational Results for Large-scale Dataset

Here we provide additional computational results that correspond to Tables 2.2 – 2.4 for the Scaled-up dataset. Tables A.1 and A.2 display dispatch probabilities for the Scaled-up dataset with 16 servers for selected number of s_R. The tables are truncated for $k \geq 5$, since

Table A.1: High Priority Dispatch Probabilities f_k^H and Loss Probabilities for High Priority Calls $P(\text{lost}|H)$ for Different Values of s_R in the Scaled-up Hanover County Dataset.

	Hypercube					Simulation				
	$s_R = 0$	$s_R = 4$	$s_R = 8$	$s_R = 12$	$s_R = 15$	$s_R = 0$	$s_R = 4$	$s_R = 8$	$s_R = 12$	$s_R = 15$
$k = 1$	0.3685	0.4032	0.4945	0.5825	0.6096	0.3601	0.3957	0.4879	0.5802	0.6105
$k = 2$	0.2117	0.2300	0.2456	0.2295	0.2145	0.2005	0.2145	0.2206	0.2061	0.1961
$k = 3$	0.1288	0.1383	0.1317	0.1047	0.0936	0.1251	0.1314	0.1230	0.1002	0.0910
$k = 4$	0.0827	0.0878	0.0667	0.0461	0.0449	0.0834	0.0863	0.0690	0.0508	0.0478
$k = 5$	0.0533	0.0534	0.0331	0.0213	0.0208	0.0555	0.0543	0.0400	0.0278	0.0239
$P(\text{lost}\|H)$	0.0226	0.0013	0.0001	0.0000	0.0000	0.0244	0.0015	0.0001	0.0000	0.0000

Table A.2: Low Priority Dispatch Probabilities f_k^L and Loss Probabilities for Low Priority Calls $P(\text{lost}|L)$ for Different Values of s_R in the Scaled-up Hanover County Dataset.

	Hypercube					Simulation				
	$s_R = 0$	$s_R = 4$	$s_R = 8$	$s_R = 12$	$s_R = 15$	$s_R = 0$	$s_R = 4$	$s_R = 8$	$s_R = 12$	$s_R = 15$
$k = 1$	0.3939	0.3965	0.3197	0.1076	0.0043	0.3885	0.3916	0.3083	0.1027	0.0040
$k = 2$	0.2163	0.2057	0.1206	0.0175	0.0000	0.2127	0.2026	0.1189	0.0174	0.0000
$k = 3$	0.1307	0.1082	0.0442	0.0022	0.0000	0.1301	0.1076	0.0439	0.0022	0.0000
$k = 4$	0.0766	0.0573	0.0134	0.0001	0.0000	0.0770	0.0573	0.0138	0.0001	0.0000
$k = 5$	0.0489	0.0303	0.0038	0.0000	0.0000	0.0497	0.0309	0.0041	0.0000	0.0000
$P(\text{lost}\|L)$	0.0226	0.1737	0.4973	0.8725	0.9957	0.0242	0.1809	0.5098	0.8776	0.9960

Table A.3: System-wide Server Busy Probability r for Different Values of s_R in the Scaled-up Hanover County Dataset.

	Hypercube					Simulation				
	$s_R = 0$	$s_R = 4$	$s_R = 8$	$s_R = 12$	$s_R = 15$	$s_R = 0$	$s_R = 4$	$s_R = 8$	$s_R = 12$	$s_R = 15$
r	0.6119	0.5682	0.4603	0.3364	0.2950	0.6188	0.5731	0.4637	0.3394	0.3003

the values when $k \geq 5$ are very small. Table A.3 presents the server busy probabilities for the Scaled-up dataset. The left hand columns for all tables report the Hypercube model results, while the right hand columns report the simulation results. As in the original dataset (Table 2.2 – 2.4), the discrepancy between the analytical Hypercube model results and simulation results are small.

B APPENDIX FOR CHAPTER 3

B.1 Proof of Theorem 3.1 (See page 54)

Proof. Proof. Let $V_t(s,d)$ denote the conditional value function assuming that the action taken for state s and time t is d (therefore, $V_t(s) = \max_d V_t(s,d)$). We decide optimal action d_t^i by comparing the value of two different choices, $d_t^i = A$ versus $d_t^i = B$, while fixing the action chosen for other call classes. Hence, $d_t^{i*}(s^A, s^B) = A$ if and only if $V_t(s, d^{-i}, 0) > V_t(s, d^{-i}, 1)$, where $d^{-i} = (d^1, \cdots, d^{i-1}, d^{i+1}, \cdots, d^m)$. By arranging terms we get

$$
\begin{aligned}
&V_t(s, d^{-i}, A) - V_t(s, d^{-i}, B) \\
&= R(s, d^{-i}, A) - R(s, d^{-i}, B) \\
&+ \sum_{j^A, j^B} \left(P_t(j^A, j^B | s, d^{-i}, A) V_{t+1}(j^A, j^B) - P_t(j^A, j^B | s, d^{-i}, B) V_{t+1}(j^A, j^B) \right) \\
&= \frac{\lambda^i}{\Lambda} \left((\omega^i f^a(s^A) U_{aA} + (1-\omega^i) f^b(s^A) U_{bA}) - (\omega^i f^a(s^B) U_{aB} + (1-\omega^i) f^b(s^B) U_{bB}) \right) \\
&+ \frac{\lambda^i}{\Lambda} (V_{t+1}(s^A - 1, s^B) - V_{t+1}(s^A, s^B - 1)).
\end{aligned}
$$

Therefore, for any time epoch t and state $s = (s^A, s^B)$, the optimal action for a class i call is to send an ALS server (i.e. $d_t^{i*}(s^A, s^B) = A$), if and only if the following equality is true:

$$
\begin{aligned}
&\omega^i U_{aA} f^a(s^A) + (1-\omega^i) U_{bA} f^b(s^A) + V_{t+1}(s^A - 1, s^B) \\
&> \omega^i U_{aB} f^a(s^B) + (1-\omega^i) U_{bB} f^b(s^B) + V_{t+1}(s^A, s^B - 1). \qquad \text{(B.1)}
\end{aligned}
$$

The arranged terms in (B.1) do not involve d^{-i}. In addition, when we solve the problem using backward induction, value functions in (B.1) ($V_{t+1}(s)$) are all known constants. In conclusion, the choice of optimal action for all other call classes does not affect the choice of optimal action for call class i. □

B.2 Proof of Theorem 3.2 (See page 55)

Proof. Proof. From the result of class separability of optimal action, $d_t^{i*}(s^A, s^B) = A$ if and only if (B.1) is true. If values of utility parameter U_{pq} satisfies (3.1), by arranging terms (B.1) is equivalent to

$$\omega^i > \frac{U_{bB}f^b(s^B) - U_{bA}f^b(s^A) + V_{t+1}(s^A, s^B - 1) - V_{t+1}(s^A - 1, s^B)}{U_{aA}f^a(s^A) - U_{bA}f^b(s^A) - U_{aB}f^a(s^B) + U_{bB}f^b(s^B)} = \bar{\omega}_t(s).$$

