2013

Optimization and Spatial Queueing Models to Support Multi-Server Dispatching Policies with Multiple Servers per Station

Sardar Ansari
Virginia Commonwealth University

Virginia Commonwealth University
Department of Statistical Sciences and Operations Research

This is to certify that the thesis prepared by Sardar Ansari entitled OPTIMIZATION AND SPATIAL QUEUEING MODELS TO SUPPORT MULTI-SERVER DISPATCHING POLICIES WITH MULTIPLE SERVERS PER STATION has been approved by his committee as a requirement for the Master of Science degree.

Laura A. McLay, PhD, Advisor, Department of Industrial and Systems Engineering, University of Wisconsin-Madison

J. Paul Brooks, PhD, Director, Department of Statistical Sciences and Operations Research

Xi Chen, PhD, Department of Statistical Sciences and Operations Research

Date:

OPTIMIZATION AND SPATIAL QUEUEING MODELS TO SUPPORT MULTI-SERVER DISPATCHING POLICIES WITH MULTIPLE SERVERS PER STATION

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree at Virginia Commonwealth University.

by

SARDAR ANSARI

Advisor: LAURA A. MCLAY
Associate Professor, Department of Industrial and Systems Engineering,
University of Wisconsin-Madison

Virginia Commonwealth University
Richmond, VA
Dec, 2013

ACKNOWLEDGMENT

First and foremost, I would like to thank my advisor, Dr. Laura McLay, for her guidance, support, encouragement and mentorship during the last few years. Special thanks to my committee members, Dr. James P. Brooks and Dr. Xi Chen for their feedback on this work. I would also like to thank my amazing family, Mehdi Ansari, Fakhri Momenzadeh and Targol Ansari, and my wonderful friends who have inspired me in every step of this path. Without their love and support, this work would have not been possible. Finally, I want to thank Mr. Dimitri Topolov who was always beside me when I needed his help.

Contents

List of Figures

List of Tables

ABSTRACT

OPTIMIZATION AND SPATIAL QUEUEING MODELS TO SUPPORT MULTI-SERVER DISPATCHING POLICIES WITH MULTIPLE SERVERS PER STATION

by Sardar Ansari

Master of Science Thesis

Virginia Commonwealth University, 2013

Advisor: Laura A. McLay, Ph.D

Associate Professor, University of Wisconsin-Madison

Director: J. Paul Brooks, Ph.D

Associate Professor, Virginia Commonwealth University

In this thesis, we propose novel optimization and spatial queueing models that expand the currently existing methods by allowing multiple servers to be located at the same station and multiple servers to be dispatched to a single call. In particular, a mixed integer linear programming (MILP) model is introduced that determines how to locate and dispatch ambulances such that the coverage level is maximized. The model allows multiple servers to be located at the same station and balances the workload among them while maintaining contiguous first priority response districts. We also propose an extension to the approximate Hypercube queueing model by allowing multi-server dispatches. Computational results suggest that both models are effective in optimizing and analyzing the emergency systems. We also introduce the $M[G]/M/s/s$ queueing model as an extension to the $M/M/s/s$ model which allows for multiple servers to be assigned to a single customer.

Chapter 1

Introduction

Optimization and spatial queueing models in emergency systems have been studied by operations research community for several decades. The performance of such systems is critical for providing effective care and protecting the safety of the urban areas. There have been numerous mathematical and statistical models proposed throughout the years to address different aspects of emergency systems. Some of these models focus on the stochastic aspects of such systems. Most of them utilize queueing approaches to compute various different quantities that characterize the stochastic aspects of these systems. For example, service providers are often interested in measuring the utilization factors of their servers, i.e., the proportion of times that a server is busy. Knowing the utilization factor for each server allows the service provider to assess the availability of each server which can lead to effective decisions regarding relocating the servers or increasing (decreasing) the number of servers at a station for instance.

The steady-state probabilities are another set of values that characterize long-term behaviour of emergency systems. These probabilities indicate the distribution of busy servers in the system, i.e., the probability that n servers are busy in the system in the long-run. Another set of values that can be computed for an emergency system is dispatch probabilities. They represent the probability that a given station responds to a call from a given call location.

Computing the exact values for these queueing factors are complicated and computationally expensive. Several approximation methods have been proposed in the literature that are manageable in terms of size and complexity. Each of these methods is specifically designed to analyze a certain type of emergency system with certain features and characteristics. We will present some of these queueing models later in the thesis. We also propose a novel queueing model to analyze server dispatching systems with multiple servers per location and multi-server dispatches.

Another type of models that have been proposed for studying the emergency systems are the optimization models. The system administrators often need to answer questions such as how many servers are required to cover the emergency calls in a certain area, where should the servers be located in the area to minimize the service times, and how should the servers be assigned to the calls to achieve a certain coverage level. The goal of these optimization models is to address questions alike and to provide optimal solutions that can improve the efficacy of the emergency systems. The early optimization models that were proposed ignored the stochastic aspect of the system. Later work tried to propose more realistic models by incorporating stochastic elements of the system. Some of these models employed the queueing methods that estimate the stochastic aspects of the system. A review of these spatial queueing and optimization models will be presented later in this chapter.

Although the models that are proposed in this work can be valid for any types of emergency systems, we will focus on emergency medical services (EMS) systems. Nevertheless, these models can be used to model fire and 911 calls as well.

In the remainder of this chapter, we discuss the aims and motivations of this work and provide a literature review. In Chapter 2, we introduce a mixed integer programming model to optimize the dispatching policy for the ambulances in an EMS model. Then, a queueing model for analyzing the emergency systems with multi-server dispatches is introduced in Chapter 3. Finally, the summary of the thesis and the future works are presented in Chapter 4.

1.1 Aims

The goal of this thesis is to expand the currently existing optimization and spatial queueing models to accommodate multiple servers per location and multi-server dispatches. We first propose a linear mixed integer programming (MIP) model that employs the currently existing spatial queueing models for analyzing the stochastic aspects of the emergency systems with multiple servers per location. Our model seeks the optimal location for the servers as well as the optimal dispatching policy that maximizes the coverage, i.e., the proportion of calls that are reached within 9 minutes from the instant the call is received. The proposed model also balances the offered load among the servers and allows for multiple call types. The model uses the correction factors derived by the Hypercube model to approximate the dispatch probabilities. It then uses those probabilities to optimize the location of the servers and the dispatching policy. The model uses an iterative approach such that the MIP remains linear despite the non-linear nature of the Hypercube model. It also balances the workloads of the servers and maintains contiguous first priority districts which leads to more intuitive dispatching policies.

The second aim of this thesis is to expand the existing approximate Hypercube models to accommodate for multi-server dispatches where a call can be responded to by more than one server. The proposed queueing model also allows for multiple servers per station, making it a more realistic model for emergency vehicles such as fire engines and police cars where multiple vehicles often respond to a single call. We also propose an iterative procedure that computes both server utilizations and dispatching policies. This approximate queueing model can be used to create optimization models to find the optimal dispatch policies for systems with multiple servers per location and multi-server dispatch policies.

1.2 Motivation

Emergency systems are one of the most time-sensitive areas in transportation. The response time of the emergency units has a critical role in the effectiveness of such systems. For example, Pell et al. (2001) reports that three quarters of the mortality caused by cardiac arrest occurs in the community and that reducing the ambulance response time is crucial in improving the survivability. Despite the ongoing work in the literature to address problems that are associated with emergency systems, there is still a need to improve the currently existing models and propose new solutions that better represent the real-world emergency systems.

There has been several spatial queueing and optimization models proposed to model the emergency systems. However, most of the focus has been on models with single server per location and single server dispatch policies. There is a need for more realistic models that can better model the emergency systems in which more than one server is often located at each station and multiple servers can respond to the calls. For example, fire departments usually locate multiple fire engines at every fire station and respond to sever calls by dispatching multiple engines. Likewise, the police departments often respond to high-risk 911 calls by sending multiple police units. As a result, it is necessary to design spatial queueing models that can analyze such systems by explicitly considering the multi-dispatch aspect of the calls. Such models should also allow for multiple servers to be located at each station to represent the real-world emergency systems such as EMS and fire stations.

1.3 Literature Review

This thesis proposes a MILP model for simultaneously locating and dispatching ambulances to emergency medical patients, where ambulance busy probabilities and random travel times are taken into account. It also proposes a new set of Hypercube correction factors and an iterative procedure along with it to analyze emergency systems with multi-server dispatch policies.

There is a rich operations research literature on ambulance location models (see Swersey

(1994); Brotcorne et al. (2003); Goldberg (2004)). Early probabilistic models for ambulance location maximize expected coverage by including ambulance busy probabilities. Daskin (1983) assumes that all ambulances share a common busy probability and that ambulances operate independently. Batta et al. (1989) lifts these assumptions by embedding a Hypercube model (Larson (1975)) into the location model to include ambulance dependencies and location specific busy probabilities. Goldberg et al. (1990) extends the model by Daskin (1983) by using stochastic travel and service times to compute ambulance busy probabilities, allows for dispatch policies through a contingency table as we do in this thesis, and considers prioritized calls for service. Only a few papers in this area, however, balance workload imbalances. Pirkul and Schilling (1988, 1991) provide mixed integer programming models that balance the workload across the ambulances by adding workload balance constraints that are similar to those considered in this thesis. Berman et al. (2009) study load balancing when locating facilities on a network that is a tree by extending the p-median problem.

Other papers propose models that consider ambulances that are not always available. ReVelle and Hogan (1989) propose two models that maximize the expected coverage over a given reliability level. The first model assumes system-wide ambulance busy probability while this assumption is relaxed in their second model by incorporating location specific busy probabilities. The busy probabilities are estimated using the calls within each ambulance's region; however, the model does not capture ambulances not being available due to a call from another region. Ball and Lin (1993) propose a reliability model that ensures that the coverage of every demand location is above a prespecified reliability level.

The papers listed thus far assume that travel times are deterministic with the exception of the paper by Goldberg et al. (1990), leading to binary coverage for individual calls (i.e., they are covered or they are not). This binary coverage is contingent on whether an ambulance is available within the call's covering area. Several papers consider random travel times that lead to coverage that is real-valued instead of binary. Ingolfsson et al. (2008) use correction factors and propose a model that minimizes the number of servers required to assure a pre-specified coverage level while allowing for stochastic travel times and delays as well as pre-trip delays. However, they assume that the dispatch policies are given in advance. Church and Roberts (1983); Berman et al. (2003); Karasakal and Karasakal (2004); Alsalloum and Rand (2006) also include partial coverage of calls via uncertain ambulance travel times. Erkut et al. (2009) compare the model proposed in Ingolfsson et al. (2008) with four other models proposed previously that ignore the stochastic aspects of EMS service completely or in part. Their evaluation shows that inclusion of stochastic elements in the model can result in up to 26% increase in the coverage, which suggests that uncertain travel times are important for model realism.

Many of the optimization models listed above use Hypercube models for estimating ambulance busy probabilities. Larson (1974) provides an exact method for modeling the statistical dependence between the ambulances serving an area, and Larson (1975) develops an approximate Hypercube model to compute Hypercube "correction factors." Both the exact and approximate Hypercube models assume Poisson arrival rates, exponential service times, service times that are independent of call locations, and one responder per call. Moreover, they use a pre-specified contingency table to represent the dispatching policy as we do in this thesis. The Hypercube model is used in several cities, including Boston, New York and Orlando, to analyze and study the travel times (Brandeau and Larson (1986); Larson and Rich (1987); Sacks and Grief (1994)). More recently, Larson (2004) uses the Hypercube model as a deployment model to respond to emergency situations such as terrorist attacks. Other studies that use the Hypercube model to build optimization models for server locations and assignment are Chiyoshi et al. (2003); Saydam and Aytuğ (2003); Galvão et al. (2005). Halpern (1977) improves the accuracy of the Hypercube model by allowing server dependent service times. Jarvis (1985) further extends the Hypercube model approximation by allowing for service times that depend on both the ambulance and the call location. Burwell et al. (1993) proposed a modification to the Hypercube model approximation to accommodate ambulances co-located at a single station through "preference ties," i.e., when multiple ambulances are equally preferred to respond to a call. The advantage of this model compared to similar models proposed earlier is that it is not limited by the computer storage. Budge et al. (2009) proposed an approximate Hypercube model that considers station-specific busy probabilities and explicitly allows multiple ambulances per location. Other extensions of the hypercube model that relax the assumptions of the original model or improve its computational complexity can be found in Chelst and Jarvis (1979); Larson and Mcknew (1982); Mendonça and Morabito (2001). We incorporate the model in Budge et al. (2009), which is summarized in Section 2.2, into our proposed optimization model in Chapter 2. The queueing model proposed in Chapter 3 extends the model in Budge et al. (2009) in order to allow for co-located servers and multi-server dispatches.

The distribution of the arrival time of the first vehicle that arrives at the scene when multiple vehicles are dispatched is studied in Daskin and Haghani (1984). Chelst and Barlach (1981) develop a model based on the Hypercube model that dispatches one or two ambulances to a single call. Iannoni and Morabito (2007) extend the Hypercube model further and propose a model for dispatching single, double and triple vehicles to a single call. Their model assumes a specific dispatching policy designed for EMS units that operate on Brazilian highways. On the contrary, the approximate Hypercube model that is proposed in Chapter 3 is neither restricted by the number of dispatches, nor it assumes a particular dispatching policy. Moreover, Chelst and Barlach (1981) assume that the service times are independent for the servers that are busy serving the same call. This is not a realistic assumption and is relaxed in the proposed model. Our queueing model also allows multiple servers to be co-located at the same station. Geroliminis et al. (2009) develop an exact Hypercube model with service times that depend on the responding vehicle, and they embed the Hypercube model in a location model that seeks to minimize the mean response time subject to meeting a coverage level target.

The papers thus far assume that a dispatching policy is known a priori. Integrating ambulance dispatch with ambulance locations is an important aspect of our proposed model. Carter et al. (1972) use a queueing optimization model to determine the locations of two EMS response areas (i.e., beats) to balance the workload between two ambulances. Each ambulance is assumed to always respond to patients within its response area, if the ambulance is available. If both ambulances are busy, patients are served by ambulances outside of the area. They show that it is not always best to dispatch the closest ambulance. Jarvis (1975) examines optimal dispatch with a Markov decision process model that embeds the Hypercube queueing model for dispatching ambulances to patients such that the mean distance traveled when responding to a call is minimized. Weintraub et al. (1999) develop a model for dispatching electric utility resources to prioritized customers.

Swersey (1982) develops a Markov model for determining how many fire engines to send to prioritized fire calls that captures the cost of under-prioritizing calls and sending too few fire engines. Ignall et al. (1982) extend Swersey's model to account for which fire engines to send when calls and fire engines are spatially distributed, and they provide a "preparedness" heuristic. Both Andersson and Värbrand (2007) and Lee (2011) propose similar "preparedness" heuristics for dispatching ambulances to calls. Iannoni et al. (2011) consider how to locate ambulances along a highway while constructing primary and secondary districts. They focus on one-dimensional location along a highway and their districts are only defined by the distance between a vehicle and potential customers.

More recently, patient survivability models are applied to the ambulance location problem using optimization models and Mont Carlo simulation in Erkut et al. (2008b). Knight et al. (2012) propose a model for locating ambulances that respond to calls from multiple classes of heterogeneous patient by maximizing the survival rate. McLay and Mayorga (2012b) develop a Markov decision process for dispatching ambulances to prioritized patients to maximize coverage that include uncertain travel times that depend on call locations and the responding ambulance. Their model is extended by McLay and Mayorga (2012a) to consider “fair” dispatching policies. One such fair policy seeks to balance the workload among the ambulances, since maximizing coverage can introduce workload inequities among service providers. In both of these papers, the optimal policies do not always conform to contingency tables. However, they note that decisions that violate a contingency table paradigm are rare, which suggests that contingency tables are reasonable to use.

Much of the previous work in emergency medical dispatch has focused on either ambulance location, queueing dynamics, or ambulance dispatch. In contrast to previous work in the area, we investigate how to integrate these three issues in a single model that both locates ambulances while determining how to use them while maintaining model realism through uncertain travel times. Also, we introduce an extension to the currently existing approximation Hypercube models by relaxing the assumption of single dispatch per call. The model allows more than one server to be dispatched when a sever call arrives.

Chapter 2

A Maximum Expected Covering Problem for Locating and Dispatching Servers

2.1 Introduction

Emergency medical service (EMS) systems deliver resources to patients, perform pre-hospital care, and deliver patients to hospitals. The timely delivery of resources to patients is necessary to ensure good patient outcomes such as high rates of patient survival. Locating and dispatching medical units are two interrelated problems that are critical for identifying effective responses for responding to and treating emergency medical patients.

When a new call arrives at an EMS dispatch center, a dispatcher determines the severity of the call and sends appropriate medical units. The severity of the call reflects the type of call (such as trauma, diabetes, or chest pains). When a call arrives, an ambulance is immediately dispatched to a patient if one is available. Otherwise, patients enter a queue and wait for an ambulance to become free. As a general rule, ambulance service cannot be preempted, and in most settings (and as considered in this thesis), when an ambulance completes service, it returns to its home station. Nearly all EMS systems in the United States use a *coverage level* performance measure to guide their use of resources, where the coverage level reflects the proportion of high-priority patients that are responded to within a fixed timeframe (usually within nine minutes of dispatch). This decision paradigm leads to two challenges for an EMS system interested in maximizing coverage. While in nearly all cases it is desirable to send a nearby ambulance to a patient, this does not imply that dispatching decisions are simple. If a call for services arises that is located almost exactly between two stations, it may be optimal to send the ambulance that is slightly farther if the call volume surrounding the slightly closer ambulance is high (McLay and Mayorga (2012b)). Therefore, effective dispatching must balance the needs of the current patient with potential future patients that may arrive to the system. The dispatching issue becomes more complicated when it is combined with the one-time design decision of where to locate ambulances, since where ambulances are located influences how they are dispatched and vice versa.

This chapter proposes a mixed integer linear programming (MILP) model that identifies how to locate and dispatch ambulances to emergency medical patients. The model objective maximizes the coverage level, the proportion of high-priority patients who are responded to within a fixed timeframe. This model locates and creates a series of districts for each open station, where an open station is defined as a station where an ambulance is located. This series of districts in turn define a dispatching policy.

This chapter makes the following three contributions to the literature. First, the model captures an important level of realism and practicality. To capture realism, the model here incorporates two sources of uncertainty: (a) uncertainty in ambulance availability captured by the inclusion of ambulance busy probabilities and (b) uncertainty in ambulance travel times that lead to real-valued coverage as opposed to 0-1 coverage. To maintain practicality, we add two sets of side constraints to the MILP model. The first set ensures that the first-priority districts are contiguous, thus leading to more intuitive policies. The second set maintains a balanced workload across the ambulances so that some personnel are not overworked. The latter issue is important, since earlier works on dispatching suggests that improvements made to dispatching may increase workload imbalances (McLay and Mayorga (2012b,a)).