□

B.3 Proof of Theorem 3.3 (See page 55)

Proof. Proof. Pick an arbitrary state s and time t such that there is at least one ALS server available, i.e. $s^A < N^A$. It is previously shown that $d_t^{i*}(s^A, s^B + 1) = A$ if and only if

$$\begin{aligned} V_{t+1}(s^A, s^B) - V_{t+1}(s^A - 1, s^B + 1) < \omega^i U_{aA}f^a(s^A) + (1 - \omega^i)U_{bA}f^b(s^A) \\ - \omega^i U_{aB}f^a(s^B + 1) - (1 - \omega^i)U_{bB}f^b(s^B + 1). \end{aligned} \tag{B.2}$$

Analogously we have $d_t^{i*}(s^A + 1, s^B + 1) = A$ if and only if

$$\begin{aligned} V_{t+1}(s^A + 1, s^B) - V_{t+1}(s^A, s^B + 1) < \omega^i U_{aA}f^a(s^A + 1) + (1 - \omega^i)U_{bA}f^b(s^A + 1) \\ - \omega^i U_{aB}f^a(s^B + 1) - (1 - \omega^i)U_{bB}f^b(s^B + 1). \end{aligned} \tag{B.3}$$

Proving that (B.2) implies (B.3) proves Theorem 3.3 after applying mathematical induction on s^A. Therefore, we seek for a sufficient condition under which (B.2) implies (B.3). One

such condition is

$$V_{t+1}(s^A+1,s^B)-V_{t+1}(s^A,s^B+1)-V_{t+1}(s^A,s^B)+V_{t+1}(s^A-1,s^B+1)\leq \tag{B.4}$$
$$\omega^i U_{aA}(f^a(s^A+1)-f^a(s^A))+(1-\omega^i)U_{bA}(f^b(s^A+1)-f^b(s^A)).$$

Note that proving (B.4) is equal to showing that $V_t(s^A,s^B,A)-V_t(s^A,s^B,B)\leq V_t(s^A+1,s^B,A)-V_t(s^A+1,s^B,B)$. We introduce Lemma B.2 and B.3 that prove (B.4) in different state space domains. We remark the following to ease the proof of Lemma B.2 and B.3.

Remark B.1. *For a given time epoch t, state s and call class i, due to the class separability proven in Theorem 3.1, whether it is optimal to send an ALS server or a BLS server to a class i call arriving in t at state s is independent of optimal actions for other call classes $i'\neq i$. Similarly, in determining whether there exists an optimal action for class i a state which is a control limit policy or not, optimal actions for other call classes $i'\neq i$ is not involved. Therefore, when proving Theorem 3.3, we can make a simplifying assumption that we only have one call class without loss of generality. For the rest of Appendix B.3, we assume that there is only one call class for the ease of notation.*

Lemma B.2. *If (B.4) holds for all pairs of (s^A,s^B) such that $1\leq s^A\leq N^A-1$ and $0\leq s^B\leq N^B-1$, then the following also holds for 'non-boundary' cases ($2\leq s^A\leq N^A-1$, $1\leq s^B\leq N^B-1$):*

$$V_t(s^A+1,s^B)-V_t(s^A,s^B+1)-V_t(s^A,s^B)+V_t(s^A-1,s^B+1)\leq \tag{B.5}$$
$$\omega^i U_{aA}(f^a(s^A+1)-f^a(s^A))+(1-\omega^i)U_{bA}(f^b(s^A+1)-f^b(s^A)).$$

Proof. Proof. Note that we are showing the proof for 'non-boundary' cases (states where there are at least 1 ALS and 1 BLS servers idle), since the system has a zero-line queue. The state transition is different for 'boundary' cases, so the proof must be done differently. Here we show for non-boundary cases first and separate the proof for boundary cases to Lemma

B.3 that follows later.

We first introduce notation in pursuit of compact expressions:

- $\mathbb{D}_t(s^A, s^B) := \big(d_t^i(s^A+1, s^B), d_t^i(s^A, s^B+1), d_t^i(s^A, s^B), d_t^i(s^A-1, s^B+1)\big)$,
- $\mathbb{V}_t(s^A, s^B) := V_t(s^A+1, s^B) - V_t(s^A, s^B+1) - V_t(s^A, s^B) + V_t(s^A-1, s^B+1)$,
- $F(s^A) := \omega^i U_{aA} f^a(s^A) + (1-\omega^i) U_{bA} f^b(s^A)$, $\mathbb{F}(s^A) = F(s^A+1) - F(s^A)$.

As a starting point, we make a simplifying assumption that $d_t^i(s^A+1, s^B) = d_t^i(s^A, s^B+1) = d_t^i(s^A, s^B) = d_t^i(s^A-1, s^B+1) = d^i \in \{A, B\}$. We lift this assumption later. With new notations, Remark B.1, and simplifying assumptions, we rewrite (B.5) for all non-boundary s^A and s^B:

$$
\begin{aligned}
&V_t(s^A+1, s^B) - V_t(s^A, s^B+1) - V_t(s^A, s^B) + V_t(s^A-1, s^B+1) \\
&= \frac{\lambda^i}{\Lambda}\big(\mathbb{1}_{d^i=A}\mathbb{V}_{t+1}(s^A-1, s^B) + \mathbb{1}_{d^i=B}\mathbb{V}_{t+1}(s^A, s^B-1)\big) && \text{(B.6)} \\
&+ \min\{N^A - s^A - 1, N^A\}\frac{\mu^A}{\Lambda}\mathbb{V}_{t+1}(s^A+1, s^B) + \min\{N^B - s^B - 1, N^B\}\frac{\mu^B}{\Lambda}\mathbb{V}_{t+1}(s^A, s^B+1) && \text{(B.7)} \\
&+ (1 - \frac{\lambda^i}{\Lambda} - \min\{N^A - s^A, N^A\}\frac{\mu^A}{\Lambda} \\
&- \min\{N^B - s^B, N^B\}\frac{\mu^B}{\Lambda})\mathbb{V}_{t+1}(s^A, s^B) && \text{(B.8)} \\
&+ \frac{\mu^B - \mu^A}{\Lambda}\big(V_{t+1}(s^A+1, s^B+1) - 2V_{t+1}(s^A, s^B+1) + V_{t+1}(s^A-1, s^B+1)\big) && \text{(B.9)} \\
&+ \frac{\lambda^i}{\Lambda}\mathbb{1}_{d^i=A}\big(\omega^i(f^a(s^A+1) - 2f^a(s^A) + f^a(s^A-1))U_{aA} \\
&+ (1-\omega^i)(f^b(s^A+1) - 2f^b(s^A) + f^b(s^A-1))U_{bA}\big), && \text{(B.10)}
\end{aligned}
$$

where (B.6) corresponds to call arrival event, (B.7) refers to service completion event, (B.8) represents dummy event (neither call arrives nor server completes service), (B.9) arranges all left terms arising from different service completion rate on different state, and (B.10) is for reward. Now we rewrite each term in more compact notation. First, by definition we have