Second, the results provide a series of districts surrounding each open station that correspond to a contingency table. In other words, an ambulance's first district captures the locations to which the ambulance would be the first preferred ambulance to send if it is available, the second district captures the locations to which the ambulance would be sent if a location's first priority ambulance is busy, and so on. Defining dispatch through district design allows for a linear formulation of the assignment of call locations to stations and is a novel way to interpret and implement ambulance dispatching policies. It is also practical, since contingency tables are widely used by EMS systems throughout the world.

Third, we provide an iterative two stage algorithm to solve the proposed MILP model that simultaneously determines how to locate and dispatch ambulances. The input parameters related to ambulance availability reflect the underlying queueing dynamics, which introduce nonlinearities into the MIP model. In the first stage, we estimate the input parameters related to queueing using a Hypercube model approximation Budge et al. (2009), and these parameters are then treated as constants in the MIP model. In the second stage, we solve the MILP model using standard algorithms. Both stages are repeated until the ambulance workloads are approximately equal (subject to tolerances). This solution procedure allows for the solution of large-scale problem instances, and offers an improvement over earlier models that relied on Markov decision process models and algorithms.

A real-world example illustrates the results across three models: (1) the base model that does not maintain contiguity or a balanced workload amongst the ambulances, (2) a model that balances the workload, and (3) a model that both maintains contiguity and a balanced workload. The results suggest that load balancing and contiguity can be achieved with a minimal impact on the coverage level. In one scenario, for example, the range of server busy probabilities was 0.135 in the base model. This range reduced to 0.016 after load balancing and contiguity constraints were enforced while the absolute reduction in coverage level was only 0.1% relative to the base model. The example suggests that the proposed model can effectively respond to temporal variations in demand by changing the location of the ambulances and the dispatching policy.

The rest of the chapter is organized as follows. We review the Hypercube model approximation that is used in the proposed MILP model in Section 2.2, since it is central to our modeling paradigm. The proposed method is introduced in Section 2.3. A description of the data used to evaluate the model follows. Section 2.4 presents the computational results and an assessment of the proposed model. Finally, the concluding remarks are given in the last section.

2.2 Correction Factors for Location-Specific Service Times and Multiple Units per Location

In this section, we review a Hypercube model approximation proposed by Budge et al. (2009). The model facilitates the computation of the correction factors that take location-specific service times into account and allows for more than one server at a station. This latter issue is a characteristic of the MILP model proposed in the next section. Table 2.1 summarizes the symbols used throughout this chapter. The Hypercube model approximation summarized in this section computes the mean service time τ, offered load ρ, utilizations r and r_w, $w \in W$, and correction factors q_{jpm}, $j \in J; p = 1, 2, ..., s; m = 1, 2, ..., c_w$, for a dispatch policy with at most c_w servers per station. Repeating the same process iteratively, the model parameters converge to a suboptimal solution to the dispatching problem.

When a dispatching policy (preference list) is given, Erkut et al. (2009) provides a mechanism to compute the server busy probabilities associated with that policy. This mechanism assumes that the calls from a customer location $j \in J$ arrive according to a to Poisson process with arrival rate λ_j, where $\lambda_j = \lambda_j^L + \lambda_j^H$ is the total demand rate from node j, and there are s servers in the system to respond to the calls. Assume that the number of servers located at station $i \in I$ is denoted by s_i. When a call arrives, one of the available servers located at a station $i \in I$ responds to that call. The calls are responded to based on

Table 2.1: Summary of the Symbols

Symbol	Description	Domain
J	Set of all customer (demand) nodes.	
W	Set of all potential station locations.	
s	Total number of servers in the system.	
s_i	Number of servers at station i ($\sum_{i \in W} s_i = s$).	$i \in I$
c_w	Capacity of station w (the maximum number of servers that can be located at w).	$w \in W$
I	Set of all open stations (i.e., $I = \{i \in W : \ s_i > 0\}$), with $I \subseteq W$.	
λ_j^H (λ_j^L)	Mean high-priority (low-priority) call arrival rate from node j, with $\lambda_j = \lambda_j^H + \lambda_j^L$.	$j \in J$
λ	System-wide total call arrival rate with $\lambda = \sum_{j \in J}(\lambda_j^H + \lambda_j^L) = \sum_{j \in J} \lambda_j$.	
τ_{wj}	Mean service time for calls originated from node j and served by a server from a potential station w (replace w with i for open stations).	$w \in W$, $j \in J$
τ	System-wide mean service time.	
ρ	System-wide mean offered load per server.	
r	System-wide mean server utilization.	
r_w	Utilization factor for a server located at station w.	$w \in W$
a_{ij}	Preference of server i for responding to a call from node j in the dispatching policy.	$i \in I$, $j \in J$
b_{kj}	kth preferred station for calls from node j in the dispatching policy.	$k = 1, ..., s$, $j \in J$
P_s	Loss probability: probability that all s servers are busy.	
P_0	Idle probability: probability that all servers are idle.	
R_{wj}	Fraction of calls from j that are reached by servers from station w in nine minutes.	$w \in W$, $j \in J$
$Q_j(\{s_i\}, \rho, k)$	The correction factors as a function of the distribution of servers, offered load and the server.	$i \in I$, $k = 1, ..., s$
q_{jpm}	A precomputed correction factor (constant) for customer j's p^{th} priority station at which there are m servers located.	$j \in J$, $p = 1, ..., s$, $m = 1, ..., c_{b_{pj}}$
N_{wj}	Set of demand nodes that are neighbors to j and are closer to station w than j.	$w \in W$, $j \in J$
ε	Server utilization deviation tolerance.	
δ	Server utilization imbalance tolerance.	

a dispatching policy, where a_{ij} denotes station i's order in the preference list in responding to a call from location j, $i \in I$, $j \in J$. That is, if $a_{ij} = 2$, then station i is the second most preferred station to respond to calls at j, and therefore, servers from station i respond to calls at j only when servers at the most preferred station are unavailable. Alternatively, the kth preferred station for customer location j is denoted by b_{kj}. For example, if station i is the second preferred station for customer location j, then $a_{ij} = 2$ and $b_{2j} = i$.

The service time for a server from station i responding to a call from customer location j has a mean of τ_{ij}, which includes the response (travel) time, the time spent at the site, the travel time to the hospital if the patient is transferred and the travel time back to the station. Jarvis (1985) suggests that the dispatch probabilities are not sensitive with respect to the distribution of the service times. Therefore, the correction factors assume a general distribution. The system is assumed to have zero-line capacity, i.e., the calls that arrive when all the servers are busy are regarded as 'lost' demand. The loss probability, the probability that all servers are busy, is denoted by P_s. In general, P_n is the probability that n servers are busy responding to calls with P_0 denoting the idle probability, the probability that all servers are idle. It is worth noting that EMS systems are generally considered low-traffic EMS systems, where the number of servers in the system lead to extremely low loss probabilities. This observation also follows from the computational examples in Section 5.

The system-wide mean server utilization (i.e., busy probability) is given by $r = (1 - P_s)\rho$, where ρ represents the system-wide mean offered load per server. The number of servers located at the kth preferred station for node j is denoted by $s_{(k)j} = s_{b_{kj}}$. Also, the utilizations for those servers are shown by $r_{(k)j} = r_{b_{kj}}$. Assume that station i is the kth preferred station for customer location j, i.e., $a_{ij} = k$. Then dispatch probability f_{ij}, the probability that a server from i responds to a call from j, is approximated as

$$f_{ij} \approx Q_j(\{s_{(k)j}\}, \rho, k)(1 - r_i^{s_i}) \prod_{u=1}^{k-1} r_{(u)j}^{s_{(u)j}} \tag{2.1}$$

The factors $Q_j(\{s_{(k)j}\}, \rho, k)$ in (2.1) are the correction factors that approximately correct for the unrealistic assumption of server independence. The server utilizations r_i and correction factors Q_j are computed later in this section. Note that for each customer location j, the dispatch probabilities and the loss probability should add up to 1,

$$\sum_{i \in I} f_{ij} + P_s = 1. \tag{2.2}$$

The earlier papers (Larson (1975); Jarvis (1985)) assume that the correction factors are independent from the customer locations and server locations, whereas the correction factors in this model depend on both of these factors. Setting $z_{(k)j} = s_{(1)j} + s_{(2)j} + ... + s_{(k)j}$, the cumulative number of servers in the top k preferred stations for customer location j, the correction factors Q_j can be expressed as

$$Q_j(\{s_{(k)j}\}, \rho, k) = \frac{P_0 \sum_{n=z_{(k-1)j}}^{s-1} \frac{(\rho s)^n}{n!} [\prod_{u=0}^{z_{(k-1)j}-1} \frac{n-u}{s-u} - \prod_{u=0}^{z_{(k)j}-1} \frac{n-u}{s-u}]}{r^{z_{(k-1)j}} \left(1 - r^{s_{(k)j}}\right)}. \tag{2.3}$$

The server utilizations r_i, the average fraction of time server i is busy, are

$$r_i = \frac{1}{s_i} \sum_{j \in J} \lambda_j f_{ij} \tau_{ij}. \tag{2.4}$$

Equations (2.1), (2.3) and (2.4) form the elements of an iterative procedure for estimating dispatch probabilities f_{ij} and server utilizations r_i. The steps of this iterative procedure are as follows.

Step 0. The system-wide mean service time is initialized as

$$\tau^0 = \frac{1}{\lambda s} \sum_{i \in I} s_i \sum_{j \in J} \lambda_j \tau_{ij}. \tag{2.5}$$

To initialize the server utilizations, one needs to first compute the initial loss probability using Erlang's loss formula,

$$P_s^0 = \frac{(\lambda \tau^0)^s P_0^0}{s!}, \tag{2.6}$$

where the initial idle probability P_0^0 is

$$P_0^0 = \frac{1}{\sum_{i=0}^{s} \frac{(\lambda \tau^0)^i}{i!}}. \tag{2.7}$$

The server utilizations are then initialized as

$$r_i^0 = r^0 = \frac{\lambda \tau^0 (1 - P_s^0)}{s}. \tag{2.8}$$

The iteration counter t is set to one.

Step 1. The idle and loss probabilities are updated using the most recently computed system-wide mean service time, τ^{t-1},

$$P_0^t = \frac{1}{\sum_{i=0}^{s} \frac{(\lambda \tau^{t-1})^i}{i!}}, \tag{2.9}$$

$$P_s^t = \frac{(\lambda \tau^{t-1})^s P_0^t}{s!}. \tag{2.10}$$

Step 2. Let

$$V_i^t = \sum_{j \in J} \lambda_j \tau_{ij} Q_j(\{s_{(k)j}\}, \rho^{t-1}, a_{ij}) \prod_{u=1}^{a_{ij}-1} (r_{(u)j}^{t-1})^{s_{(u)j}}. \tag{2.11}$$

The server utilizations are updated when $r_i^{t-1} \leq 0.5$ using

$$r_i^t = \frac{V_i^t}{s_i + (r_i^{t-1})^{s_i - 1} V_i^t}, \tag{2.12}$$

otherwise using

$$r_i^t = \left(\frac{V_i^t}{V_i^t + \frac{s_i}{(r_i^{t-1})^{s_i - 1}}} \right)^{\frac{1}{s_i}}. \tag{2.13}$$

The correction factors $Q_j(\{s_{(k)j}\}, \rho^{t-1}, a_{ij})$ are computed by (2.3).

Step 3. The dispatch probabilities are updated using

$$f_{ij}^t \approx Q_j(\{s_{(k)j}\}, \rho^{t-1}, k)(1 - (r_i^t)^{s_i}) \prod_{u=1}^{k-1} (r_{(u)j}^t)^{s_{(u)j}}. \tag{2.14}$$

The resulting dispatch probabilities do not necessarily satisfy equation (2.2). Therefore, they should be normalized using

$$f_{ij}^t \leftarrow f_{ij}^t \frac{(1 - P_s^t)}{\sum_{i \in I} f_{ij}^t}. \tag{2.15}$$

for every $i \in I$ and $j \in J$. Next, the system-wide mean service time, mean offered load and mean server utilization are updated to

$$\tau^t = \frac{1}{\lambda(1 - P_s)} \sum_{j \in J} \lambda_j \sum_{i \in I} f_{ij}^t \tau_{ij}, \tag{2.16}$$

$$\rho^t = \frac{\lambda \tau^t}{s} \tag{2.17}$$

and

$$r^t = \frac{1}{s} \sum_{i \in I} s_i r_i^t. \tag{2.18}$$

Step 4. Check if $|r_i^t - r_i^{t-1}| < \varepsilon$ holds for all $i \in I$ for a given $\varepsilon > 0$, then terminate. Otherwise, set $t \leftarrow t + 1$ and return to ***Step* 1**.

The r_i factors in the procedure above converge to the busy probabilities associated with the dispatching policy represented by a_{ij}. These busy probabilities are then used in the mixed integer programming model introduced in the next section to optimize the nine minute coverage for high-priority calls while maintaining a balanced offered load for servers and accommodating multiple servers per station. Although the convergence of the procedure is not guaranteed in the general case, a restricted version of the method described in Budge et al. (2009) is guaranteed to converge. Note that the iterative procedure converged in all the experiments performed in this study.

2.3 The Maximum Expected Coverage with Balanced Load Problem (MEXCBL)

The proposed model, MEXCBL, aims to find the optimal way to locate s servers at a set of potential station locations W, and it assigns the demand nodes to the opened stations, thus forming a preference list such that the nine-minute coverage is maximized. Each demand node, j, is assigned an ordered list of opened stations, such that a server from the most preferred station in the list that has an available server responds to a call generated from j. If there are more than one server available at that station, one of the available servers is randomly chosen and dispatched. Two of the important features of MEXCBL are (1) balanced workload among the servers and (2) contiguity in the servers' first priority districts. The computational example in Section 2.4 successively adds these two features to a base model, thus leading to three models for comparison.

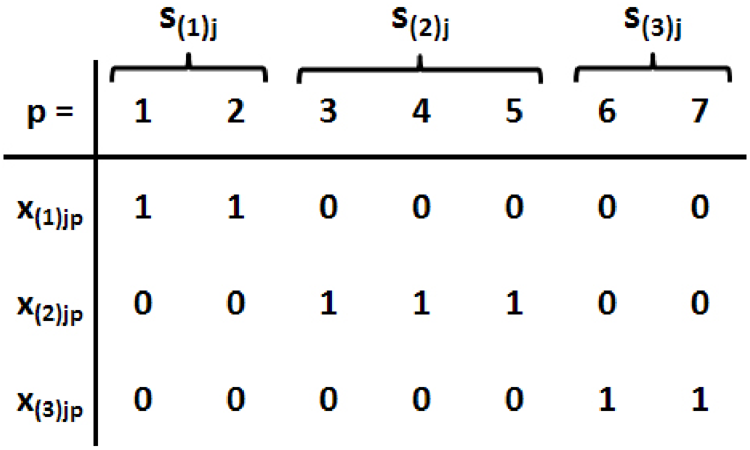
The variables z_{wjpm} capture the dispatching policy, and these variables are defined as follows.

$$z_{wjpm} = \begin{cases} 1 & \text{if there are } p-1 \text{ servers located at stations} \\ & \text{that node } j \text{ prefers over } w \text{ and there are } m \quad \forall w \in W, \forall j \in J, \forall p = 1, ..., s, \\ & \text{servers located at station } w, \qquad\qquad\qquad\qquad\qquad \forall m = 1, ..., \kappa_{wp}, \\ 0 & \text{otherwise,} \end{cases} \tag{2.19}$$

where $\kappa_{wp} = \min(c_w, s-p+1)$. The variable x_{wjp} captures the demand node/station assignments and the preferences associated with those assignments:

$$x_{wjp} = \begin{cases} 1 & \text{if } p' < p \leq s - p'' \text{ where } p' \text{ is the number of} \\ & \text{servers located at stations that node } j \text{ prefers} \\ & \text{over } w, \text{ and } p'' \text{ is the number of servers located} \quad \forall w \in W, \forall j \in J, \\ & \text{at stations that node } j \text{ prefers less than } w, \qquad \forall p = 1, ..., s, \\ 0 & \text{otherwise.} \end{cases}$$

Note that variables x_{wjp} and z_{wjpm} are defined such that the model can locate multiple servers per station. The number of servers located at station w is specified by the decision variables y_w, $w \in W$, in the model. Also, the real-valued and non-negative decision variables o_w capture the total offered load for servers at station $w \in W$.



	$s_{(1)j}$		$s_{(2)j}$			$s_{(3)j}$	
p =	1	2	3	4	5	6	7
$x_{(1)jp}$	1	1	0	0	0	0	0
$x_{(2)jp}$	0	0	1	1	1	0	0
$x_{(3)jp}$	0	0	0	0	0	1	1

Figure 2.1: Illustration of an example with the values of x_{ijp} for a particular customer location j. There are 7 servers and the number of servers at each station are shown on the top.

It is worth comparing the variable x_{wjp} to the preference list b_{kj}. The main difference is that x_{wjp} describes servers while b_{kj} describes stations: the index p in x_{wjp} grows as we encounter servers that are located at stations, whereas the index k in b_{kj} grows with stations regardless of the number of servers located at those station. For example, assume that there are 7 servers in the system ($s = 7$) which are located at 3 stations ($|I| = 3$) and there are 2, 3, and 2 servers at the first, second and third preferred stations for demand node j, respectively ($s_{b_{1j}} = s_{(1)j} = 2$, $s_{b_{2j}} = s_{(2)j} = 3$ and $s_{b_{3j}} = s_{(3)j} = 2$). Let $x_{(k)jp}$ be equal to x_{wjp} when $a_{wj} = k$. The corresponding values for $x_{(k)jp}$ are shown in Figure 2.1. There are 2 servers at the station that is more preferred than $b_{(2)j}$ and 2 servers at the station that is less preferred than $b_{(2)j}$. Therefore, $x_{(2)jp}$ is 1 when $3 \leq p \leq 5 = s - 2$, as shown in Figure 2.1.

The MILP model for MEXCBL is formally stated.

$$\max \sum_{w\in W}\sum_{j\in J}\sum_{p=1}^{s}\sum_{m=1}^{\kappa_{wp}} h_{wjpm} z_{wjpm} \tag{2.20}$$

subject to

$$\sum_{p=1}^{s}\sum_{m=1}^{\kappa_{wp}} z_{wjpm} \leq 1 \qquad j \in J, w \in W \tag{2.21}$$

$$\sum_{p=1}^{s}\sum_{m=1}^{\kappa_{wp}} z_{wjpm} \leq y_w \qquad j \in J, w \in W \tag{2.22}$$

$$x_{wjp'} = \sum_{p=\max(1,p'-c_w+1)}^{p'} \sum_{m=p'-p+1}^{\kappa_{wp}} z_{wjpm} \qquad j \in J, w \in W, p' = 1, ..., s \tag{2.23}$$

$$\sum_{w\in W} x_{wjp} = 1 \qquad j \in J, p = 1, ..., s \tag{2.24}$$

$$\sum_{p=1}^{s} x_{wjp} = y_w \qquad j \in J, w \in W \tag{2.25}$$

$$y_w \leq c_w \qquad w \in W \tag{2.26}$$

$$\sum_{w\in W} y_w = s \tag{2.27}$$

$$o_w = \sum_{j\in J}\sum_{p=1}^{s}\sum_{m=1}^{\kappa_{wp}} (\lambda_j^H + \lambda_j^L) q_{jpm}(1-r^m)r^{p-1}\tau_{wj} z_{wjpm} \qquad w \in W \tag{2.28}$$

$$o_w \geq (r-\delta) y_w \qquad w \in W \tag{2.29}$$

$$o_w \leq (r+\delta) y_w \qquad w \in W \tag{2.30}$$

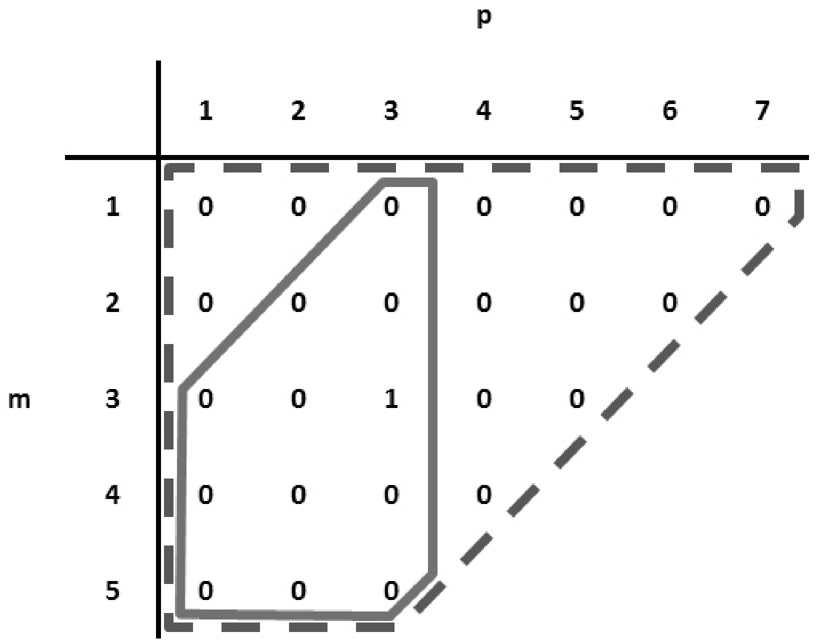
$$x_{wj'1} \geq x_{wj1} \qquad j \in J, w \in W, j' \in N_{wj} \tag{2.31}$$

where

$$h_{wjpm} = q_{jpm}(1-r^m)r^{p-1}\lambda_j^H R_{wj}. \tag{2.32}$$

The objective function in (2.20) reflects the high-priority coverage level. The coefficient h_{wjpm} is composed of two parts. The first part is the probability that a server from station w is dispatched to a call from location j. It is computed as the probability that the $p-1$ servers at stations which are more preferred than w are all busy (r^{p-1}) times the probability that at least one of the m servers at station w are available ($1-r^m$). The effect of the independence assumption is obviated by multiplying this product with the correction factor q_{jpm}; the lowercase q here reflects that it is a MILP input parameter that is treated as a constant, whereas the uppercase Q in the previous section reflects the queueing dynamics. The second part in computing the coefficient h_{wjpm} is the proportion of high priority calls from location j that can be covered within the time limit (9 minutes here) by a server from station w ($\lambda_j^H R_{wj}$). Thus, h_{wjpm} captures the partial coverage that is obtained if a station w with m servers is assigned to be station j's pth preferred station. Note that the correction factors q_{jpm} and the components r^m and r^{p-1} are precomputed for all $j \in J$, $p = 1, ..., s$ and $m = 1, ..., c_j$ based on the mean server utilization r and are treated as constants in the model. Therefore, the objective function is a linear function of z_{wjpm} variables. Similarly, all the constraints (2.21)-(2.31) are linear functions of the model variables.

Next, we describe the constraints in MEXCBL. The first set of constraints, (2.21), guarantees that a customer location is not assigned to a station more than once. The second set of constraints (2.22) ensures that a station is assigned to a customer location only if that station is open. Consider a system similar to that of Figure 2.1 with $s = 7$ servers where w is the second preferred station for j ($a_{wj} = 2$) with 3 servers ($y_w = 3$). Assume that there



	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	
3	0	0	1	0	0		
4	0	0	0	0			
5	0	0	0				

Figure 2.2: The bounds for the sums in (2.21) and (2.22) (the dotted lines) and the bounds for the sums in (2.23) when $p' = 3$ (the solid lines) over m and p . The example used is the same as the one in Figure 1.

are 2 servers located at w', station j's first preferred station ($y_{w'} = 2$), and the capacity of station w is 5 ($c_w = 5$). The corresponding terms included in the sums in (2.21) and (2.22) are shown in Figure 2.2 with the dotted box.

The variables x_{wjp}, set via (2.23), are included in the model to facilitate explaining and understanding the model. These variables and the constraint (2.23) could be eliminated by substituting (2.23) into (2.24) and (2.25). The bounds for the sums in (2.23) are shown by the solid box in Figure 2.2. To have a preference list, every customer location should have an rank ordering of stations which contains all the preferences, which is enforced by (2.24). Note that these preferences are formed around the servers instead of stations due to the definition of x_{wjp}. The number of ones in x_{ijp} for $j \in J$, $w \in W$ and $p = 1, ..., s$ should be the same as the number of servers located at j (y_j), which is imposed by (2.25). Moreover, (2.26) ensures that the number of servers located at a station does not exceed the station capacity and (2.27) makes sure that the number of located servers is equal to the number of available servers.

The server utilization o_w for server w captured by (2.28) is constrained by lower and upper limits in (2.29) and (2.30) to balance the workload among the servers within a tolerance of δ relative to the mean server utilization. Each term in (2.28) has a coefficient that is composed of two parts. The first part is the probability that a server from station w responds to a call from j ($q_{jpm}(1-r^m)r^{p-1}$ where there are $p-1$ servers at stations that j prefers over w and there are m servers at w). The second part is the utilization of the servers at w when they respond to calls from j. Balancing the workload could lead to non-contiguous dispatching regions. To avoid this problem, contiguity for the first priority districts is enforced by the method that is suggested by Mehrotra et al. (1998). Constraint set (2.31) allows w to be the first preferred station for j only if there is a neighbor of j, namely j', whose first preferred station is w and is geographically closer to w than j. We assume that the customer locations are small enough that at most one station is located within each of them, i.e., $\forall w, w' \in W, w \neq w' : g_w \neq g_{w'}$ where g_w denotes the customer location in which station w is located.

Three different versions of MEXCBL are compared in Section 2.4:

1. **Base:** base model without load-balancing or contiguity. This model contains constraints (2.21-2.27) and excludes the load balancing and contiguity constraints, (2.28-

2.31).

2. **LBM**: a model with the load-balancing but without contiguity. This model includes constraints (2.21-2.27) as well as the constraints that balance the offered loads, (2.28-2.30).

3. **LBCM**: the full model with load-balancing and contiguity, containing constraints (2.21–2.31).

One can show that the dispatching policy for the Base model is contiguous when servers (ambulances) have been pre-located at stations such that no more than one server located at each station with the additional assumptions that the travel time distribution is non-decreasing with distance, which implies that travel time is merely a function of the distance between all j and w. This leads to values of R_{wj} that are monotonically decreasing with spatial distance, which in turn leads to first order districts that are spatially contiguous.

The following results describe the optimal dispatching policies for this restricted version of the Base model. Theorem 1 indicates that the optimal solution to the Base model always sends the closest server. Theorem 2 indicates that the policy of sending the closest server leads to contiguous first-priority districts, and Corollary 1 shows that the first priority districts for the Base model are contiguous when there is at most one server per station. The results hold for any set of server locations, however, they cannot be used to identify the optimal server locations.

Theorem 1 *Let D_{wj} denote the distance between demand node j and station w and assume that the travel times are captured by a monotonically decreasing function of distance for all* *pairs of j and w, i.e., $R_{wj} = f(D_{wj})$. Then, the first priority districts for the Base model when there is no more than one server located at a station are equivalent to a 'send-the-closest-server' policy.*

Theorem 2 *The 'send-the-closest-server' policy results in contiguous first priority districts.*

Corollary 1 *The first priority districts for the Base model are contiguous if there is no more than one server located at every station.*

The proofs for Theorem 1 and Theorem 2 can be found in the Appendix B. Corollary 1 follows directly from Theorems 1 and 2. If some stations have more than one server, then (2.31) may be needed to maintain contiguity in the Base model. Discontiguity did not occur in any of the Base model scenarios presented in Section 5, and therefore, (2.31) did not need to be added to the Base model.

2.3.1 Iterative Procedure

The MILP model proposed in the previous section uses the system-wide mean server utilization, r, to compute the coverage and server utilizations in (2.20) and (2.28), respectively, since the model approximately balances the load by equalizing the server utilizations. The mean utilization depends on the current dispatching policy and changes as a result of the optimization process. Therefore, we employ an iterative procedure to recompute the mean utilization and update the correction factors. For clarity, we index all of the variables by iteration t. The steps of the procedure are as follows.

Step 0. Choose an initial dispatching policy, such as the 'send-the-closest-server' policy. In this chapter, the initial dispatching policy is formed by solving a simplified version of the MILP proposed in the previous section, where the servers are assumed to be independent. Thus, all the correction factors q_{wjp} are set to 1 for $w \in W, j \in J, p = 1, ..., s$. The initial mean server utilization r is computed as $r = \lambda\overline{\tau}/s$ where $\overline{\tau} = \lambda^{-1} \sum_{w \in W} \sum_{j \in J} \lambda_j \tau_{wj}$ assuming that $P_s = 0$. In addition, the server utilization imbalance tolerance δ is set to 1.0, thus the load balancing constraint is not enforced. Using the resulting initial dispatching policy, z^0_{wjpm}, and the procedure explained in Section 2.2, we compute the initial server utilization r^0 and correction factors q^0_{wjpm}. Moreover, the initial server utilization imbalance tolerance δ^0 is set to the smallest tolerance value that encompasses all the server utilizations r_w, with $\delta = (\max r_w - \min r_w)/2$. Set t to 1.

Step 1. Solve the MILP model with the most recent values for server utilization (r^{t-1}) and correction factors (q^{t-1}_{wjpm}) to find the new dispatching policy z^t_{wjpm}.

Step 2. Form the preference lists a^t_{ij} and b^t_{kj} and the server distributions s^t_i using the z^t_{wjpm} variables (see the pseudo-code following this procedure.) Using these new values, compute the new server utilization r^t and the new correction factors q^t_{wjpm} using the algorithm in Section 3. We reduce the server utilization imbalance margin by reducing the imbalance tolerance δ^t to be equal to the standard deviation of server utilizations in z^{t-1}_{wjpm}, which ensures a shrinking margin. We then update the values for the mean server utilization, the correction factors and the server utilization imbalance tolerance in the MILP model.

Step 3. There are two termination criteria. The first is infeasibility of the MILP model that can occur when constraints (2.29) and (2.30) cannot be satisfied, i.e., reducing the load imbalance any further leads to infeasibility. The second condition is reaching a server utilization imbalance level that is less than an acceptable server utilization imbalance threshold δ. If the termination criteria is not met, then increase t by 1 and go to ***Step* 1**.

Step 4. Derive the preference lists a_{ij} and b_{kj} and the distribution of servers from the optimal dispatching policy z_{wjpm}.

The iterative procedure never enters a cycle where two or more solutions to the MILP are repeated in an infinite loop, since the server utilization imbalance tolerance, δ, is reduced after every iteration. Therefore, the procedure is guaranteed to reach a stage where either the IP model is infeasible or the server utilization imbalance level falls below δ.

The preference lists a_{ij} and b_{kj} are based on the dispatching policy that is found by the IP in ***Step* 2** and ***Step* 4**. This procedure is captured by the following pseudo-code.

Pseudo-code for forming the preference lists:

```
for all $j \in J$ do
    $p \leftarrow 1$ and $p' \leftarrow 1$
    while $p \leq s$ do
        $(w, m) \leftarrow$PREFERENCE($j$,$p$)
        $a_{wj} \leftarrow p'$, $b_{p'j} \leftarrow w$, $s_w \leftarrow m$, $p \leftarrow p + m$, $p' \leftarrow p' + 1$
    end while
end for
```

where

```
function PREFERENCE($j$,$p$)
    for all $w \in W$ do
        for $m = 1 \to c_w$ do
            if $z_{wjpm} = 1$ then return ($w$,$m$)
            end if
        end for
    end for
end function
```

Both Base and LBCM are NP-complete in the strong sense. The proofs for NP-completeness are presented in the Appendix.

2.4 Computational Results

The model is illustrated using real-world data from Hanover County, Virgina, a semi-rural, semi-suburban county in the metropolitan Richmond area. The data contains the history of emergency calls during a period of 19 months, which provides response times, service times, and call arrival rates needed for the input parameters. The county has fifteen stations and is divided into 175 two mile by two mile cells plus 36 one mile by one mile cells, which are the nodes where calls arise. The cells and the distribution of demand among them averaged across all days and times are illustrated in Figure 2.3. The potential stations are marked by the black dots. We construct eight scenarios corresponding to each combination of day—

weekdays (WD) or weekends (WE)—and time, where a day is divided into four six-hour periods labeled as 12am6am, 6am12pm, 12pm6pm, and 6pm12am. Weekdays start from Sunday 6pm and end at Friday 6pm. Likewise, weekends are from Friday 6pm to Sunday 6pm. Figures C.1 and C.2 in Appendix C show the changes in demand across the eight time periods.

The proposed iterative procedure and MILP are evaluated using the data described above. Six available servers ($s = 6$) are located at 15 stations in the experiments presented in this section, unless otherwise is stated. All station capacities are set to two servers per station. Parameters N_{wj} and g_w are formed based on the map shown in Figure 2.3. Recall that we compare three different versions of the model in this section, the base model without load-balancing (Base), a model with load-balancing but without the contiguity constraint (LBM), and a model with load-balancing and contiguity constraints (LBCM).

All computations were performed on a 2GHz quad core CPU with 3GB of RAM. The iterative procedure for solving the Hypercube model approximation were implemented and run in Python and the MILP models were solved using Gurobi 5.0.0. The average total running times (including multiple solutions to the MILP as the parameters are updated iteratively) for Base, LBM and LBCM were 3, 35 and 69 minutes, respectively. The average number of iterations performed by the MILP were 1, 7.25 and 6.25 for Base, LBM and LBCM respectively. Moreover, the iterative approximate queueing model introduced in Section 2.2 converged in 1 iteration within every iteration of MILP that was run throughout this work.

We confirm the analytical results by simulation, which we present at the end of this section.

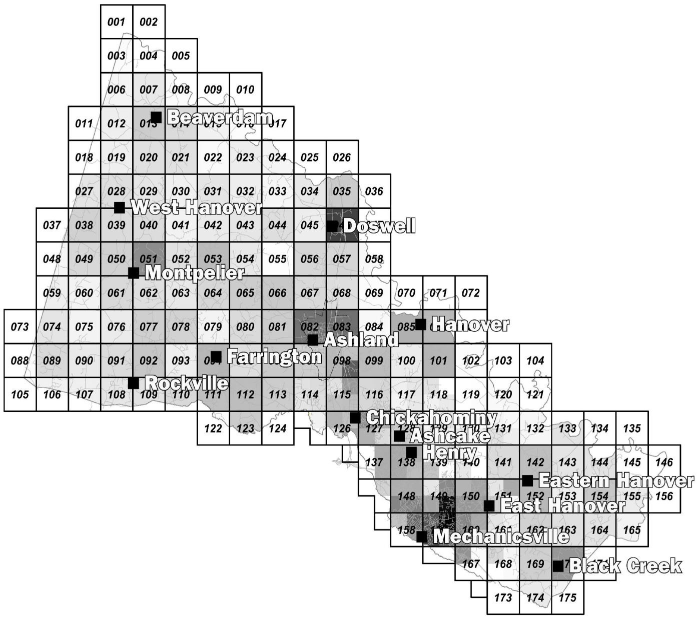


Figure 2.3: The distribution of demand among the cells aggregated across all eight scenarios. Red indicates a higher call volume while white represents the cells with very small or zero demand.

The simulation considers a $M/M/s/s$ queue with $s = 6$, which matches the assumptions made in Budge queueing model (see Section 2.2). The true distribution of the inter-arrival times is a mixture of two lognormal distributions, corresponding to the calls where a patient is and is not transferred to a hospital. The probability of transfer and the lognormal distributions are computed based on historic data. For each time period, 30 runs of simulation are performed, and each simulation is run until 10000 calls are served.

The MILP results across the three model variations (Base, LBM, and LBCM) illustrate how server locations and district boundaries change under different operating criteria. Figures 2.4-2.6 depict the optimal stations to open and the first priority districts are depicted for Base, LBM and LBCM, respectively, for the WD6pm12am period. Six stations are opened in Base and one server located at each one of them. On the other hand, there are five open stations in LBM and LBCM with two servers located at *Ashland* station and one server located at every other open station.

First, we aggregate all eight scenarions to illustrate the importance of the load balancing and contiguous constraints. Figure 2.4 illustrates the server locations and first priority districts in the Base model. Without the contiguity constraints, the first priority districts in LBM contain many discontiguities, as shown in Figure 2.5. The optimal solution to LBCM balances the workload and maintains contiguous first priority districts, as shown in Figure 2.6. Figure 2.7 illustrates the offered loads associated with the three models.