(B.10) $= \frac{\lambda^i}{\Lambda}\big(\mathbb{F}(s^A) - \mathbb{F}(s^A - 1)\big)$. Also, by mathematical induction we have

$$
\begin{aligned}
&\text{(B.6)} \le \frac{\lambda^i}{\Lambda}\big(\mathbb{1}_{d^i=A}\mathbb{F}(s^A - 1) + \mathbb{1}_{d^i=B}\mathbb{F}(s^A)\big),\\
&\text{(B.7)} \le \min\{N^A - s^A - 1, N^A\}\frac{\mu^A}{\Lambda}\mathbb{F}(s^A + 1) + \min\{N^B - s^B - 1, N^B\}\frac{\mu^B}{\Lambda}\mathbb{F}(s^A),\\
&\text{(B.8)} \le (1 - \frac{\lambda^i}{\Lambda} - \min\{N^A - s^A, N^A\}\frac{\mu^A}{\Lambda} - \min\{N^B - s^B, N^B\}\frac{\mu^B}{\Lambda})\mathbb{F}(s^A).
\end{aligned}
$$

Lastly, we can show (B.9) $\le \mathbb{F}(s^A)\dfrac{\mu^A I_{\{}s^A \ge 0\} + \mu^B I_{\{}s^B \ge 0\}}{\Lambda}$ under one of the following sufficient conditions:

- $\mu^A = \mu^B$,
- $\mu^A \le \mu^B$ (BLS serves faster than ALS) and $V_{t+1}(s^A, s^B)$ is concave in s^A, or
- $\mu^A \le \mu^B$ and $V_{t+1}(s^A + 1, s^B + 1) - 2V_{t+1}(s^A, s^B + 1) + V_{t+1}(s^A - 1, s^B + 1)$ $\le \mathbb{F}(s^A)\dfrac{\mu^A I_{\{}s^A \ge 0\} + \mu^B I_{\{}s^B \ge 0\}}{\mu^B - \mu^A}$.

Note that the concavity of $V_{t+1}(s^A, s^B)$ shall be proven by additional mathematical induction which we do not consider.

Now, combining all inequalities listed above for (B.6) – (B.10) yields

$$
\begin{aligned}
&V_t(s^A + 1, s^B) - V_t(s^A, s^B + 1) - V_t(s^A, s^B) + V_t(s^A - 1, s^B + 1)\\
&\le \frac{\lambda^i}{\Lambda}\big((\mathbb{1}_{d^i=A}\mathbb{F}(s^A - 1) + \mathbb{1}_{d^i=B}\mathbb{F}(s^A)\big)\\
&+ \min\{N^A - s^A - 1, N^A\}\frac{\mu^A}{\Lambda}\mathbb{F}(s^A + 1) + \min\{N^B - s^B - 1, N^B\}\frac{\mu^B}{\Lambda}\mathbb{F}(s^A)\\
&+ (1 - \frac{\lambda^i}{\Lambda} - \min\{N^A - s^A, N^A\}\frac{\mu^A}{\Lambda} - \min\{N^B - s^B, N^B\}\frac{\mu^B}{\Lambda})\mathbb{F}(s^A)\\
&+ \frac{\lambda^i}{\Lambda}\mathbb{1}_{d^i=A}\big(\mathbb{F}(s^A) - \mathbb{F}(s^A - 1)\big)\\
&\le \mathbb{F}(s^A).
\end{aligned}
$$

Finally, note that the above proof holds when there are at least two available ALS servers. We will continue to prove for $s^A = 1$, and then show the same result for non-homogeneous action set $\mathbb{D}$. By induction, there is at least one optimal action $\mathbb{D}_t^i$ that is a control limit type policy. Hence, the following list of all possible control limit type policies must include at least one optimal action:

$$\{(A,A,A,A),(B,B,B,B),(A,B,A,B),(B,A,B,A),$$
$$(A,A,B,A),(A,A,B,B),(A,B,B,B),(B,A,B,B)\}$$

So far we have shown that Lemma B.2 holds for (A,A,A,A) and (B,B,B,B). We continue to show the same result for all other action sets in the above list.

First, when $\mathbb{D} = (A,B,A,B)$, we have

$$\begin{aligned}
&\mathbb{V}_{t+1}(s^A,s^B,\mathbb{D}) = \frac{\lambda^i}{\Lambda}\big(V_{t+1}(s^A,s^B) - V_{t+1}(s^A,s^B) - V_{t+1}(s^A-1,s^B) + V_{t+1}(s^A-1,s^B)\big)\\
&+\min\{N^A-s^A-1,N^A\}\frac{\mu^A}{\Lambda}\mathbb{V}_{t+1}(s^A+1,s^B) + \min\{N^B-s^B-1,N^B\}\frac{\mu^B}{\Lambda}\mathbb{V}_{t+1}(s^A,s^B+1)\\
&+(1-\frac{\lambda^i}{\Lambda}-\min\{N^A-s^A,N^A\}\frac{\mu^A}{\Lambda}-\min\{N^B-s^B,N^B\}\frac{\mu^B}{\Lambda})\mathbb{V}_{t+1}(s^A,s^B)\\
&+\frac{\mu^B-\mu^A}{\Lambda}\big(V_{t+1}(s^A+1,s^B+1) - 2V_{t+1}(s^A,s^B+1) + V_{t+1}(s^A-1,s^B+1)\big)\\
&+\frac{\lambda^i}{\Lambda}\big(\omega^i(f^a(s^A+1)-f^a(s^A))U_{aA} + (1-\omega^i)(f^b(s^A+1)-f^b(s^A))U_{bA}\big)\\
&\le (1-\frac{\lambda^i}{\Lambda}-\min\{N^A-s^A,N^A\}\frac{\mu^A}{\Lambda}-\min\{N^B-s^B,N^B\}\frac{\mu^B}{\Lambda})\mathbb{F}(s^A) + \frac{\lambda^i}{\Lambda}\mathbb{F}(s^A) \le \mathbb{F}(s^A).
\end{aligned}$$

The second case is when $\mathbb{D} \in \{(B,A,B,A),(A,A,B,B)$. We show for (B,A,B,A), as the case for $\mathbb{D} = (A,A,B,B)$ follows a nearly identical proof. Since we have been following mathematical induction, by assuming that (B.4) holds for $t+1$ we also get $V_{t+1}(s^A,s^B+$