The coverage level for LBCM is always lower than the coverage levels for Base and LBM. However, the examples suggest that the coverage loss due to balancing the load and main-

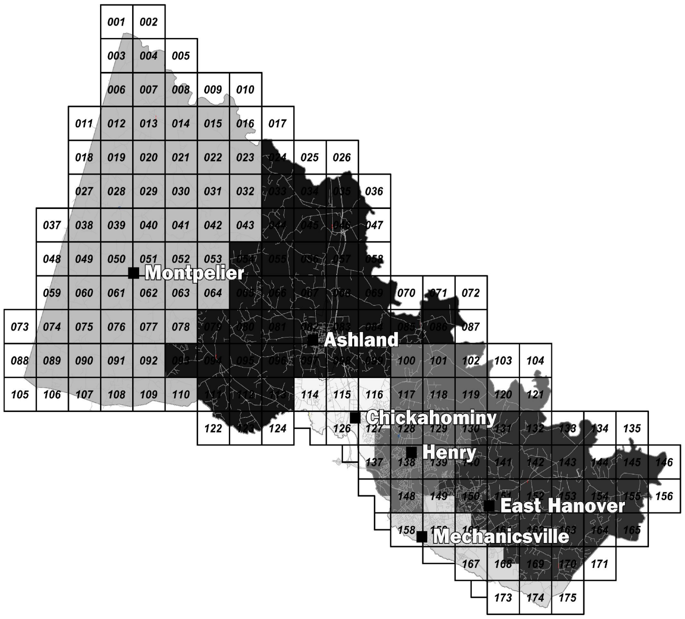


Figure 2.4: The first priority districts and the open stations when six servers are located and assigned to demand nodes for Base on WD6pm12am.

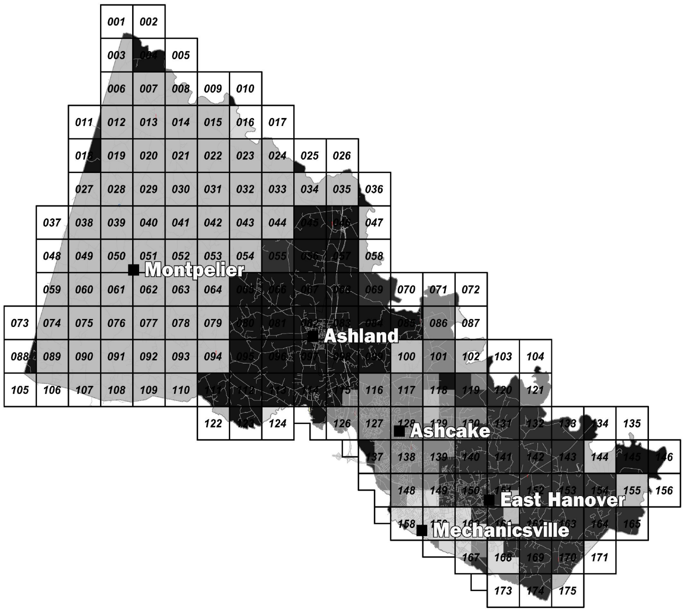


Figure 2.5: The first priority districts and the open stations when six servers are located and assigned to demand nodes for LBM on WD6pm12am.

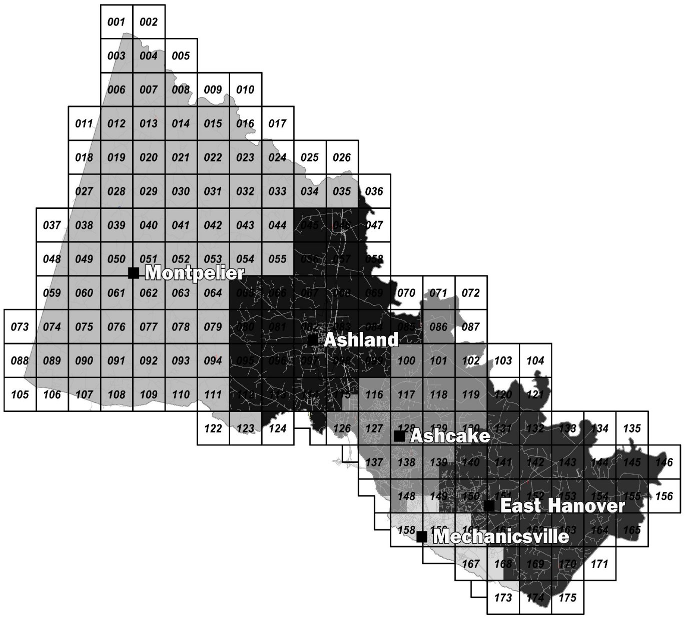


Figure 2.6: The first priority districts and the open stations when six servers are located and assigned to demand nodes for LBCM on WD6pm12am.

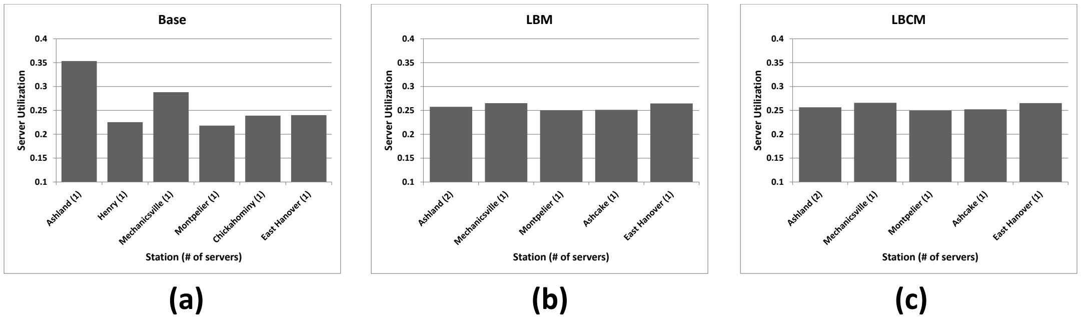


Figure 2.7: Server utilizations for WD6pm12am for (a) Base (b) LBM (c) LCBM.

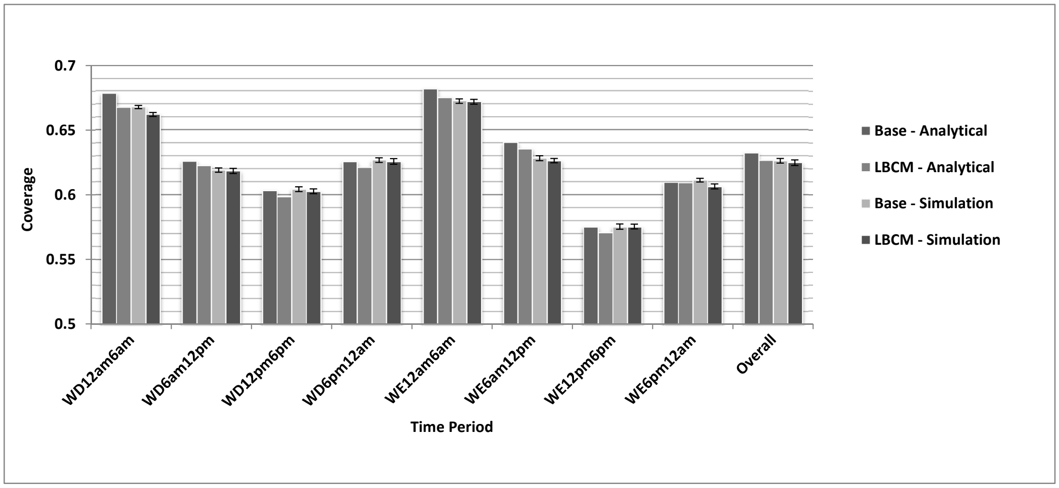


Figure 2.8: The coverage level during each time period for Base and LBCM, which are computed both numerically via optimization and using simulation. For the simulation results, the 95% confidence intervals are shown by the black bars.

taining contiguity is extremely small. Figure 2.8 shows the nine-minute coverage for high priority calls during different time periods. The first two bars in each cluster indicate the coverage level for Base and LBCM, derived numerically by the model (objective value of the MILP). The coverage levels for LBCM are lower than the coverage levels for Base.

We note that the assumption of a common server utilization for the Base model is inaccurate. To address this issue, we retrospectively recompute the server utilizations after solving the MILP through the iterative procedure to more accurately capture the server utilizations. Figure 2.7 shows these recomputed server utilizations for the WD6pm12am time period. It shows that the server utilizations for the Base model solutions range from 0.22 to 0.35, whereas the server utilizations for LBM and LBCM are all approximately 0.26. The unequal server utilizations in the Base model lead to inflated coverage levels, since some servers receive a disproportionate amount of the total offered load. The simulation reflects the actual queueing dynamics and thus reflects a more accurate coverage level, and therefore, Figure 2.8 shows the simulated coverage level for Base and LCBM together with their 95% confidence intervals (the third and fourth bars). For both Base and LBCM, the difference between the numerical and simulated coverage levels relative to the MILP coverage level are no more than 2% of the MILP solution values. Note that these values are within the 2% error associated with the Hypercube model approximations. The simulated coverage for the LCBM solutions are, on average, within 0.1% of the MILP solution values. The maximum and average simulated difference between the coverage levels between a Base scenario and the corresponding LBCM solution are 0.58% and 0.2% respectively, which again suggest that the reduction in the coverage level when the workload is balanced may be minimal in practice.

Next, we discuss the server locations and dispatching policies via the districts. Table 2.2 and Figure 2.9 shows the number of servers located at each station and the first priority districts for different time periods in LBCM, respectively. The server locations change in response to the changes in the distribution of calls in the regions. The sources of these changes vary, and they include interstate travel, activities at an amusement park, a regular "workday" pattern of people being at work, and people being at home in evenings, overnight, and on weekends. Appendix C describes these changes in demand in greater detail.

Figure 2.9 illustrates the first priority districts for the LBCM solutions across all the time periods. The *Montpelier* and *East Hanover* districts, which have the lowest offered

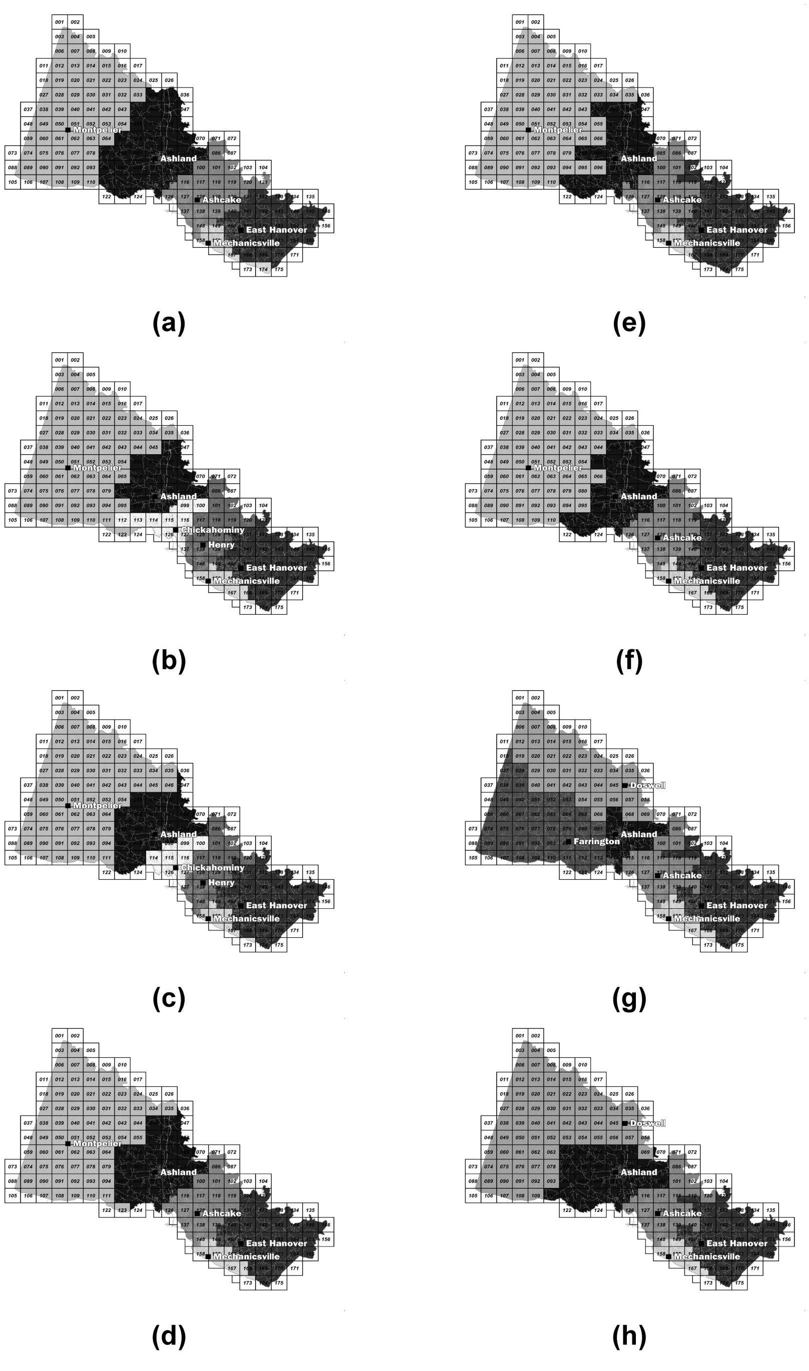


Figure 2.9: The figure illustrates the first priority districts and the open stations in LBCM at (a) WD12am6am (b) WD6am12pm (c) WD12pm6pm (d) WD6pm12am (e) WE12am6am (f) WE6am12pm (g) WE12pm6pm (h) WE6pm12am. The total number of available servers is six.

Table 2.2: The table shows the number of servers located at different stations for each time period. The times correspond to the beginning of the time period. Each column sums to six since total number of available servers is six.

Station	Weekdays (WD)				Weekends (WE)			
	12am	6am	12pm	6pm	12am	6am	12pm	6pm
Ashland	2	1	1	2	2	2	1	2
Beaverdam	0	0	0	0	0	0	0	0
Eastern Hanover	0	0	0	0	0	0	0	0
Doswell	0	0	0	0	0	0	1	1
Hanover	0	0	0	0	0	0	0	0
Henry	0	1	1	0	0	0	0	0
Mechanicsville	1	1	1	1	1	1	1	1
Montpelier	1	1	1	1	1	1	0	0
Rockville	0	0	0	0	0	0	0	0
Chickahominy	0	1	1	0	0	0	0	0
Farrington	0	0	0	0	0	0	1	0
Black Creek	0	0	0	0	0	0	0	0
Ashcake	1	0	0	1	1	1	1	1
East Hanover	1	1	1	1	1	1	1	1
West Hanover	0	0	0	0	0	0	0	0

loads in the Base model, cover larger geographic areas in the LBM and LBCM solutions. On the other hand, the area surrounding the *Mechanicsville* station has high call volumes, and therefore, it covers the smallest geographic area. The *Chickahominy* district in the Base model solution is reallocated to two stations in the LBCM solution. One part is merged with the district on its east, *Henry*, to form a new district which is covered by a server located at the *Ashcake* station. Furthermore, the call locations in the western part of *Chickahominy* district are added to the *Ashland* district. The *Ashland* district stations has two servers in the LBCM solution. The solutions highlight the importance of the *Ashland* station in the regionwide coverage due to its central location in the county. Servers located at this station can effectively provide backup coverage to other districts in the county, reflected in the second through sixth priority districts.

Figure 2.10 illustrates the sensitivity of the Base, LBM, and LBCM coverage levels with respect to the number of servers s. The coverage levels for Base, LBM and LBCM are very close in both the numerical and simulation results. For the numerical results, the differences between the coverage level of Base and LBCM are less than 1%. The only exception is when $s = 8$, where the coverage of Base is higher than LBM and LBCM by 1.5% in the numerical results. This is due to the larger variation in the utilizations in Base when $s = 8$ compared to cases with fewer servers (e.g., a standard deviation of 0.063 when $s = 8$ versus 0.051 when $s = 6$). Due to a larger deviance between the actual and assumed workloads in Base, balancing the load leads to a larger decline in the coverage. Similarly, the simulation results show that Base has 0.5% higher coverage level than LBCM when $s = 8$. On the other hand, the difference between the coverage levels of Base and LBCM is less than 0.3% when the number of servers is less than 8. The overall pattern shows that the cost of enforcing load balancing and contiguity on the coverage is very small across different numbers of servers and is not sensitive to s. The simulation results deviate from the analytical results by no more than 1.4% across all scenarios.

2.5 Conclusions

This chapter proposes a linear MIP model that simultaneously locates ambulances and dispatches ambulances to patients. The model maximizes the coverage level, the expected proportion of high-priority calls that are covered within nine minutes, while maintaining a

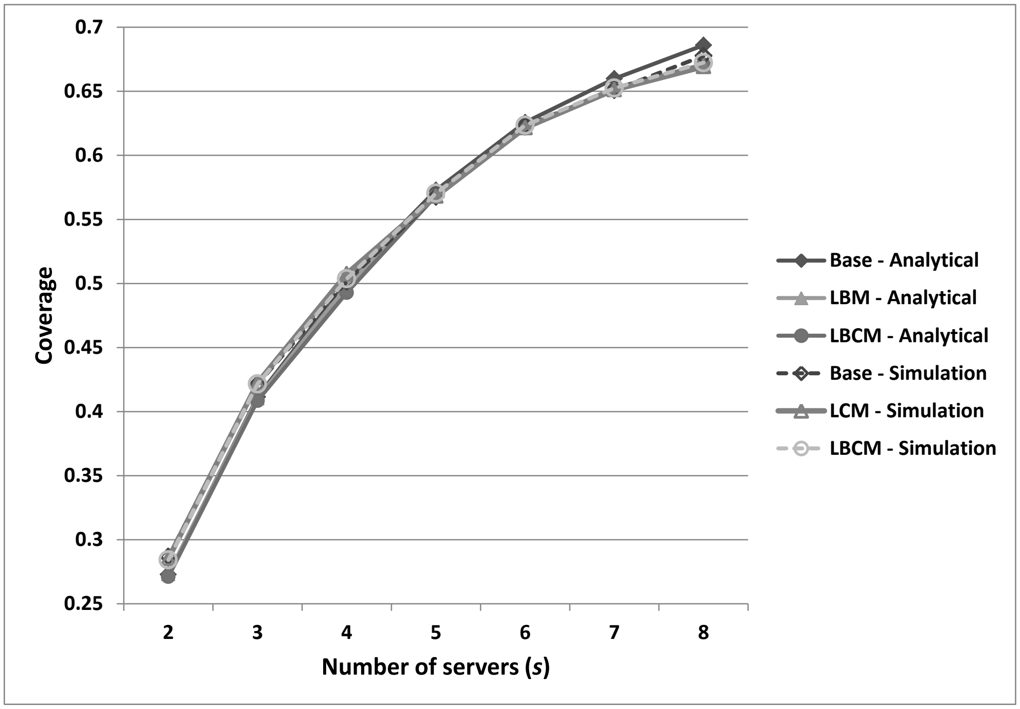


Figure 2.10: The figure illustrates the coverage level for different numbers of servers at WD6pm12am for the analytical and simulated results. The solid lines show the analytical and the dashed lines show the simulation results.

balanced load and contiguity in the first priority response districts. The model simultaneously locates ambulances at a set of potential stations and forms a preference list assigning the demand nodes to those stations. The model also takes into account the stochastic aspects of dispatching such as stochastic travel times and server availabilities. Moreover, the model can locate multiple servers per location, resembling the reality of EMS systems. The experiments show that balancing the workload amongst the servers leads to dispatching districts that are not contiguous. The additional constraints for load balancing and contiguity lead to a lower objective value and thus a lower coverage than in the Base model. However, the experimental results suggest that the proposed model effectively balances the offered load and maintains contiguity with a minimal negative effect on the coverage.