$1, A) - V_{t+1}(s^A, s^B + 1, B) \geq V_{t+1}(s^A - 1, s^B + 1, A) - V_{t+1}(s^A - 1, s^B + 1, B)$. Therefore,

$$\begin{aligned}
&\mathbb{V}_t(s^A, s^B, \mathbb{D}) \\
&= V_t(s^A + 1, s^B, B) - V_t(s^A, s^B + 1, A) - V_t(s^A, s^B, B) + V_t(s^A - 1, s^B + 1, A) \\
&\leq V_t(s^A + 1, s^B, B) - V_t(s^A, s^B + 1, B) - V_t(s^A, s^B, B) + V_t(s^A - 1, s^B + 1, B)) \\
&= \mathbb{V}_t(s^A, s^B, (B, B, B, B)) \leq \mathbb{F}(s^A).
\end{aligned}$$

The last case is when $\mathbb{D} \in \{(A, A, B, A), (B, A, B, B), (A, A, A, B),$ or $(A, B, B, B)\}$. We show for (A, A, B, A),and then the proof for the rest is analogous.

$$\begin{aligned}
\mathbb{V}_t(s^A, s^B, \mathbb{D}) &= V_t(s^A + 1, s^B, A) - V_t(s^A, s^B + 1, A) - V_t(s^A, s^B, B) + V_t(s^A - 1, s^B + 1, A) \\
&\leq V_t(s^A + 1, s^B, A) - V_t(s^A, s^B + 1, A) - V_t(s^A, s^B, A) + V_t(s^A - 1, s^B + 1, A) \\
&= \mathbb{V}_t(s^A, s^B, (A, A, A, A)) \leq \mathbb{F}(s^A).
\end{aligned}$$

The first inequality holds due to the optimality of $\mathbb{D}^{i*}_{t+1}(s^A, s^B) = (A, A, B, A)$. Note that the problem for saturated states does not arise in any of these action sets. □

Lemma B.3. *If* (B.4) *holds for all pairs of* (s^A, s^B) *such that* $1 \leq s^A \leq N^A - 1$ *and* $0 \leq s^B \leq N^B - 1$, *then* (B.5) *also holds for all 'boundary' cases:* $s^A = 1, 0 \leq s^B \leq N^B - 1$ *or* $1 \leq s^A \leq N^A - 1, s^B = 0$.

Proof. Proof. First let's show that the following inequality is true:

$$\mathbb{1}_{d^i=A}\big(V_{t+1}(1, s^B) - V_{t+1}(0, s^B + 1)\big) \leq \mathbb{1}_{d^i=A}\big(F(1) - F(s^B + 1)\big), \qquad \forall s^B \geq 0. \tag{B.11}$$

We have shown that if $d^i = d^{i^*}_t(1, s^B + 1) = A$ than $V_{t+1}(1, s^B) - V_{t+1}(0, s^B + 1) \leq F(1) - F(s^B + 1)$, which is sufficient to show (B.11). Otherwise, if $d^{i^*}_t(1, s^B + 1) = B$, then both sides of (B.11) is 0.

Case 1. Let's rewrite (B.4) for $s^A = 1, s^B \geq 1$.

$$
\begin{aligned}
&V_t(2, s^B) - V_t(1, s^B + 1) - V_t(1, s^B) + V_t(0, s^B + 1) \\
&= \frac{\lambda^i}{\Lambda}\big(\mathbb{1}_{d^i=A}(V_{t+1}(1, s^B) - V_{t+1}(0, s^B + 1)) + \mathbb{1}_{d^i=B}\mathbb{V}_{t+1}(1, s^B - 1)\big) \\
&+ (N^A - 2)\frac{\mu^A}{\Lambda}\mathbb{V}_{t+1}(2, s^B) + \min\{N^B - s^B - 1, N^B\}\frac{\mu^B}{\Lambda}\mathbb{V}_{t+1}(1, s^B + 1) \\
&+ (1 - \frac{\lambda^i}{\Lambda} - (N^A - 1)\frac{\mu^A}{\Lambda} - \min\{N^B - s^B, N^B\}\frac{\mu^B}{\Lambda})\mathbb{V}_{t+1}(1, s^B) \\
&+ \frac{\mu^B - \mu^A}{\Lambda}\big(V_{t+1}(2, s^B + 1) - 2V_{t+1}(1, s^B + 1) + V_{t+1}(0, s^B + 1)\big) \\
&+ \frac{\lambda^i}{\Lambda}\mathbb{1}_{d^i=A}\big(\omega^i(f^a(2) - f^a(1) - f^a(1) + f^a(s^B + 1)U_{aA} \\
&+ (1 - \omega^i)(f^b(2) - f^b(1) - f^b(1) + f^b(s^B + 1))U_{bA}\big) \\
&\leq \frac{\lambda^i}{\Lambda}\big(\mathbb{1}_{d^i=A}(F(1) - F(s^B + 1)) + \mathbb{1}_{d^i=B}\mathbb{F}(1)\big) \\
&+ (N^A - 2)\frac{\mu^A}{\Lambda}\mathbb{F}(2) + \min\{N^B - s^B - 1, N^B\}\frac{\mu^B}{\Lambda}\mathbb{F}(1) \\
&+ (1 - \frac{\lambda^i}{\Lambda} - (N^A - 1)\frac{\mu^A}{\Lambda} - \min\{N^B - s^B, N^B\}\frac{\mu^B}{\Lambda})\mathbb{F}(1) \\
&+ \frac{\lambda^i}{\Lambda}\mathbb{1}_{d^i=A}\big(\mathbb{F}(1) - F(1) + F(s^B + 1)\big) \\
&\leq \mathbb{F}(1) = \mathbb{F}(s^A).
\end{aligned}
$$

Case 2. Next, the derivation is slightly different when $s^A = 1, s^B = 0$, since when a low priority call arrives at $s_t = (s^A, s^B) = (1, 0)$, an ALS servers the call.

$$
\begin{aligned}
&V_t(2, 0) - V_t(1, 1) - V_t(1, 0) + V_t(0, 1) \\
&= \frac{\lambda^i}{\Lambda}\big(\mathbb{1}_{d^i=A}(V_{t+1}(1, 0) - V_{t+1}(0, 1)) + \mathbb{1}_{d^i=B} \times 0\big) \\
&+ (N^A - 2)\frac{\mu^A}{\Lambda}\mathbb{V}_{t+1}(2, 0) + (N^B - 1)\frac{\mu^B}{\Lambda}\mathbb{V}_{t+1}(1, 1) \\
&+ (1 - \frac{\lambda^i}{\Lambda} - (N^A - 1)\frac{\mu^A}{\Lambda} - N^B\frac{\mu^B}{\Lambda})\mathbb{V}_{t+1}(1, 0) \\
&+ \frac{\mu^B - \mu^A}{\Lambda}\big(V_{t+1}(2, 1) - 2V_{t+1}(1, 1) + V_{t+1}(0, 1)\big) \\
&+ \frac{\lambda^i}{\Lambda}\mathbb{1}_{d^i=A}\big(\omega^i(f^a(2) - 2f^a(1) + f^a(1)U_{aA} + (1 - \omega^i)(f^b(2) - 2f^b(1) + f^b(1))U_{bA}\big)
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{\lambda^i}{\Lambda}\mathbb{1}_{d^i=B}\big(\omega^i(f^a(2)-2f^a(1)+f^a(1))U_{aA}+(1-\omega^i)(f^b(2)-2f^b(1)+f^b(1))U_{bA}\big)\\
&\leq \frac{\lambda^i}{\Lambda}\big(\mathbb{1}_{d^i=A}(F(1)-F(1))+\mathbb{1}_{d^i=B}\times 0\big)+(N^A-2)\frac{\mu^A}{\Lambda}\mathbb{F}(2)+(N^B-1)\frac{\mu^B}{\Lambda}\mathbb{F}(1)\\
&+(1-\frac{\lambda^i}{\Lambda}-(N^A-1)\frac{\mu^A}{\Lambda}-N^B\frac{\mu^B}{\Lambda})\mathbb{F}(1)+\frac{\lambda^i}{\Lambda}\mathbb{1}_{d^i=A}\big(\mathbb{F}(1)-F(1)+F(1)\big)+\frac{\lambda^i}{\Lambda}\mathbb{1}_{d^i=B}\big(\mathbb{F}(1)\big)\\
&\leq \mathbb{F}(1)=\mathbb{F}(s^A).
\end{aligned}
$$

Case 3. Finally, when $s^A \geq 2$ but $s^B = 0$, the derivation is analogous to the above case 1 and 2 for low priority call arrivals and is analogous to the proof for lemma B.2 for the rest of events. □

Now we are ready to prove Theorem 3.3 using Lemma B.2 and B.3. We use mathematical induction on time epoch t to show that (B.4) holds for all t, s^A, s^B.

noitemsep,nolistsep For $t = T-1$, the inequality is true since LHS of (B.4) is 0 while RHS is nonnegative.

noiitemsep,noliistsep Now assume that (B.4) holds for $t = n \leq T$, for all s^A, s^B.

noiiitemsep,noliiistsep Then by Lemma B.2 and B.3, (B.4) is true for $t = n-1$, for all s^A, s^B.

□

Lastly, the proof of the theorem in terms of the number of BLS servers is symmetrical.

B.4 Proof of Theorem 3.4 and Theorem 3.5 (See page 56)

Sketch of the proof of Theorem 3.4. The proof for Theorem 3.1 and 3.2 for the Queue system is the identical to the proof for the Loss System, except for the fact that we expand the state domain to negative values. For Theorem 3.3, here we provide the basic outline of the

proof since it is also analogous to the proof for Loss System. We still use the mathematical induction, and the only difference arises on how we deal with boundary cases. When $s^A \geq 2, s^B \geq 1$, follow the proof for Lemma B.2. When $s^A = 1, s^B \geq 0$ or $s^A \geq 2, s^B = 0$, follow the proof for Lemma B.3. Other state pairs, such as $(s^A, s^B) = (-1, 1)$, are never reached due to the boundary condition.

Sketch of the proof of Theorem 3.5. All proofs in Appendices B.1 – B.3 for Theorem 3.1 – 3.3 are based on mathematical induction on time epoch t of the finite horizon of the base model. This can be reinterpreted as the corresponding iteration in value iteration used to solve the infinite horizon model.

B.5 Details of Generating Call Classes from the Call Log

The dataset used for numerical experiments in Section 3.5 is extracted from the emergency call logs from Hanover County, Virginia from 2009 – 2011, including 33,810 calls for service. Each record contains information regarding call location, primary complains, response times, and the type of service provided. In Hanover County, emergency medical 911 calls are assigned one of three priorities: Priority 1, 2, or 3. Priority 1 patients are life-threatening emergencies, Priority 2 patients may be life-threatening, and Priority 3 patients are not experiencing life-threatening symptoms. In addition to the classified priority, the call log also provides the primary complaint of the patient at dispatch which belongs to one of the followings: (alphabetic order) Abdominal/Back Pain, Diabetic problem, Difficulty Breathing, Fall Victim, Heart Problems, Motor Vehicle Accidents, Sick Person, Seizure/Stroke, Other Problems, Traumatic Injury, Transfer, Unconscious, Unknown Problem.

Based on the assigned priorities and primary complaints, we divide calls into 15 classes as described in Table B.1. Specifically, all priority 1 calls are classified as class 1. Their signal value is set to 1.0 regardless of the primary complaints since priority 1 patients are

Table B.1: Generating call classes using assigned priorities and primary complaints

Priority / Complaint	Priority 1	Priority 2	Priority 3
Diabetic Problem		Class 2	
Heart Problems		Class 3	
Unconscious		Class 4	
Unknown Problem		Class 5	
Seizure/Stroke		Class 6	
Sick Person		Class 7	
Difficulty Breathing	Class 1	Class 8	Class 15
Other Problems		Class 9	
Transfer		Class 10	
Motor Vehicle Accidents		Class 11	
Trauma		Class 12	
Abdominal/back pain		Class 13	
Fall Victim		Class 14	

anticipated to require ALS treatment. Likewise, all priority 3 calls, independent of their primary complaints, are classified as class 15. Their signal value is set to 0 since those patients are experiencing conditions that can be treated by both types of ambulances. In the meantime, priority 2 patients are classified into 13 different call classes (classes 2 – 14) based on the primary complaint. We estimate the signal parameter for each call class from the call log based on the type of vehicles that responded to and transported those calls. The estimated signal value is depicted in Figure 3.1 in decreasing order of ω^i. Note that since we consider all priority 1 patients and priority 3 patients as separate classes independent of their complaints, each value in the figure represents the conditional probability that a patient with given complaint is an advanced patient given that the patient's assigned priority is 2, rather than the fraction of advanced patients among all patients with given complaint. Therefore, primary complaints that seem urgent may appear to have low w^i values in Figure 3.1.

C APPENDIX FOR CHAPTER 4

C.1 Class Separability of the Optimal MDP Policy

For a given time epoch t, state $s_t = (s_{AR}, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT})$ and patient group i, We determine the optimal action by comparing all possible actions and choose the action d_t that results in the highest value function $V_t(s, d)$. For now, we consider the model with three patient groups ($i \in \{1, 2, 3\}$ as in the original MDP model), but the idea here is easily generalized into more patient groups (such as the approximate spatial model).