The proposed iterative procedure updates the model's stochastic factors, such as busy probabilities and correction factors. Next, the model uses these factors to find an optimal dispatch policy. This policy is then used to reevaluate and update the stochastic factors. The process repeats iteratively until the procedure converges to a solution. The solution is potentially sub-optimal due to the queueing model approximation and due to the iterative procedure.

The model can be extended in several ways. First, the possibility of incorporating reliability factors into the model to ensure a minimum coverage level at every call location Ball and Lin (1993) can be investigated. Erkut et al. (2008a) reviewed models that use maximum reliability/availability as their objective function. The thesis concludes that the models that maximize coverage are more effective than those maximizing reliability. However, it is sensible to build a model which maximizes the coverage level while ensuring that every customer receives service with a minimum reliability level. That would result in a model that is more *fair* to the customers while it maintains a balanced load. A second extension could consider the impact of using multiple server types to respond to calls. There are cases where several types of servers are dispatched to respond to a single call (e.g., many EMS departments use both ambulances and fire engines for EMS calls). Third, this model could be extended to integrate traffic and weather information into the model to effectively update the dispatch policy in real-time.

Chapter 3

Queueing Model for Server Dispatching with Multiple Servers per Location and Multi-Server Dispatches

3.1 Introduction

We start by introducing the notation that is used throughout this chapter, shown in Table 3.1, which is an extension of the notation that was used in the previous chapter. We often replace the index i with $(k)j$ which indicates the k^{th} preferred station for customer j, e.g., $s_{(k)j}$ is the number of servers at customer j's k^{th} priority station. When we use k, it should be interpreted as the priority of station i in the preference list of customer j where i and j can be determined from the context.

In this chapter, we first introduce an iterative procedure to compute the steady-state probabilities for an $M[G]/M/s/s$ queueing model, which we define as an extension to the $M/M/s/s$ model with multi-server dispatches where G represents a general probability mass function (pmf) for the number of servers that each incoming call requests. Then, we derive

Table 3.1: Summary of the Symbols

Symbol	Description	Domain
J	Set of all customer (demand) nodes.	
I	Set of all open stations.	
a_{ij}	Preference of server i for responding to a call from node j in the dispatching policy.	$i \in I$, $j \in J$
b_{kj}	kth preferred station for calls from node j in the dispatching policy.	$k = 1, ..., \|I\|$, $j \in J$
s	Total number of servers in the system.	
s_i	Number of servers at station i ($\sum_{i \in I} s_i = s$).	$i \in I$
$s_{(k)j}$	The number of servers at the k^{th} priority station for customer i, i.e., $s_{(k)j} = s_{b_{kj}}$.	$k = 1, ..., \|I\|$, $j \in J$
λ_j	Call arrival rate for customer j.	$j \in J$
λ	System-wide total call arrival rate with $\lambda = \sum_{j \in J} \lambda_j$.	
τ_{ij}	Mean service time for calls originated from node j and served by a server from station i.	$i \in I$, $j \in J$
τ	System-wide mean service time.	
ρ	System-wide mean offered load per server.	
r	System-wide mean server utilization.	
r_i	Utilization factor for a server located at station i.	$i \in I$
k	The priority of the current station i responding to the current call from j, i.e., $k = a_{ij}$.	
z_k	The number of servers at the k most preferred station for the current call.	$k = 1, ..., \|I\|$
A_{ijm}	The event that station i dispatches m servers to a call from customer i.	$i \in I$, $j \in J$, $m = 0, ..., s_i$
C_{ijl}	The event that the stations that are more preferred than i for customer j dispatch l servers to a call from j.	$i \in I$, $j \in J$, $l = 0, ..., s$
D_{ijm}	The event that station i dispatches m servers to a call from customer j assuming an $M[G]/M/s/s$ model.	$i \in I$, $j \in J$, $m = 0, ..., s_i$
E_{ijl}	The event that the stations that are more preferred than i for customer j dispatch l servers a to call from j assuming an $M[G]/M/s/s$ model.	$i \in I$, $j \in J$, $l = 0, ..., s$
R_d	The event that d servers are requested by the current call.	
$d_{\max}$	The maximum number of servers that is requested by a single call, i.e., $Pr\{R_d\} = 0$ for $d > d_{\max}$.	
Z_n	The event that there are exactly n busy servers in the system.	$n = 0, ..., s$
Z_n^i	The event that there are exactly n busy servers at station i.	$n = 0, ..., s_i$
f_{ijm}	Probability that station i dispatches m servers to a call from customer j, i.e., $f_{ijm} = Pr\{A_{ijm}\}$.	$i \in I$, $j \in J$, $m = 0, ..., s_i$
p_{ijm}	Probability that station i dispatches m servers to a call from customer j assuming an $M[G]/M/s/s$ model, i.e., $p_{ijm} = Pr\{D_{ijm}\}$.	$i \in I$, $j \in J$, $m = 0, ..., s_i$
P_n	Probability that there are exactly n busy servers in the system, i.e., $P_n = Pr\{Z_n\}$.	$n = 0, ..., s_i$
Q_{ijm}	The correction factor for when station i dispatches m servers to customer j.	$i \in I$, $j \in J$, $m = 0, ..., s_i$
ε	Server utilization deviation tolerance.	

the expressions for the server utilizations and dispatch probabilities in Section 3.3, using the correction factors that are introduced in Section 3.4. We use the results from the previous sections and integrate them into an iterative procedure that computes the queueing measures for the dispatching models with multi-server dispatches. We discuss the simulation and data that are used to evaluate the proposed model and present the results of the evaluations in Section 3.6. Finally, the chapter is concluded in Section 3.7.

3.2 Steady-State Probabilities for a $M[G]/M/s/s$ Model

Two of the most common queueing models are the $M/M/s/s$ and $M/M/s/\infty$ models with exponential inter-arrival and service times and a queue length of zero and infinity, respectively. However, it often happens in the emergency systems that some of the arriving calls request multiple servers depending on the severity of the incident. For example, an EMS dispatcher might receive a request to dispatch multiple servers to a severe car accident with several casualties on a highway. Likewise, the fire engines and police cars are usually dispatched in bulks in response to sever fire and 911 calls. Such models cannot be analyzed as an $M/M/s$ model due to the difference in the arrival patterns. As a result, we introduce an extension to the $M/M/s$ model that allows for multi-server dispatches. Namely, a $M[G]/M/s$ model is a queueing model with s servers, Poisson arrivals, exponential service times, and G denotes a general pmf function for the number of servers that are requested per incoming call.

We will only focus on the zero-length case in this thesis since that is the assumption used by most of the emergency systems (Iannoni and Morabito (2007)), i.e., the calls that arrive when all the servers are busy will be served by backup vehicles or by servers from the neighboring districts. The $M[G]/M/s$ model allows for partial dispatches when there are not enough available servers in the system to complete the call, i.e., if an incoming call requests four servers and there are only two servers available in the system, the dispatcher will dispatch the two servers and the remaining two servers will be dispatched by a source external to the system.

Before we derive the steady-state probabilities for this model, we need to define the state-space. The steady-state probabilities in a $M/M/s$ model only depend on the number of busy servers in the system. Similarly, the steady-state probabilities for a system with multi-server dispatches depend only on the number of calls that are being served by one, two, ... servers. Therefore, state m is a vector $B_m = [n_1, n_2, \cdots, n_s]$ where n_i is a non-negative integer that indicates the number of calls in the system that are being served by i servers. The state-space $S(B)$ can be defined as

$$S(B) = \{[n_1, n_2, \cdots, n_s] : 0 \leq n_i \leq \lfloor s/i \rfloor, \sum_{i=1}^{s} i n_i \leq s\}. \tag{3.1}$$

We also define $w_i(B_m) = n_i$, $w(B_m) = \sum_{i=1}^{s} n_i = \sum_{i=1}^{s} w_i(B_m)$ and $V(B_m) = \sum_{i=1}^{s} i n_i$ where $w(B_m)$ is the total number of calls being served and $V(B_m)$ is the total number of busy servers in state B_m. We represent the state transitions by a vector whose elements are all zero expect the i^{th} element,

$$C_i = \{B : w_i(B) = 1, w_j(B) = 0 \text{ for } j \neq i, j = 1, \cdots, s\}. \tag{3.2}$$

Now, if a call arrives which requires d servers while the system is in state B_m, the system will move to state $B_m + C_d$ if there are at least d servers available in the system. Likewise, if the servers finish serving a call that required d servers, the system will move to state $B_m - C_d$.

First, we derive the balance equations for the states for which $V(B_m) < s$, i.e., the system is non-exhausted,

$$\begin{aligned}(\lambda + \mu w(B_m))Pr\{B_m\} = &\sum_{d=1:w_d(B_m)>0}^{s} \lambda Pr\{R_d\}Pr\{B_m - C_d\} \\ &+ \sum_{d=1:w(B_m)+d\leq s}^{s} \mu(w_d(B_m)+1)Pr\{B_m + C_d\}, \qquad (V(B_m) < s).\end{aligned} \tag{3.3}$$

The left-hand side in equation above corresponds to the transitions out of state B_m while the right-hand side represents the transitions into state B_m. The first sum on the right hand side accounts for the arrivals that lead to state B_m and the second term corresponds to the service completions that leave the system in state B_m. We also introduce the balance equations for an exhausted system as

$$\begin{aligned}\mu w(B_m)Pr\{B_m\} &= \sum_{d=1:w_d(B_m)>0}^{s} \sum_{d'=i}^{d_{\max}} \lambda Pr\{R_{d'}\}Pr\{B_m - C_d\} \\ &= \sum_{d=1:w_d(B_m)>0}^{s} \lambda Pr\{B_m - C_d\}[1 - \sum_{d'=1}^{d-1} Pr\{R_{d'}\}], \qquad (V(B_m) = s).\end{aligned} \tag{3.4}$$

The left-hand side of the equation above correspond to the departures from state B_m through service completions, and the right-hand side accounts for arrivals that lead to state B_m, including the ones that will be responded to by partial dispatches due to insufficient number of available servers.

Now, we can iteratively compute the probabilities $Pr\{B_m\}$ for all the states and use them to compute the steady-state probabilities as

$$P_n = \sum_{m:V(B_m)=n} Pr\{B_m\}. \tag{3.5}$$

These steady-state probabilities will be later used in section 3.5 to compute the stochastic components of the queueing model with an iterative procedure.

3.3 Server Utilizations and Dispatch Probabilities

Our main objective in this section is to compute the server utilizations r_i, the proportion of times when server i is busy, and the dispatch probabilities f_{ijm}, the probabilities that station i dispatches m servers to a call generated by customer j. We start by conditioning on the event that the stations that customer j prefers more than i have already dispatched l servers and that d servers have been requested by the current call. For now, we assume that $m > 0$, meaning that the more preferred stations than i do not have enough available servers to dispatch the requested number of servers.

$$\begin{aligned} h_{ijm} = Pr\{A_{ijm}\} &= \sum_{d=m}^{d_{\max}} \sum_{l=0}^{\min(z_{k-1},d-m)} Pr\{A_{ijm}|C_{ijl}R_d\}Pr\{C_{ijl}|R_d\}Pr\{R_d\} \\ &= \sum_{d=m}^{d_{\max}} \sum_{l=0}^{\min(z_{k-1},d-m)} Pr\{A_{ijm}|C_{ijl}R_d\}Pr\{C_{ijl}\}Pr\{R_d\}. \end{aligned} \tag{3.6}$$

The last step above holds since we assumed that $m > 0$; hence, the more preferred stations have already dispatched all of their available servers, the probability for that does not depend on d. The probability $Pr\{A_{ijm}|C_{ijl}R_d\}$ can be computed for two separate cases when the current station has enough servers to complete the request by sending the remaining number of requested servers ($l + m = d$) and when it does not ($l + m < d$). In the latter case, all servers at i will be examined to find the available servers. Therefore, this is a binomial probability with probability of success (busy server) equal to r_i,

$$Pr\{A_{ijm}|C_{ijl} \cap R_d \cap (l+m<d)\} = \binom{s_i}{m} r_i^{s_i-m}(1-r_i)^m. \tag{3.7}$$

We condition on the event that there are n busy servers at station i, Z_n^i, to compute the probability in the case where $l+m=d$,

$$\begin{aligned} &Pr\{A_{ijm}|C_{ijl} \cap R_d \cap (l+m=d)\} \\ &= \sum_{n=0}^{s_i-m} Pr\{A_{ijm}|C_{ij(d-m)} \cap R_d \cap Z_n^i\} Pr\{Z_n^i\}. \end{aligned} \tag{3.8}$$

Notice that the first term in the product is equal to 1 since we have conditioned on $l+m=d$, i.e., the more preferred stations than i have dispatched $d-m$ servers, and there are at least m servers available at i. Therefore, the equation above can be simplified as

$$Pr\{A_{ijm}|C_{ijl} \cap R_d \cap (l+m=d)\} = \sum_{n=0}^{s_i-m} Pr\{Z_n^i\} = \sum_{n=0}^{s_i-m} \binom{s_i}{n} r_i^n (1-r_i)^{s_i-n}. \tag{3.9}$$

The second term in the products in (3.6) are the probability that the more preferred stations dispatch l servers in response to a call from j. This probability, which we will denote by ξ_{ijl}, can be expressed as

$$\xi_{ijl} = Pr\{C_{ijl}\} = \sum_{\phi \in \Phi_l^{ij}} \prod_{u=1}^{k} \binom{s_i}{\phi(u)} r_i^{s_i-\phi(u)}(1-r_i)^{\phi(u)}, \tag{3.10}$$

where Φ_l^{ij} is the set of all possible vectors of k non-negative integer numbers whose sum is equal to l such that the u^{th} number is less than or equal to $s_{(u)j}$ and $\phi(u)$ is the u^{th} number in vector ϕ corresponding to the u^{th} priority station. In other words, (3.10) sums over all the possible combinations of l available servers and $z_{k-1} - l$ busy server at the first $k-1$ stations, and computes the probability of each combination as the product of the binomial probabilities corresponding to each station. This is a special case of the Poisson binomial distribution, an extension of the binomial distribution for which the Bernouli trials are not identical, i.e., the probability of success is not necessarily the same for all the trials. Therefore, $Pr\{C_{ijl}\}$ can be computed using the pmf of this distribution where l is the number of successes and $(1-r_i)$'s are the probabilities of success. Fernandez and Williams (2010) and Hong (2011) proposed a closed-form expression for the pmf of the Poisson binomial distribution using the Inverse Fourier transform of the characteristic function of the distribution. Using this closed-form expression, we can express (3.10) as

$$\xi_{ijl} = \frac{1}{z_{k-1}+1} \sum_{u=0}^{z_{k-1}} K^{-ul} \prod_{v=1}^{k-1} (1 + (K^u - 1)(1 - r_{(v)j}))^{s_{(v)j}}, \tag{3.11}$$

where $K = \exp(\frac{2z\pi}{z_{k-1}+1})$ and z is the imaginary unit. Moreover, we can use the CDF of the Poisson binomial distribution, denoted by $\Xi_{ij}(q)$, to compute the probability that the more preferred stations than i dispatch no more than q servers to a call from j,

$$\Xi_{ij}(q) = \sum_{l=0}^{q} \xi_{ijl} = \frac{1}{z_{k-1}+1} \sum_{u=0}^{z_{k-1}} \frac{1 - K^{u(q+1)}}{1 - K^u} \prod_{v=1}^{k-1} (1 + (K^u - 1)(1 - r_{(v)j}))^{s_{(v)j}}. \tag{3.12}$$

By substituting (3.7), (3.9) and (3.11) in (3.6), we can obtain the equation for h_{ijm} as

$$\begin{aligned} h_{ijm} &= \sum_{d=m}^{d_{\max}} \left(\sum_{l=0}^{\min(z_{k-1},d-m-1)} \binom{s_i}{m} r_i^{s_i-m}(1-r_i)^m \xi_{ijl} \right) Pr\{R_d\} \\ &+ \sum_{d=m}^{z_{k-1}+m} \left(\sum_{n=0}^{s_i-m} \binom{s_i}{n} r_i^n (1-r_i)^{s_i-n} \xi_{ij(d-m)} \right) Pr\{R_d\} \\ &= \binom{s_i}{m} r_i^{s_i-m}(1-r_i)^m \sum_{t=0}^{d_{\max}-m} \left[\sum_{l=0}^{\min(z_{k-1},t-1)} \xi_{ijl} \right] Pr\{R_{t+m}\} \\ &+ \left(\sum_{n=0}^{s_i-m} \binom{s_i}{n} r_i^n (1-r_i)^{s_i-n} \right) \left(\sum_{t=0}^{z_{k-1}} \xi_{ijt} Pr\{R_{t+m}\} \right). \end{aligned} \tag{3.13}$$

Notice that $d-m$ is substituted with t and the bounds for the sums are adjusted accordingly.

The sum inside the brackets is the CDF of the Poisson binomial distribution, mentioned in (3.12). Hence, we can rewrite (3.13) as

$$\begin{aligned} h_{ijm} =& \binom{s_i}{m} r_i^{s_i-m}(1-r_i)^m \sum_{t=0}^{d_{\max}-m} \left[\Xi_{ij}(\min(z_{k-1},t-1)) Pr\{R_{t+m}\} \right] \\ &+ \left(\sum_{n=0}^{s_i-m} \binom{s_i}{n} r_i^n (1-r_i)^{s_i-n} \right) \left(\sum_{t=0}^{z_{k-1}} \xi_{ijt} Pr\{R_{t+m}\} \right), \qquad (m > 0). \end{aligned} \tag{3.14}$$

Also, we can find h_{ijm} when $m = 0$ as

$$h_{ij0} = 1 - \sum_{m=1}^{s_i} h_{ijm}. \tag{3.15}$$

We can rewrite (3.14) as

$$h_{ijm} = (1-r_i)^{s_i} w_{ijm}, \tag{3.16}$$

where w_{ijm} is defined as

$$\begin{aligned} w_{ijm} =& \binom{s_i}{m} r_i^{s_i-m}(1-r_i)^{-s_i+m} \sum_{t=0}^{d_{\max}-m} \left[\Xi_{ij}(\min(z_{k-1},t-1)) Pr\{R_{t+m}\} \right] \\ &+ \left(\sum_{n=0}^{s_i-m} \binom{s_i}{n} r_i^n (1-r_i)^{-n} \right) \left(\sum_{t=0}^{z_{k-1}} \xi_{ijt} Pr\{R_{t+m}\} \right), \qquad (m > 0), \end{aligned} \tag{3.17}$$

and

$$\begin{aligned} w_{ij0} &= \frac{h_{ij0}}{(1-r_i)^{s_i}} = \frac{1-\sum_{m=1}^{s_i} h_{ijm}}{(1-r_i)^{s_i}} \\ &= \frac{1-\sum_{m=1}^{s_i}(1-r_i)^{s_i} w_{ijm}}{(1-r_i)^{s_i}} = \frac{1}{(1-r_i)^{s_i}} - \sum_{m=1}^{s_i} w_{ijm}. \end{aligned} \tag{3.18}$$

An approximation for h_{ijm} can be obtained by using the system-wide mean server utilization, r, instead of the individual server utilizations. The approximation, h^c_{ijm} can be derived by replacing the Poisson binomial probabilities with binomial probabilities, as shown below.

$$\begin{aligned} h^c_{ijm} =& \binom{s_i}{m} r^{s_i-m}(1-r)^m \sum_{t=0}^{d_{\max}-m} \left[\sum_{l=0}^{\min(z_{k-1},t-1)} \binom{z_{k-1}}{l} r^{z_{k-1}-l}(1-r)^l Pr\{R_{t+m}\} \right] \\ &+ \left(\sum_{n=0}^{s_i-m} \binom{s_i}{n} r^n (1-r)^{s_i-n} \right) \left(\sum_{t=0}^{z_{k-1}} \binom{z_{k-1}}{t} r^{z_{k-1}-t}(1-r)^t Pr\{R_{t+m}\} \right), \qquad (m>0). \end{aligned} \tag{3.19}$$

Likewise, the value for h^c_{ijm} when $m = 0$ can be computed using (3.15) by summing over h^c_{ijm}'s instead of h_{ijm}'s. The probabilities in (3.14) are computed based on the assumption that the status of each server is statistically independent from other servers. The probabilities h^c_{ijm} will be used in the next section to find the correction factors. These correction factors are multiplied with h_{ijm}'s to reduce the effect of the server independence assumption. Hence, the dispatch probabilities, f_{ijm}, can be approximated as

$$f_{ijm} = Q_{ijm} h_{ijm} = Q_{ijm}(1-r_i)^{s_i} w_{ijm}, \tag{3.20}$$

where Q_{ijm} is the correction factor for customer j and m servers being dispatched from station i.

The utilization for the servers that are located at station i can be calculated as,

$$\begin{aligned} r_i =&\frac{1}{s_i}\sum_{j\in J}\sum_{m=0}^{s_i}\lambda_j f_{ijm} m\tau_{ij} = \frac{1}{s_i}\sum_{j\in J}\sum_{m=0}^{s_i}\lambda_j Q_{ijm}(1-r_i)^{s_i} w_{ijm} m\tau_{ij} \\ =&\frac{1}{s_i}(1-r_i)^{s_i} V_i, \end{aligned} \tag{3.21}$$

where

$$V_i = \sum_{j\in J}\sum_{m=0}^{s_i}\lambda_j Q_{ijm} w_{ijm} m\tau_{ij}. \tag{3.22}$$

We can rewrite the equation for r_i as follows.

$$\begin{aligned} &r_i = \frac{1}{s_i}(1-r_i)^{s_i}V_i \Rightarrow \frac{s_i r_i^{s_i}}{r_i^{s_i-1}} = (1-r_i)^{s_i}V_i \Rightarrow \left(\frac{1-r_i}{r_i}\right)^{s_i} = \frac{s_i}{r_i^{s_i-1}V_i} \Rightarrow \\ &\frac{1-r_i}{r_i} = \left(\frac{s_i}{r_i^{s_i-1}V_i}\right)^{1/s_i} \Rightarrow r_i = \frac{1}{\left(\frac{s_i}{r_i^{s_i-1}V_i}\right)^{1/s_i}+1} = \frac{(r_i^{s_i-1}V_i)^{1/s_i}}{s_i^{1/s_i}+(r_i^{s_i-1}V_i)^{1/s_i}}. \end{aligned} \tag{3.23}$$

We will use this equation for r_i later in our iterative procedure. Next, we will derive the correction factors.

3.4 Correction Factors

In this section, we derive the correction factors by assuming that the system works according to a $M[G]/M/s/s$ model, an extension of the $M/M/s/s$ model with multi-server dispatches where G denotes a general pmf for the number of the requested servers per incoming call call. The model does not incur a line and the calls that arrive when all the servers are busy or the calls that are partially covered will receive service from the backup vehicles or from the neighboring counties.

The first step in deriving the correction factors is to compute the probabilities p_{ijm}. We do so by conditioning on E_{ijl}, Z_n and R_d.

$$
\begin{aligned}
p_{ijm} = Pr\{D_{ijm}\} &= \sum_{n=0}^{s} \sum_{d=m+L_1}^{d_{\max}} \sum_{l=L_2}^{U_2} Pr\{D_{ijm}|E_{ijl}R_dZ_n\}Pr\{E_{ijl}R_dZ_n\} \\
&= \sum_{n=0}^{s} \sum_{d=m+L_1}^{d_{\max}} \sum_{l=L_2}^{U_2} Pr\{D_{ijm}|E_{ijl}R_dZ_n\}Pr\{E_{ijl}|R_dZ_n\}Pr\{R_d\}Pr\{Z_n\},
\end{aligned}
\tag{3.24}
$$

where $L_1 = \max(z_{k-1} - n, 0)$, $L_2 = \max(z_{k-1} - n, z_k - m - n, 0)$ and $U_2 = \min(d - m, z_{k-1}, s - n - m)$. We assume that the number of requested servers does not depend on the status of the system and therefore R_d and Z_n are independent. Probabilities $Pr\{R_d\}$ are known from the input and probabilities $Pr\{Z_n\}$ are the steady-state probabilities that we have already derived in section 3.2. Next, we need to find the expressions for $Pr\{D_{ijm}|E_{ijl}R_dZ_n\}$ and $Pr\{E_{ijl}|R_dZ_n\}$. Here, it is assumed that $m > 0$, i.e., the current station will dispatch at least one server. There are two possibilities in computing $Pr\{D_{ijm}|E_{ijl}R_dZ_n\}$, the current station either has enough available servers to complete the request ($l+m = d$) or it does not ($l+m < d$). We first consider the later case. The probability that the first sampled server at the k^{th} preferred station is busy given that there are n busy servers in the system and that the more preferred stations have already dispatched l server in response to the current call is $\frac{n-(z_{k-1}-l)}{s-z_{k-1}}$. Likewise, the probability that the second sampled server is busy is $\frac{n-(z_{k-1}-l)-1}{s-z_{k-1}-1}$, and so on. Therefore, we can write,

$$
\begin{aligned}
&Pr\{D_{ijm}|E_{ijl} \cap R_d \cap Z_n \cap (l+m<d)\} \\
&= \left[\left(1 - \frac{n-(z_{k-1}-l)}{s-z_{k-1}}\right) \times \left(1 - \frac{n-(z_{k-1}-l)}{s-z_{k-1}-1}\right) \times \cdots \times \left(1 - \frac{n-(z_{k-1}-l)}{s-z_{k-1}-(m-1)}\right) \right. \\
&\left. \times \frac{n-(z_{k-1}-l)}{s-z_{k-1}-m} \times \frac{n-(z_{k-1}-l)-1}{s-z_{k-1}-(m+1)} \times \cdots \times \frac{n-(z_{k-1}-l)-(s_i-m-1)}{s-z_{k-1}-(s_i-1)} \right] + \cdots \\
&+ \left[\frac{n-(z_{k-1}-l)}{s-z_{k-1}} \times \frac{n-(z_{k-1}-l)-1}{s-z_{k-1}-1} \times \cdots \times \frac{n-(z_{k-1}-l)-(s_i-m-1)}{s-z_{k-1}-(s_i-m-1)} \right. \\
&\times \left(1 - \frac{n-(z_{k-1}-l)-(s_i-m)}{s-z_{k-1}-(s_i-m)}\right) \times \left(1 - \frac{n-(z_{k-1}-l)-(s_i-m)}{s-z_{k-1}-(s_i-m+1)}\right) \times \cdots \\
&\left. \times \left(1 - \frac{n-(z_{k-1}-l)-(s_i-m)}{s-z_{k-1}-(s_i-1)}\right) \right],
\end{aligned}
\tag{3.25}
$$

where the sum involves all the possible orders of sampling m available servers and $s_i - m$ busy servers at station i. There are $\binom{s_i}{m}$ terms in the sum and it can be simplified as

$$
\begin{aligned}
&Pr\{D_{ijm}|E_{ijl} \cap R_d \cap Z_n \cap (l+m<t)\} \\
&= \binom{s_i}{m} \frac{\prod_{u=0}^{s_i-m-1}(n-(z_{k-1}-l)-u) \prod_{u=0}^{m-1}(s-n-l-u)}{\prod_{u=0}^{s_i-1}(s-z_{k-1}-u)}.
\end{aligned}
\tag{3.26}
$$

Next, we find the same probabilities given that $l+m=d$.

$$
\begin{aligned}
&Pr\{D_{ijm}|E_{ijl}\cap R_d \cap Z_n \cap (l+m=d)\}\\
&=\left[\left(1-\frac{n-(z_{k-1}-l)}{s-z_{k-1}}\right)\times\left(1-\frac{n-(z_{k-1}-l)}{s-z_{k-1}-1}\right)\times\cdots\times\left(1-\frac{n-(z_{k-1}-l)}{s-z_{k-1}-(m-1)}\right)\right]+\cdots\\
&+\left[\frac{n-(z_{k-1}-l)}{s-z_{k-1}}\times\frac{n-(z_{k-1}-l)-1}{s-z_{k-1}-1}\times\cdots\times\frac{n-(z_{k-1}-l)-U_3+1}{s-z_{k-1}-U_3+1}\right.\\
&\times\left(1-\frac{n-(z_{k-1}-l)-U_3}{s-z_{k-1}-U_3}\right)\times\left(1-\frac{n-(z_{k-1}-l)-U_3}{s-z_{k-1}-(U_3+1)}\right)\times\cdots\\
&\left.\times\left(1-\frac{n-(z_{k-1}-l)-U_3}{s-z_{k-1}-(U_3+m-1)}\right)\right].
\end{aligned}
\tag{3.27}
$$

where $U_3=\min(s_i-m, n-z_{k-1}+r-m)$. We use v to denote the number of busy servers that are sampled before we find the m^{th} available server at the k^{th} preferred station. Now we can simplify (3.27) as

$$
\begin{aligned}
&Pr\{D_{ijm}|E_{ijl}\cap R_d \cap Z_n \cap (l+m=t)\}\\
&=\sum_{v=0}^{U_3}\binom{v+m-1}{v}\frac{\prod_{u=0}^{v-1}(n-(z_{k-1}-l)-u)\prod_{u=0}^{m-1}(s-n-l-u)}{\prod_{u=0}^{v+m-1}(s-z_{k-1}-u)}.
\end{aligned}
\tag{3.28}
$$

Next, we need to compute the probability $Pr\{E_{ijl}|R_dZ_n\}$ in a similar manner. Since we are assuming that $m>0$, the more preferred stations should have dispatched all of their available servers. Hence, the probability that l servers are dispatched by the more preferred stations is only dependent on the number of busy (free) servers at those stations and not d.

As a result, we can write

$$\begin{aligned}
&Pr\{E_{ijl}|R_dZ_n\} = Pr\{E_{ijl}|Z_n\}\\
&= \left[\left(1-\frac{n}{s}\right)\times\left(1-\frac{n}{s-1}\right)\times\cdots\times\left(1-\frac{n}{s-(l-1)}\right)\times\frac{n}{s-l}\times\frac{n-1}{s-(l+1)}\times\cdots\right.\\
&\left.\times\frac{n-(z_{k-1}-l-1)}{s-(z_{k-1}-1)}\right]+\cdots+\left[\frac{n}{s}\times\frac{n-1}{s-1}\times\cdots\times\frac{n-(z_{k-1}-l-1)}{s-z_{k-1}-l-1}\right.\\
&\left.\times\left(1-\frac{n-(z_{k-1}-l)}{s-(z_{k-1}-l)}\right)\times\left(1-\frac{n-(z_{k-1}-l)}{s-(z_{k-1}-l+1)}\right)\times\cdots\times\left(1-\frac{n-(z_{k-1}-l)}{s-(z_{k-1}-1)}\right)\right]\\
&= \binom{z_{k-1}}{l}\frac{\prod_{u=0}^{z_{k-1}-l-1}(n-u)\prod_{u=0}^{l-1}(s-n-u)}{\prod_{u=0}^{z_{k-1}-1}(s-u)}.
\end{aligned}\tag{3.29}$$

Now we can rewrite (3.24) as

$$\begin{aligned}
p_{ijm} = Pr\{D_{ijm}\} =&\sum_{n=0}^{s}\sum_{d=m+L_1}^{d_{\max}}\left[\left(\sum_{l=L_2}^{U_4}Pr\{D_{ijm}|E_{ijl}R_dZ_n\}Pr\{E_{ijl}|Z_n\}\right)\right.\\
&\left.+Pr\{D_{ijm}|C_{ij(d-m)}R_dZ_n\}Pr\{C_{ij(d-m)}|Z_n\}\right]Pr\{R_d\}Pr\{Z_n\},
\end{aligned}\tag{3.30}$$

where $U_4 = \min(d-m-1, z_{k-1}, s-n-m)$. We can obtain the first term by multiplying (3.26) and (3.29). Given that $(l+m<d)$,

$$\begin{aligned}
&Pr\{D_{ijm}|E_{ijl}R_dZ_n\}Pr\{E_{ijl}|Z_n\}\\
&= \binom{s_i}{m}\frac{\prod_{u=0}^{s_i-m-1}(n-(z_{k-1}-l)-u)\prod_{u=0}^{m-1}(s-n-l-u)}{\prod_{u=0}^{s_i-1}(s-z_{k-1}-u)}\\
&\times\binom{z_{k-1}}{l}\frac{\prod_{u=0}^{z_{k-1}-l-1}(n-u)\prod_{u=0}^{l-1}(s-n-u)}{\prod_{u=0}^{z_{k-1}-1}(s-u)}\\
&= \binom{s_i}{m}\binom{z_{k-1}}{l}\frac{\prod_{u=0}^{z_{k-1}-l-m-1}(n-u)\prod_{u=0}^{l+m-1}(s-n-u)}{\prod_{u=0}^{s_i-1}(s-u)}
\end{aligned}\tag{3.31}$$

Likewise, we use (3.28) and (3.29) to obtain the second term in (3.30).

$$\begin{aligned}
&Pr\{D_{ijm}|C_{ij(d-m)}R_dZ_n\}Pr\{C_{ij(d-m)}|Z_n\}\\
&=\sum_{v=0}^{U_3}\binom{v+m-1}{v}\frac{\prod_{u=0}^{v-1}(n-(z_{k-1}-(d-m))-u)\prod_{u=0}^{m-1}(s-n-(d-m)-u)}{\prod_{u=0}^{v+m-1}(s-z_{k-1}-u)}\\
&\times\binom{z_{k-1}}{d-m}\frac{\prod_{u=0}^{z_{k-1}-(d-m)-1}(n-u)\prod_{u=0}^{(d-m)-1}(s-n-u)}{\prod_{u=0}^{z_{k-1}-1}(s-u)}\\
&=\binom{z_{k-1}}{d-m}\sum_{v=0}^{U_3}\binom{v+m-1}{v}\frac{\prod_{u=0}^{z_{k-1}-(d-m)+(v-1)}(n-u)\prod_{u=0}^{d-1}(s-n-u)}{\prod_{u=0}^{z_{k-1}+v+m-1}(s-u)}
\end{aligned}\tag{3.32}$$

The final equation for p_{ijm} can be obtained by substituting (3.31) and (3.32) in (3.30),

$$\begin{aligned}
p_{ijm}=\sum_{n=0}^{s}\sum_{d=L_1}^{d_{\max}}\Bigg[&\left(\sum_{l=L_2}^{U_4}\binom{s_i}{m}\binom{z_{k-1}}{l}\frac{\prod_{u=0}^{z_{k-1}-l-m-1}(n-u)\prod_{u=0}^{l+m-1}(s-n-u)}{\prod_{u=0}^{z_i-1}(s-u)}\right)\\
&+\binom{z_{k-1}}{d-m}\sum_{v=0}^{U_3}\binom{v+m-1}{v}\frac{\prod_{u=0}^{z_{k-1}-(d-m)+(v-1)}(n-u)\prod_{u=0}^{d-1}(s-n-u)}{\prod_{u=0}^{z_{k-1}+v+m-1}(s-u)}\Bigg]Pr\{R_d\}Pr\{Z_n\}.
\end{aligned}\tag{3.33}$$

The correction factors can be found by dividing p_{ijm} and h^c_{ijm},

$$Q_{ijm}=\frac{p_{ijm}}{h^c_{ijm}}.\tag{3.34}$$

These correction factors would reduce to the correction factors that Budge et al. (2009) proposed when $Pr\{R_1\}=1$ and $Pr\{R_d\}=0$ for $d\neq 1$.

3.5 Iterative Procedure

So far, we have derived the balance equations and the steady-state probabilities for an $M[G]/M/s/s$ queueing model, the equations for the dispatch probabilities, the server utilizations and the correction factors. We will now introduce an iterative procedure that involves these elements to compute different stochastic components of a dispatching model with multiple servers per station and multi-server dispatches.