Let $V_t(s_t, d_t)$ be the conditional value function given that we choose action d_t; therefore, $V_t(s_t) = \max_{d_t} V_t(s_t, d_t)$. Assume that we always compare a pair of actions: in this case, d'_t is strictly better than d''_t if and only if $V_t(s_t, d'_t) - V_t(s_t, d''_t) > 0$. By arranging terms we get

$$\begin{aligned} & V_t(s_t, d'_t) - V_t(s_t, d''_t) \\ & = R_t(s_t, d'_t) - R_t(s_t, d''_t) \\ & + \sum_{s_{t+1} \in S_{t+1}} \big(P_t(s_{t+1}|s_t, d'_t)V_{t+1}(s_{t+1}) - P_t(s_{t+1}|s_t, d''_t)V_{t+1}(s_{t+1})\big). \end{aligned} \tag{C.1}$$

Without the loss of generality, we pick $i = 2$, and compare $d'_t = (d_{1A}, d_{1B}, 1, 0, d_{3A}, d_{3B})$ to $d''_t = (d_{1A}, d_{1B}, 1, 1, d_{3A}, d_{3B})$, i.e., we compare sending an ALS ambulance to sending both an ALS ambulance and a BLS ambulance to a priority 2 call, while holding the action for other patient priority groups the same. For any other choice of group i and action pair d'_t and d''_t, the logic is analogous. Then (C.1) is arranged as follows:

$$\begin{aligned} & V_t(s_t, d'_t) - V_t(s_t, d''_t) \\ & = \frac{\lambda_{2t}}{\Lambda}\big(\omega(U_{HA} - U_{HAB}) + \omega(U_{LA} - U_{LAB})\big) \\ & = \frac{\lambda_{2t}}{\Lambda}\big(V_{t+1}(s_{AR} + 1, s_{BR}, s_{ABR1}, s_{ABR2}, s_{AT}, s_{BT}) \\ & - V_{t+1}(s_{AR}, s_{BR}, s_{ABR1} + 1, s_{ABR2}, s_{AT}, s_{BT})\big). \end{aligned} \tag{C.2}$$

As it can be seen from the fact that (C.2) does not include the action for other patient groups $(d_{1A}, d_{1B}, d_{3A}, d_{3B})$, whether $V_t(s_t, d'_t) - V_t(s_t, d''_t) > 0$ holds or not does not depend on the choice of the action for the other patient group. When we solve the optimality equation (4.1) with backward induction, all value functions $V_{t+1}(s_{t+1})$ that appears in (C.2) are constants. Therefore, we can determine the optimal action for a patient group without considering the all possible combinations of actions.

REFERENCES

Alagoz, Oguzhan, and Mehmet U.S. Ayvaci. 2010. Uniformization in Markov decision processes. In *Wiley encyclopedia of operations research and management science*, ed. J. J. Cochran, L. A. Cox, P. Keskinocak, J. P. Kharoufeh, and J. C. Smith. John Wiley & Sons, Inc.

Alanis, Ramon, Armann Ingolfsson, and Bora Kolfal. 2013. A Markov chain model for an EMS system with repositioning. *Production and Operations Management* 22(1):216–231.

Ansari, Sardar, Laura Albert McLay, and Maria E. Mayorga. 2015. A maximum expected covering problem for district design. *Transportation Science* 51(1):376 – 390.

Ansari, Sardar, Soovin Yoon, and Laura A. Albert. 2017. An approximate hypercube model for public service systems with co-located servers and multiple response. *Transportation Research Part E: Logistics and Transportation Review* 103:143 – 157.

Argon, Nilay Tanik, Serhan Ziya, and James E. Winslow. 2011. Triage in the aftermath of mass-casualty incidents. In *Wiley encyclopedia of operations research and management science*, ed. J. J. Cochran, L. A. Cox, P. Keskinocak, J. P. Kharoufeh, and J. C. Smith. John Wiley & Sons, Inc.

Argon, Nilay Tanık, and Serhan Ziya. 2009. Priority assignment under imperfect information on customer type identities. *Manufacturing & Service Operations Management* 11(4): 674–693.

Aringhieri, R., M.E. Bruni, S. Khodaparasti, and J.T. van Essen. 2017. Emergency medical services and beyond: addressing new challenges through a wide literature review. *Computers and Operations Research* 78:349–368.

Aringhieri, R., G. Carello, and D. Morale. 2016. Supporting decision making to improve the performance of an italian emergency medical service. *Annals of Operations Research* 236(1): 131–148.

Bakalos, G., M. Mamali, C. Komninos, E. Koukou, A. Tsantilas, S. Tzima, and Rosenberg T. 2011. Advanced life support versus basic life support in the pre-hospital setting: A meta-analysis. *Resuscitation* 82:1130 – 1137.

van Barneveld, T.C., R.D. van der Mei, and S. Bhulai. 2017. Compliance tables for an ems system with two types of medical response units. *Computers & Operations Research* 80:68 – 81.

Batta, Rajan, June M. Dolan, and Nirup N. Krishnamurthy. 1989. The maximal expected covering location problem: Revisited. *Transportation Science* 23(4):277–287.

Benders, J. F. 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* 4(1):238–252.

Benn, B. A. 1966. Hierarchical car pool systems in railroad transportation. Ph.D. thesis, Case Institute of Technology, Cleveland, Ohaio.

Beraldi, P., and M.E. Bruni. 2009. A probabilistic model applied to emergency service vehicle location. *European Journal of Operational Research* 196(1):323 – 331.

Birge, John R. 1982. The value of the stochastic solution in stochastic linear programs with fixed recourse. *Mathematical Programming* 24(1):314–325.

Bodur, Merve, Sanjeeb Dash, Oktay Günlük, and James Luedtke. 2017. Strengthened benders cuts for stochastic integer programs with continuous recourse. *INFORMS Journal on Computing* 29(1):77–91.

Bodur, Merve, and James R. Luedtke. 2016. Mixed-integer rounding enhanced benders decomposition for multiclass service-system staffing and scheduling with arrival rate uncertainty. *Management Science* 63(7):2073–2091.

Boujemaa, Rania, Aida Jebali, Sondes Hammami, Angel Ruiz, and Hanen Bouchriha. 2018. A stochastic approach for designing two-tiered emergency medical service systems. *Flexible Services and Manufacturing Journal* 30(1):123–152.

Boyaci, B., and N. Geroliminis. 2015. Approximation methods for large-scale spatial queueing systems. *Transportational Research Part B; Methodological* 74:151–181.

Brotcorne, Luce, Gilbert Laporte, and Frédéric Semet. 2003. Ambulance location and relocation models. *European Journal of Operational Research* 147(3):451 – 463.

Budge, Susan, Armann Ingolfsson, and Erhan Erkut. 2009. Technical note–approximating vehicle dispatch probabilities for emergency service systems with location-specific service times and multiple units per location. *Operations Research* 57(1):251–255.