The dispatching model that is proposed in this thesis can respond to a call by sending multiple servers. The dispatched servers can be from different stations. In this work, all the dispatched servers that are serving the same call will have identical service times, as opposed to Chelst and Barlach (1981) which assumes independent service times for the servers that are dispatched to serve the same call. As a result, service times should depend on the identity of all the stations that are involved in serving the current call. This will lead to an exponential number of service times which will not be easy to manage and also reduces the scalability of the model. However, it is known that service times are composed of response time, time at the scene, travel time to the hospital, and return time. Time at the scene and travel time to the hospital only depend on the type and location of the call and not on the stations from which the servers have been dispatched. Time at the scene is usually the longest time among the four. Moreover, the emergency systems always try to minimize the response and return times in order to maximize the coverage and minimize the server idle times. Thus, it is reasonable to assume that the service times only depend on the customer. This assumption greatly reduces the complexity of maintaining the individual service times and computing the system-wide mean service time. In particular, the mean service time only depends on the call arrival rates and the individual service times,

$$\tau = \frac{1}{\lambda} \sum_{j \in J} \lambda_j \tau_j. \tag{3.35}$$

Therefore, τ remains constant during the iterative procedure. Likewise, the steady-state probabilities are a function of $\mu = 1/\tau$, λ and the probabilities $Pr\{R_j\}$, none of which changes during the iterative procedure. Hence, the steady-state probabilities remain constant as well. The system wide mean server utilization can be computed as

$$r = \sum_{n=1}^{s} nP_n. \tag{3.36}$$

Therefore, r does not change since P_n's are constant during the procedure. As a result, τ, P_n's and r can be computed in advance and used within the iterative procedure.

The algorithm starts with the initialization step, then it enters the main computation loop, and terminates when the change in the server utilizations from the previous iteration becomes less than a pre-specified threshold, ε.

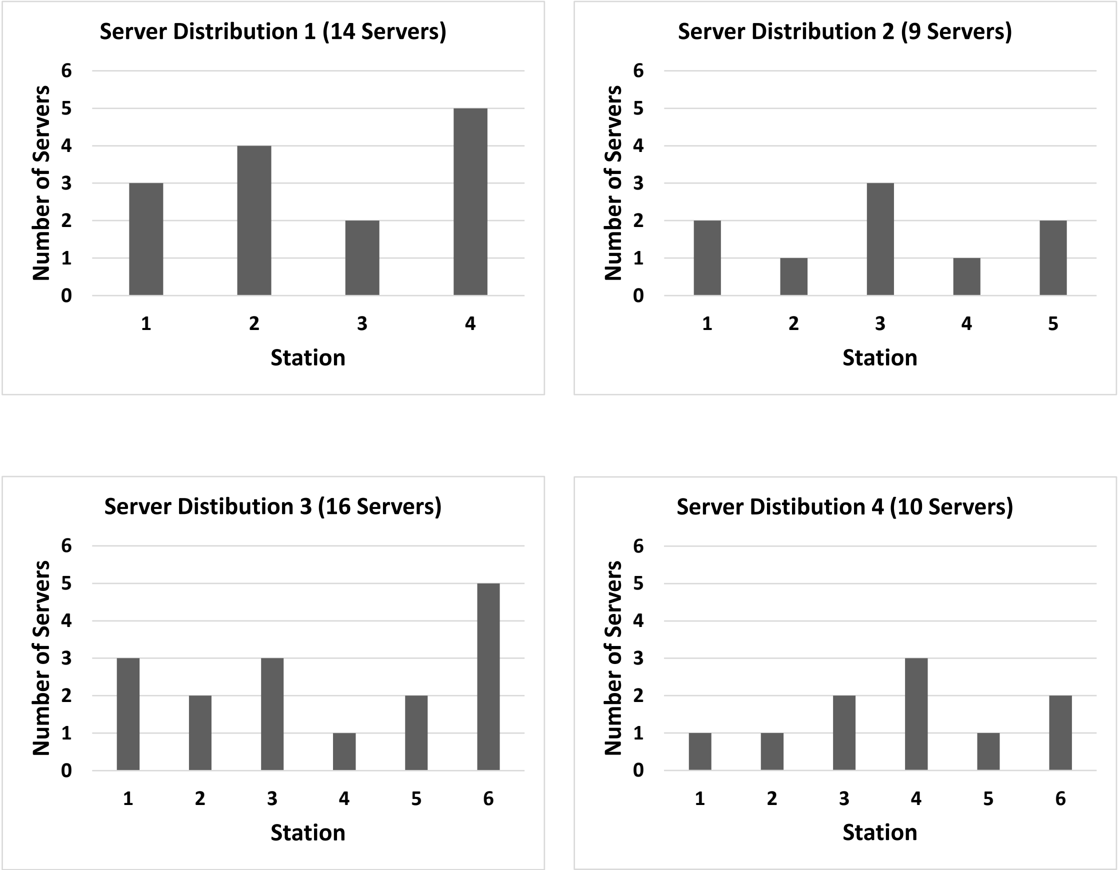
Step 0. Compute τ using (3.35), P_n's iteratively using (3.3),(3.4) and (3.5), and r using (3.36). Initialize the station-specific utilities as $r_i = r$. Set the iteration counter q to 1.

Step 1. Compute the correction factors Q_{ijm} using (3.19), (3.33) and (3.34).

Step 2. Use (3.11), (3.12), (3.17), (3.18) and the correction factors from step 1 to compute V_i^q by (3.22). Compute r_i^q using (3.23) having V_i^q and r_i^{q-1}. Normalize the utilizations using

$$r_i = r\frac{sr_i}{\sum_{i' \in I} s_{i'} r_{i'}}. \tag{3.37}$$

Step 3. If $\max|r_i^q - r_i^{q-1}| > \varepsilon$, set $q \leftarrow q + 1$ and go to step 1; otherwise, go to step 4.


Server Distribution 1 (14 Servers)

Server Distribution 2 (9 Servers)

Server Distibution 3 (16 Servers)

Server Distibution 4 (10 Servers)

Figure 3.1: The number of servers at each station for the four server distributions.

Step 4. Find the dispatch probabilities f_{ijm} using (3.20).

It is not guaranteed that the iterative procedure above always converges, but it did in every case that we ran.

3.6 Simulation and Results

We evaluate the proposed approximate Hypercube model that was introduced in the previous section. The results of the model are compared to those of a discrete-event simulation for

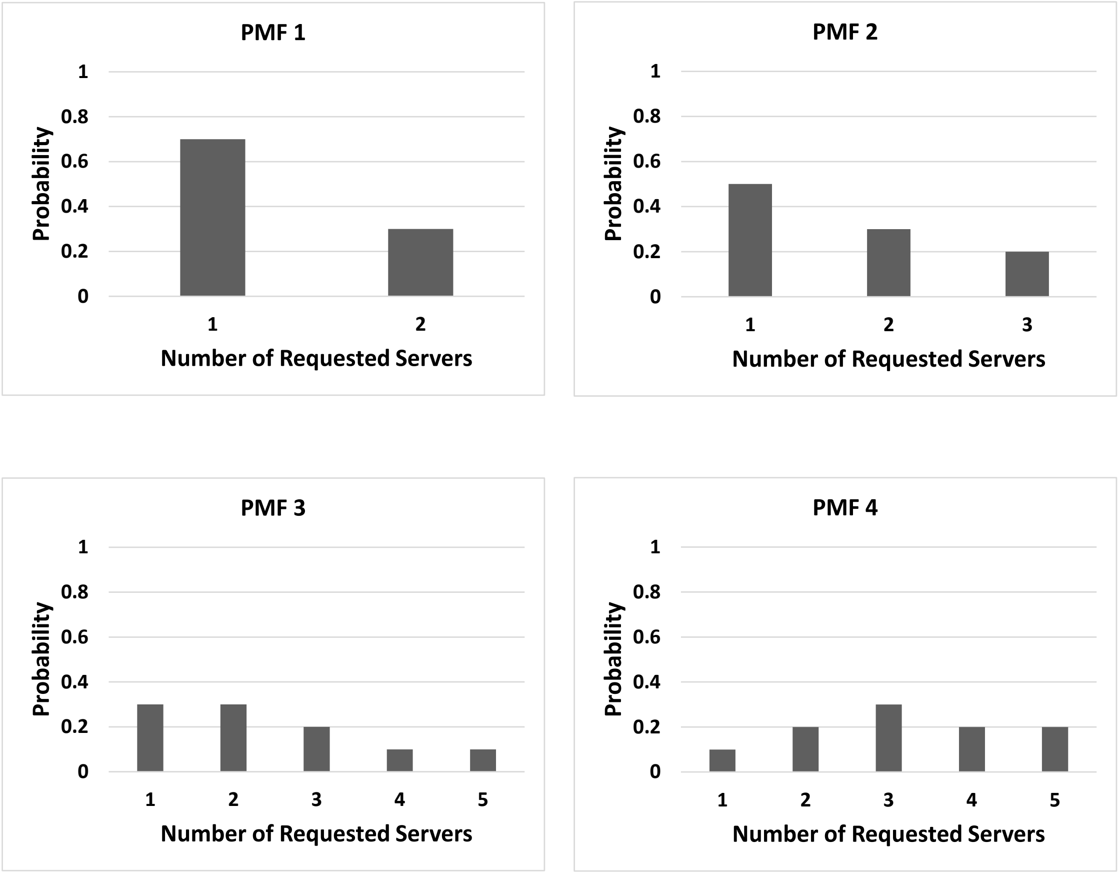


Figure 3.2: The PMF of the number of requested servers per call for the four test cases.

different number of servers, different server distributions and different pmf's for the number of requested server per each call. We use four different server distributions and four different pmf's that are depicted in Figures 3.1 and 3.2. The dispatching policies are randomly generated for each server distribution. The first three pmf's have decreasing probabilities as the number of requested servers increases, with different spreads. The fourth pmf has the highest probability at 4 requested servers, and then at 3 and 6 servers. Sixteen different scenarios have been generated for the combinations of server distributions and pmf's and are used to test the proposed Hypercube model. Each scenario has 100 customer locations where the call arrival rates and service times are generated randomly according to a uniform distribution such with $\rho = 0.4$ where

$$\rho = \overline{d}\frac{\sum_{j \in J} \lambda_j \tau_j}{s\lambda}, \tag{3.38}$$

and $\overline{d} = \sum_{t=1}^{d_{\max}} tPr\{R_t\}$. The simulation for each scenario was run 30 times and each run served 100000 calls. The results were then averaged across the runs. The relative error for each scenario is shown in Figure 3.3. All the relative errors are below 2% for the first two server distributions. The third distribution has the largest relative errors compared to the other distributions. In particular, the relative error for the combination of the third server distribution and pmf 1 is 6% which is the largest relative error among all the scenarios. The relative error for the fourth distribution is less than 2.5% for all the combinations. Except one of the stations in distribution 3, the rest of the combinations have relatively small errors; hence, the proposed Hypercube model is successful in approximating the queueing dynamics of most of the scenarios.

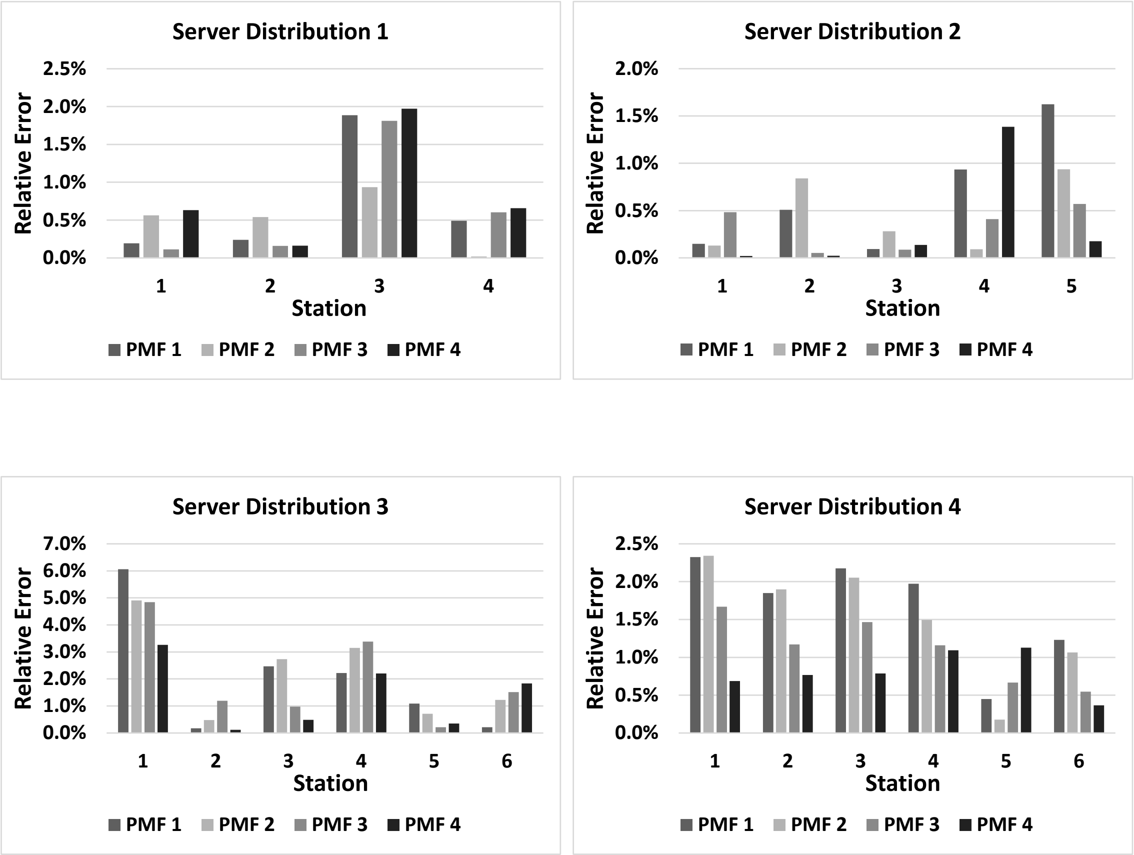


Figure 3.3: The relative error for server utilizations for sixteen scenarios.

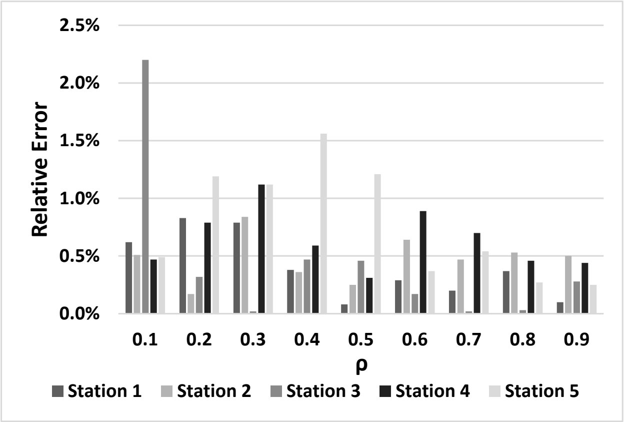


Figure 3.4: The results of sensitivity analysis on ρ for server distribution 2 and PMF 2.

Table 3.2: The absolute and relative errors comparing the server utilizations for the Approximate Hypercube (AH) model and the simulation results with different PMF's for Hanover county. The stations are 1:Ashland, 4:Doswell, 6:Henry, 7:Mechanicsville, 8:Montpelier, 10:Chickahominy, 11:Farrington, 14:East Hanover.

	Station	1	4	6	7	8	10	11	14	Average
	s_i	2	1	1	2	1	1	1	2	1.375
PMF 1 ($\rho = 0.13$)	AH r_i	0.216	0.118	0.186	0.205	0.166	0.170	0.136	0.154	0.169
	Sim r_i	0.221	0.114	0.180	0.208	0.166	0.166	0.134	0.154	0.168
	Abs. Err.	0.005	0.004	0.006	0.004	0.000	0.004	0.002	0.000	0.003
	Rel. Err. (%)	2.176	3.251	3.337	1.777	0.242	2.474	1.267	0.130	1.832
PMF 2 ($\rho = 0.17$)	AH r_i	0.274	0.170	0.239	0.254	0.186	0.234	0.189	0.213	0.220
	Sim r_i	0.280	0.166	0.231	0.259	0.181	0.230	0.184	0.217	0.218
	Abs. Err.	0.006	0.004	0.009	0.005	0.005	0.004	0.005	0.003	0.005
	Rel. Err. (%)	2.109	2.598	3.729	1.890	2.936	1.607	2.776	1.524	2.396
PMF 3 ($\rho = 0.24$)	AH r_i	0.348	0.253	0.327	0.328	0.240	0.324	0.277	0.299	0.299
	Sim r_i	0.351	0.247	0.327	0.330	0.226	0.326	0.272	0.304	0.298
	Abs. Error	0.003	0.005	0.000	0.002	0.014	0.002	0.005	0.005	0.005
	Rel. Error (%)	0.769	2.183	0.061	0.697	6.142	0.583	1.691	1.742	1.733
PMF 4 ($\rho = 0.32$)	AH r_i	0.420	0.334	0.400	0.398	0.301	0.402	0.357	0.381	0.374
	Sim r_i	0.419	0.330	0.409	0.398	0.281	0.410	0.355	0.386	0.373
	Abs. Error	0.001	0.004	0.009	0.001	0.020	0.008	0.002	0.005	0.006
	Rel. Error (%)	0.263	1.364	2.151	0.201	6.970	1.878	0.592	1.323	1.843

A sensitivity analysis on the system-wide offered load was also performed. Server distribution 2 and PMF 2 were used for this analysis and ρ was varied from 0.1 to 0.9. The corresponding relative errors are shown in Figure 3.4. The relative error is smaller than 2% for all the cases except for station 3 when $\rho = 0.1$. The relative errors decrease on average as ρ increases. This is because the relative error formula is more sensitive when the values for server utilizations are smaller.