Burwell, Timothy H., James P. Jarvis, and Mark A. McKnew. 1993. Modeling co-located servers and dispatchties in the hypercube model. *Computers & Operations Research* 20(2): 113–119.

Channouf, Nabil, Pierre L'Ecuyer, Armann Ingolfsson, and Athanassios N. Avramidis. 2007. The application of forecasting techniques to modeling emergency medical system calls in Calgary, Alberta. *Health Care Management Science* 10(1):25–45.

Cho, Soo-Haeng, Hoon Jang, Taesik Lee, and John Turner. 2014. Simultaneous location of trauma centers and helicopters for emergency medical service planning. *Operations Research* 62(4):751–771.

Chong, Kenneth C., Shane G. Henderson, and Mark E. Lewis. 2016. The vehicle mix decision in emergency medical service systems. *Manufacturing & Service Operations Management* 18(3):347–360.

———. 2017. Two-class routing with admission control and strict priorities. *Probability in the Engineering and Informational Sciences* 32(2):1–16.

Church, Richard, and Charles ReVelle. 1974. The maximal covering location problem. *Papers in Regional Science* 32(1):101–118.

Daskin, M. S. 1987. Location, dispatching, and routing model for emergency services with stochastic travel times. In *Spatial analysis and location allocation models*, 224 – 265. Van Nostrang Reinhold Company.

Daskin, Mark S. 1983. A maximum expected covering location model: Formulation, properties and heuristic solution. *Transportation Science* 17(1):48–70.

Enayati, Shakiba, Osman Ozaltin, Maria Mayorga, and Cem Saydam. 2018. Ambulance redeployment and dispatching under uncertainty with personnel workload limitations. *IIE Transactions* 50(9):777–788.

Erkut, Erhan, Armann Ingolfsson, and Güneş Erdoğan. 2008. Ambulance location for maximum survival. *Naval Research Logistics (NRL)* 55(1):42–58.

Erkut, Erhan, Armann Ingolfsson, Thaddeus Sim, and Güneş Erdoğan. 2009. Computational comparison of five maximal covering models for locating ambulances. *Geographical Analysis* 41(1):43–65.

Figueira, Gonçalo, and Bernardo Almada-Lobo. 2014. Hybrid simulation–optimization methods: A taxonomy and discussion. *Simulation Modelling Practice and Theory* 46: 118–134.

Gendreau, M., G. Laporte, and F. Semet. 2001. A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel Computing* 27(12):1641 – 1653.

Geroliminis, Nikolas, Matthew G. Karlaftis, and Alexander Skabardonis. 2009. A spatial queuing model for the emergency vehicle districting and location problem. *Transportation Research Part B: Methodological* 43(7):798–811.

Goldberg, Jeffrey, and Luis Paz. 1991. Locating emergency vehicle bases when service time depends on call location. *Transportation Science* 25(4):264–280.

Goldberg, Jeffrey B. 2004. Operations research models for the deployment of emergency services vehicles. *EMS management Journal* 1(1):20–39.

Grannan, Benjamin C., Nathaniel D. Bastian, and Laura A. McLay. 2015. A maximum expected covering problem for locating and dispatching two classes of military medical evacuation air assets. *Optimization Letters* 9(8):1511–1531.

Iannoni, Ana Paula, and Reinaldo Morabito. 2007. A multiple dispatch and partial backup hypercube queuing model to analyze emergency medical systems on highways. *Transportation Research Part E: Logistics and Transportation Review* 43(6):755–771.

Ingolfsson, Armann, Susan Budge, and Erhan Erkut. 2008. Optimal ambulance location with random delays and travel times. *Health Care Management Science* 11:262–274.

Jacobson, Evin Uzun, Nilay Tanık Argon, and Serhan Ziya. 2012. Priority assignment in emergency response. *Operations Research* 60(4):813–832.

Jagtenberg, C. J., S. Bhulai, and R. D. van der Mei. 2017. Dynamic ambulance dispatching: is the closest-idle policy always optimal? *Health Care Management Science* 20(4):517–531.

Jaiswal, N. K. 1968. *Priority queues*. New York: Academic Press.

Jarvis, J. P. 1985. Approximating the equilibrium behavior of multi-server loss systems. *Management Science* 31(2):235–239.

Kim, S. H., and W. Whitt. 2014. Are call center and hospital arrivals well modeled by non-homogeneous Poisson processes? *Manufacturing & Service Operations Management* 16(3):464–480.

Kimmel, Katarzyna, and David Persse. 2015. Background and advantages of a tiered ems response in a large, fire-based EMS model. *Health Care Current Reviews* 3(1):138.

Kleywegt, Anton J., Alexander Shapiro, and Tito Homem de Mello. 2002. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization* 12(2):479–502.

Kolesar, Peter, and Warren E. Walker. 1974. An algorithm for the dynamic relocation of fire companies. *Operations Research* 22(2):249–274.

Larsen, Mary P., Mickey S. Eisenberg, Richard O. Cummins, and Alfred P. Hallstrom. 1993. Predicting survival from out-of-hospital cardiac arrest: A graphic model. *Annals of Emergency Medicine* 22(11):1652 – 1658.

Larson, Richard C. 1974. A hypercube queuing model for facility location and redistricting in urban emergency services. *Operations Research* 1(1):67–95.

———. 1975. Approximating the performance of urban emergency service systems. *Operations Research* 23(5):845–868.

Li, Xueping, Zhaoxia Zhao, Xiaoyan Zhu, and Tami Wyatt. 2011. Covering models and optimization techniques for emergency response facility location and planning: a review. *Mathematical Methods of Operations Research* 74(3):281–310.

Lord, Dominique, and Fred Mannering. 2010. The statistical analysis of crash-frequency data: A review and assessment of methodological alternatives. *Transportation Research Part A: Policy and Practice* 44(5):291 – 305.

Maio, Valerie J. De, Ian G. Stiell, George A. Wells, and Daniel W. Spaite. 2003. Optimal defibrillation response intervals for maximum out-of-hospital cardiac arrest survival rates. *Annals of Emergency Medicine* 42(2):242 – 250.

Mandell, Marvin B. 1998. Covering models for two-tiered emergency medical service systems. *Location Science* 6:355 – 368.

Marianov, V., and C. ReVelle. 1992. A probabilistic fire-protection siting model with joint vehicle reliability requirements. *Papers in Regional Science* 71:212 – 241.

Maxwell, Matthew S., Eric Cao Ni, Chaoxu Tong, Shane G. Henderson, Huseyin Topaloglu, and Susan R. Hunter. 2014. A bound on the performance of an optimal ambulance redeployment policy. *Operations Research* 62(5):1014–1027.

Maxwell, Matthew S., Mateo Restrepo, Shane G. Henderson, and Huseyin Topaloglu. 2010. Approximate dynamic programming for ambulance redeployment. *INFORMS Journal on Computing* 22(2):266–281.

McLay, Laura A. 2009. A maximum expected covering location model with two types of servers. *IIE Transactions* 41(8):730–741.

———. 2010. Emergency medical service systems that improve patient survivability. In *Wiley encyclopedia of operations research and management science*, ed. J. J. Cochran, L. A. Cox, P. Keskinocak, J. P. Kharoufeh, and J. C. Smith. John Wiley & Sons, Inc.

McLay, Laura A., and Maria E. Mayorga. 2013a. A dispatching model for server-to-customer systems that balances efficiency and equity. *Manufacturing & Service Operations Management* 15(2):205–220.

———. 2013b. A model for optimally dispatching ambulances to emergency calls with classification errors in patient priorities. *IIE Transactions* 45(1):1–24.

McLay, Laura A., and Henri Moore. 2012. Hanover county improves its response to emergency medical 911 patients. *Interfaces* 42(4):380–394.

Mendonça, FC, and R Morabito. 2001. Analysing emergency medical service ambulance deployment on a brazilian highway using the hypercube model. *Journal of the Operational Research Society* 261–270.

Miller, B. 1968. Finite state continuous time Markov decision processes with a finite planning horizon. *SIAM Journal on Control* 6(2):266–280.

Moore, George C., and Charles ReVelle. 1982. The hierarchical service location problem. *Management Science* 28(7):775–780.

Naoum-Sawaya, Joe, and Samir Elhedhli. 2013. A stochastic optimization model for real-time ambulance redeployment. *Computers & Operations Research* 40(5):1972 – 1978.

Nickel, Stefan, Melanie Reuter-Oppermann, and Francisco Saldanha da Gama. 2016. Ambulance location under stochastic demand: A sampling approach. *Operations Research for Health Care* 8(Supplement C):24 – 32.

Noyan, Nilay. 2010. Alternate risk measures for emergency medical service system design. *Annals of Operations Research* 181(1):559–589.

O'Keeffe, Colin, Jon Nicholl, Janette Turner, and Steve Goodacre. 2011. Role of ambulance response times in the survival of patients with out-of-hospital cardiac arrest. *Emergency Medicine Journal* 28(8):703–706.

Peng, Chun, Erick Delage, and Jinlin Li. 2018. Dynamic emergency medical services network design: A novel probabilistic envelope constrained stochastic model and decomposition scheme. Working paper.

Puterman, M. L. 1994. *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons, Inc.

Rajagopalan, H.K., C. Saydam, and J. Xiao. 2008. A multiperiod set covering location model for dynamic redeployment of ambulances. *Computers & Operations Research* 35(3): 814 – 826.

Reilly, Michael J. 2006. Accuracy of a priority medical dispatch system in dispatching cardiac emergencies in a suburban community. *Prehospital and Disaster Medicine* 21(2): 77–81.

Restrepo, M., S.G. Henderson, and Huseyin Topaloglu. 2009. Erlang loss models for the static deployment of ambulances. *Health Care Management Science* 12:67–79.

Rettke, Aaron J., Matthew J. Robbins, and Brian J. Lunday. 2016. Approximate dynamic programming for the dispatch of military medical evacuation assets. *European Journal of Operational Research* 254(3):824 – 839.

Reuter-Oppermann, Melanie, Pieter L. van den Berg, and Julie L. Vile. 2017. Logistics for emergency medical service systems. *Health Systems* 6(3):187–208.

ReVelle, C., and V. Marianov. 1991. A probabilistic fleet model with individual vehicle reliability requirements. *European Journal of Operational Research* 53:93 – 105.

Robbins, H., and S. Monro. 1951. A stochastic approximation method. *Annals of Mathematical Statistics* 22:400–407.

Schaack, Christian, and Richard C. Larson. 1986. An n-server cutoff priority queue. *Operations Research* 34(2):257–266.

Schilling, David, D. Jack Elzinga, Jared Cohon, Richard Church, and Charles ReVelle. 1979. The team/fleet models for simultaneous facility and equipment siting. *Transportation Science* 13(2):163 – 175.

Serfozo, Richard F. 1979. Technical note—an equivalence between continuous and discrete time markov decision processes. *Operations Research* 27(3):616–620.

Setzler, Hubert, Cem Saydam, and Sungjune Park. 2009. Ems call volume predictions: A comparative study. *Computers & Operations Research* 36(6):1843 – 1851.

de Souza, R. M., R. Morabito, F. Y. Chiyoshi, and A. P. Iannoni. 2015. Incorporating priorities for waiting customers in the hypercube queuing model with application to an emergency medical service system in Brazil. *European Journal of Operational Research* 242(1):274–285.

Stidham, Shaler, and Richard Weber. 1993. A survey of markov decision models for control of networks of queues. *Queueing Systems* 13(1):291–314.

Sudtachat, Kanchala, Maria E. Mayorga, and Laura A. McLay. 2016. A nested-compliance table policy for emergency medical service systems under relocation. *Omega* 58(Supplement C):154 – 168.

Taylor, I. D. S., and J. G. C. Templeton. 1980. Waiting time in a multi-server cutoff-priority queue, and its application to an urban ambulance service. *Operations Research* 28(5): 1168–1188.

Valenzuela, Terence D., Denise J. Roe, Shan Cretin, Daniel W. Spaite, and Mary P. Larsen. 1997. Estimating effectiveness of cardiac arrest interventions. *Circulation* 96(10):3308–3313.

Waalewijn, Reinier A., Rien de Vos, Jan G.P. Tijssen, and Rudolph W. Koster. 2001. Survival models for out-of-hospital cardiopulmonary resuscitation from the perspectives of the bystander, the first responder, and the paramedic. *Resuscitation* 51(2):113 – 122.

Yoon, Soovin, and Laura A. Albert. 2018a. Dynamic resource assignment for emergency response with multiple types of vehicles. Technical Report, University of Wisconsin-Madison.

———. 2018b. An expected coverage model with a cutoff priority queue. *Health Care Management Science* 21(4):517–533.

Yue, Yisong, Lavanya Marla, and Ramayya Krishnan. 2012. An efficient simulation-based approach to ambulance fleet allocation and dynamic redeployment. In *Proceedings of the twenty-sixth aaai conference on artificial intelligence*.

Zaffar, Muhammad Adeel, Hari K. Rajagopalan, Cem Saydam, Maria Mayorga, and Elizabeth Sharer. 2016. Coverage, survivability or response time: A comparative study of performance statistics used in ambulance location models via simulation–optimization. *Operations Research for Health Care* 11(Supplement C):1 – 12.

Zhou, Zhengyi, David S. Matteson, Dawn B. Woodard, Shane G. Henderson, and Athanasios C. Micheas. 2015. A spatio-temporal point process model for ambulance demand. *Journal of the American Statistical Association* 110(509):6–15.