Next, we used the data from Hanover county and simulated the ambulance dispatcher with multi-server dispatches. The county has 269 call locations and the call arrival rates and service times are recorded by county's EMS center. The service times are distributed according to a mixture of two lognormal distributions. One of the lognormal distributions corresponds to the calls where the patient is transferred to the hospital and the other one

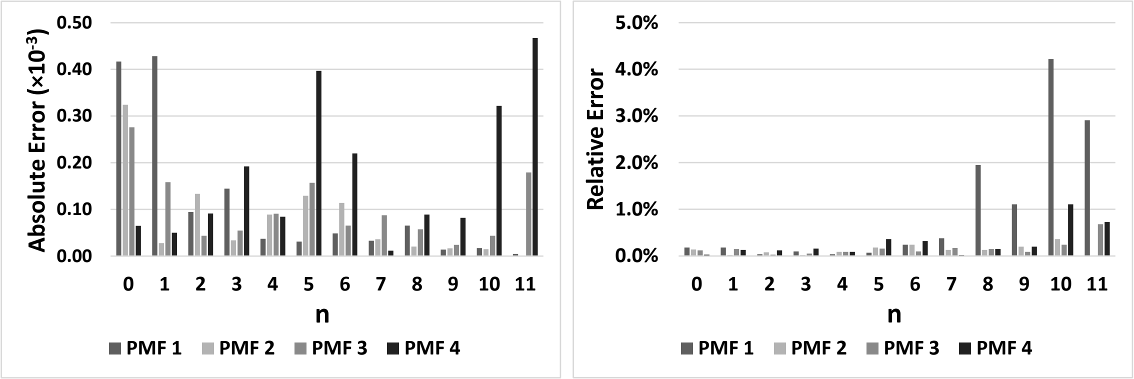


Figure 3.5: The absolute and relative errors for the steady-state probabilities (P_n) comparing the results of balance equations in Section 3.2 and the simulation model with different PMF's for Hanover county.

corresponds to the calls where the ambulance returns to its station after it treats the patient at the scene. The probability of transfer and the parameters of the lognormal distribution are computed based on the historical data. We ran four different simulations for the four PMF's shown in Figure 3.2, which resulted in ρ's of 0.13, 0.17, 0.24 and 0.32, computed using (3.38). Each simulation was run 30 times and 100000 calls were treated in each run. Eleven servers were located at eight station using the MEXCBL introduced in Chapter 2 and the optimal policy was used in the simulation. The number of servers located at each station along with the utilizations computed by the approximate Hypercube model and the simulation and the absolute and the relative errors are shown in Table 3.2. All the absolute errors are smaller than 0.02 and all the relative errors are smaller than 4% except two cases for Montpelier station when the third and fourth PMF's are used. Overall, the average absolute and relative errors are around 0.005 and 2%, respectively. As a result, the proposed Hypercube model approximates the server utilizations for Hanover country data effectively.

Moreover, the queueing factors seem to be insensitive to the distribution of the service times beyond their mean, which was previously reported in Gross and Harris (1998).

The balance equations that are derived in Section 3.2 are used in the iterative procedure to compute the steady-state probabilities. These probabilities are compared to the ones obtained from the simulation and the errors are depicted in Figure 3.5. All the absolute errors are smaller than 0.05%. Also, all the relative errors are smaller than 1% except for $n = 8$, 9, 10 and 11 for PMF 1. The large relative errors in these four cases are due to very small values for the steady-state probabilities ($P_8 = 0.0033$, $P_9 = 0.0012$, $P_{10} = 0.0004$ and $P_{11} = 0.0002$ from the simulation). These are the values that the absolute values are divided by, resulting in large relative errors.

All in the all, the results of the experiments with simulated data and the data from Hanover county indicate that the proposed approximate Hypercube has a relative error of less than 2% in most of the cases and thus it is sufficiently accurate for analyzing the stochastic aspects of the emergency systems.

3.7 Conclusions

In this chapter, we expanded the $M/M/s/s$ queueing model and introduced the $M[G]/M/s/s$ model to accommodate for policies with multi-server dispatches, where G denotes a general pmf for the number of requested servers per call. We derived the balance equations for this queueing model and used them to find the steady-state probabilities. Then, we derived the equations for the server utilizations, dispatch probabilities and the correction factors. An iterative procedure was presented then to find the server utilizations and dispatch probabilities. Finally, we evaluated the proposed Hypercube model by comparing its results to those that were obtained from a simulation model. The results indicate that the proposed model is sufficiently accurate.

The advantage of the proposed method is that it considers both co-located servers and multi-server dispatches explicitly. Moreover, it is neither limited to a particular dispatch policy nor it imposes any restriction on the number of servers per location or the number of dispatched servers.

The drawback of this model is that it is computationally more complex than the previous methods. Hence, a revised version of the model which is computationally more efficient would be beneficial. Furthermore, the relative error exceeds 5% in some cases. This is due to very small denominators in the relative error formula in some cases. In other cases, the proposed model needs to be modified to improve the accuracy.

Chapter 4

Summary and Future Work

4.1 Summary

The contributions of this thesis are three-fold. First, we introduced a MILP model to locate ambulances in an area and design response districts by forming a dispatch policy that creates priority lists for each call location. The proposed model balances the offered load, allows for multiple ambulances to be located at the same station, and considers high and low priority calls. The model allows for uncertainties in service times and ambulance availabilities by incorporating the Hypercube correction factors that model the stochastic aspects of the model. The model uses an iterative procedure that maintains a linear model although the correction factors are highly non-linear.

The results obtained from the proposed model closely matches the simulation results. Also, it effectively reduces the amount of load imbalance among the ambulances. It was shown that the effect of load balancing and enforcing contiguity for the first priority districts on the coverage level is minimal. Hence, the model achieves its goal of balancing the offered load and maintaining the contiguity without incurring a significant negative impact on the coverage.

Second, we expanded the popular $M/M/s/s$ queueing model and introduced the $M[G]/M/s/s$ model that allows multiple servers to serve a single call. The distribution of the number of servers that are requested for service per call is determined by the general pmf G. We found the steady-state probabilities for this model by iteratively solving the balance equations. The results in Section 3.6 indicate that this iterative method is very accurate.

The third contribution of this thesis is extending the approximate Hypercube model that was proposed in Budge et al. (2009) to accommodate for multi-server dispatches. Unlike the previous works in the literature, our model allows for arbitrary number of servers to be dispatched in response to a single call and it does not assume any particular dispatch policy, which makes it more general than the previous models. Moreover, this model can accommodate multiple servers per location as well. The proposed model is a step ahead in providing spatial queueing models that better represent the real-world emergency systems.

The results of the proposed approximate Hypercube model are compared to simulation results. The results indicate that the proposed model can successfully approximate the queueing dynamics of an emergency system. The relative error between the results of the Hypercube model and that of the simulation is less than 2% in most of the cases that we tested, which is comparable to the other Hypercube models that were introduced before in the literature.

4.2 Future Work

The MILP model that was presented in this work can be extended by incorporating lower limits on the coverage level for all the call locations. This will prevent some districts from being under-served and provides a more *fair* dispatch policy that guarantees a minimum level of coverage for all the customers. Moreover, multiple types of vehicles can be incorporated into the model to achieve a higher level of realism.

The proposed extension for the $M/M/s/s$ can be adapted for the infinite-length queues. The balance equations for the $M[G]/M/s/\infty$ model can be formed. The equations can then be solved to find the steady-state probabilities.

The approximate Hypercube model can be extended to handle dependence between the number of requested server and the location from which the call is originated. It can also incorporate more complex ways of modeling the service times for the systems with multi-server dispatching policies. Furthermore, the model will become more practical by assuming service times for the servers that are dispatched to serve the same call to be dependent but not identical.

The proposed approximate Hypercube model can be used to extend the MILP model in Chapter 2 in order to optimize the server locations and dispatching policies for systems that allow for multi-server dispatches.

Bibliography

Alsalloum, O. I. and Rand, G. K. (2006). Extensions to emergency vehicle location models. *Computers & Operations Research*, 33:2725 – 2743.

Andersson, T. and Värbrand, P. (2007). Decision support tools for ambulance dispatch and relocation. *Journal of the Operational Research Society*, 58:195 – 201.

Ball, M. O. and Lin, F. L. (1993). A reliability model applied to emergency service vehicle location. *Operations Research*, 41(1):18 – 36.

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

Berman, O., Drezner, Z., Tamir, A., and Wesolowsky, G. O. (2009). Optimal location with equitable loads. *Annals of Operations Research*, 167(1):307–325.

Berman, O., Krass, D., and Drezner, Z. (2003). The gradual covering decay location problem on a network. *European Journal of Operational Research*, 151:474 – 480.

Brandeau, M. L. and Larson, R. C. (1986). *Extending and applying the hypercube queueing model to deploy ambulances in Boston*. National Emergency Training Center.

Brotcorne, L., Laporte, G., and Semet, F. (2003). Ambulance location and relocation models. *European Journal of Operations Research*, 147:451 – 463.

Budge, S., Ingolfsson, A., and Erkut, E. (2009). Approximating vehicle dispatch probabilities for emergency service systems with location-specific service times and multiple units per location. *Operations Research*, 57:251–255.

Burwell, T. H., Jarvis, J. P., and McKnew, M. A. (1993). Modeling co-located servers and dispatch ties in the hypercube model. *Computers & Operations Research*, 20(2):113–119.

Carter, G. M., Chaiken, J. M., and Ignall, E. (1972). Response areas for two emergency units. *Operations Research*, 20(3):571 – 594.

Chelst, K. and Jarvis, J. P. (1979). Technical noteestimating the probability distribution of travel times for urban emergency service systems. *Operations Research*, 27(1):199–204.

Chelst, K. R. and Barlach, Z. (1981). Multiple unit dispatches in emergency services: Models to estimate system performance. *Management Science*, 27(12):1390 – 1409.

Chiyoshi, F. Y., Galvão, R. D., and Morabito, R. (2003). A note on solutions to the maximal expected covering location problem. *Computers & Operations Research*, 30(1):87–96.

Church, R. L. and Roberts, K. L. (1983). Generalized coverage models and public facility location. *Papers in Regional Science*, 53(1):117 – 135.

Daskin, M. and Haghani, A. (1984). Multiple vehicle routing and dispatching to an emergency scene. *Environment and Planning A*, 16(10):1349–1359.

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

Erkut, E., Ingolfsson, A., and Budge, S. (2008a). Maximum availability/reliability models for selecting ambulance station and vehicle locations: a critique. *Natural Sciences and Engineering Research Council of Canada*, pages 1–11.

Erkut, E., Ingolfsson, A., and Erdoğan, G. (2008b). Ambulance location for maximum survival. *Naval Research Logistics (NRL)*, 55(1):42–58.

Erkut, E., Ingolfsson, A., Sim, T., and Erdoğan, G. (2009). Computational comparison of five maximal covering models for locating ambulances. *Geographical Analysis*, 41(1):43–65.

Fernandez, M. and Williams, S. (2010). Closed-form expression for the poisson-binomial probability density function. *Aerospace and Electronic Systems, IEEE Transactions on*, 46(2):803–817.

Galvão, R. D., Chiyoshi, F. Y., and Morabito, R. (2005). Towards unified formulations and extensions of two classical probabilistic location models. *Computers & Operations Research*, 32(1):15–33.

Geroliminis, N., Karlaftis, M. G., and Skabardonis, A. (2009). A spatial queuing model for the emergency vehicle districting and location problem. *Transportation research part B: methodological*, 43(7):798–811.

Goldberg, J., Dietrich, R., Ming Chen, J., Mitwasi, M. G., Valenzuela, T., and Criss, E.

(1990). Validating and applying a model for locating emergency medical vehicles in tuczon, az. *European Journal of Operational Research*, 49(3):308–324.

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

Gross, D. and Harris, C. M. (1998). *Fundamentals of queuing theory*. Wiley New York.

Halpern, J. (1977). The accuracy of estimates for the performance criteria in certain emergency service queueing systems. *Transportation Science*, 11(3):223–242.

Hardy, G., Littlewood, J., and Pólya, G. (1952). *Inequalities*. Cambridge Mathematical Library. Cambridge University Press.

Hong, Y. (2011). On computing the distribution function for the sum of independent and nonidentical random indicators. *Technical Reports*, 11.

Iannoni, A. P. and Morabito, R. (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.

Iannoni, A. P., Morabito, R., and Saydam, C. (2011). Optimizing large-scale emergency medical system operations on highways using the hypercube queuing model. *Socio-Economic Planning Sciences*, 45(3):105–117.

Ignall, E., Carter, G., and Rider, K. (1982). An algorithm for the initial dispatch of fire companies. *Management Science*, 28(4):366 – 372.

Ingolfsson, A., Budge, S., and Erkut, E. (2008). Optimal ambulance location with random delays and travel times. *Health Care Management Science*, 11(3):262–274.

Jarvis, J. P. (1975). Optimization in stochastic systems with distinguishable servers. Technical report tr-19-75, MIT, Cambridge, MA.

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

Karasakal, O. and Karasakal, E. K. (2004). A maximal covering location model in the presence of partial coverage. *Computers & Operations Research*, 31:1515 – 1526.

Knight, V., Harper, P., and Smith, L. (2012). Ambulance allocation for maximal survival with heterogeneous outcome measures. *Omega*, 40(6):918–926.

Larson, R. (2004). Or models for homeland security.

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

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

Larson, R. C. and Mcknew, M. A. (1982). Police patrol-initiated activities within a systems queueing model. *Management Science*, 28(7):759–774.

Larson, R. C. and Rich, T. F. (1987). Travel-time analysis of new york city police patrol cars. *Interfaces*, 17(2):15–20.

Lee, S. (2011). The role of preparedness in ambulance dispatching. *Journal of the Operational Research Society*, 62(10):1888 – 1897.

McLay, L. A. and Mayorga, M. E. (2012a). A dispatching model for server-to-customer systems that balances efficiency and equity. *Manufacturing & Service Operations Management*, (to appear).

McLay, L. A. and Mayorga, M. E. (2012b). A model for optimally dispatching ambulances to emergency calls with classification errors in patient priorities. *IIE Transactions*, (to appear).

Mehrotra, A., Johnson, E. L., and Nemhauser, G. L. (1998). An optimization based heuristic for political districting. *Management Science*, 44(8):1100–1114.

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

Pell, J. P., Sirel, J. M., Marsden, A. K., Ford, I., and Cobbe, S. M. (2001). Effect of reducing ambulance response times on deaths from out of hospital cardiac arrest: cohort study. *BMJ: British Medical Journal*, 322(7299):1385.

Pirkul, H. and Schilling, D. A. (1988). The maximal covering location problems with capacities on total workload. *Management Science*, 34(7):896 – 908.

Pirkul, H. and Schilling, D. A. (1991). The maximal covering location problems with capacities on total workload. *Management Science*, 37(2):233 – 248.

ReVelle, C. and Hogan, K. (1989). The maximum availability location problem. *Transportation Science*, 23(3):192–200.

Sacks, S. R. and Grief, S. (1994). Orlando magic: Efficient design of police patrol districts. *OR/MS Today*, 21(1):30–32.

Saydam, C. and Aytuğ, H. (2003). Accurate estimation of expected coverage: revisited. *Socio-Economic Planning Sciences*, 37(1):69–80.

Swersey, A. J. (1982). A Markovian decision model for deciding how many fire companies to dispatch. *Management Science*, 28(4):352 – 365.

Swersey, A. J. (1994). *Operations Research and the Public Sector*, volume 6 of *Handbooks in Operations Research and Management Science*, chapter The Deployment of Police, Fire, and Emergency Medical Units, pages 151 – 200. ElseVier, North-Holland, Amsterdam.

Weintraub, A., Aboud, J., Fernandez, C., Laporte, G., and Ramirez, E. (1999). An emergency vehicle dispatching system for an electric utility in chile. *Journal of the Operational Research Society*, 50:690–696.

Appendix A

The NP-Completeness Proofs

A.1 Base Model

The Base Model is shown to be NP-complete in the strong sense through a polynomial transformation from the k-median problem. We consider the equivalent minimization form of the Base model objective function:

$$\min - \sum_{w \in W} \sum_{j \in J} \sum_{p=1}^{s} \sum_{m=1}^{\kappa_{wp}} h_{wjpm} z_{wjpm}$$

The decision version of k-median can be expressed as follows.

k-`Median Instance:` A set of n data points, S, their distances, D_{ij} where $i = 1, ..., n$ and $j = 1, ..., n$, an integer k and a threshold t.

`Question:` Is it possible to find a subset of S of size k, called cluster centers C, such that the sum of distances from each data point to the nearest cluster center is less than t?

The decision version of the Base model can be obtained by modifying the objective function and interpreting it as a threshold constraint where (2.20) is less than or equal to t. An instance of k-median can be transformed into an instance of the Base model by setting both

J and W to be equal to S, the set of input data points to k-median. Here, J represents the data points while W represents the potential cluster centers. Also, the parameter R_{wj} is equal to the distance between data point j and a potential cluster center w, i.e., $R_{wj} = -D_{wj}$, for $\forall w \in W$ and $j \in J$. The rest of the parameters for the Base model are set as $s = k$, $\lambda_j^H = 1$, $\lambda_j^L = 0$, $\tau_{wj} = 0$, $r = 0$ and $c_w = 1$, for $\forall j \in J$ and $\forall w \in W$. This transformation implicitly assumes that all ambulances are available all the time since service times are zero, and therefore, the districts other than the first priority districts are inconsequential.

The transformation requires $\mathcal{O}(n^2)$ time, and therefore, the transformation is polynomial. In order to prove the validity of the transformation, one needs to show that a *yes* instance of k-median would translate in a *yes* instance of the Base model and a *yes* instance of the Base model corresponds to a *yes* instance of k-median.

Assume a *yes* instance of k-median. We transform the k-median solution to a solution to the Base model by setting $y_w = 1$ if the point corresponding to w is a cluster center and $y_w = 0$ otherwise, and $z_{wj11} = 1$ if point j belongs to cluster w, and $z_{wj11} = 0$ otherwise. The remaining z_{wjp1} variables, where $p \neq 1$, are set arbitrarily such that z_{wjp1} forms a preference list satisfying (2.23)-(2.25). Note that the index m becomes singular since $c_w = 1$. Moreover, $\kappa_{wp} = 1$ and $\max(1, p' - c_w + 1) = p'$ for $\forall w \in W$, $p = 1, ..., s$ and $p' = 1, ..., s$. Therefore, $x_{wjp'} = z_{wjp'1}$ by (2.23), and in particular $x_{wj1} = z_{wj11}$. Also, (2.21) and (2.22) become redundant since they are satisfied whenever (2.25) holds. Furthermore, the number of cluster centers is k, therefore $\sum_{w \in W} y_w = k = s$. The correction factor, q_{jp1}, is equal to 1 when $p = 1$, and thus $h_{wjpm} = -R_{wj}$, which is equal to the distance between cluster center

w and data point j. When $p > 1$, $h_{wjpm} = 0$ since $r^{p-1} = 0^{p-1} = 0$. Therefore, the objective function reduces to $\sum_{w\in W}\sum_{j\in J} R_{wj}z_{wj11}$ which is the same as the objective function of k-median. Consequently, a solution to k-median with an objective value less than t leads to a solution to the Base model with an objective value less than t. Having satisfied all the constraints for the decision version of the Base model, a *yes* instance of k-median leads to a *yes* instance for the Base model.

A *yes* instance of the Base model consists of a preference list, z_{wjpm}, for which (2.21)-(2.27) are satisfied. It also has an objective value that is less than t. This solution to the Base model can be transformed into a solution to k-median by assigning a data point j to a cluster w if $z_{wj11} = 1$ for $\forall j \in J$. The first priority assignment of this preference list is a partition of the data points along with an assignment from partitions to the cluster centers. The number of cluster centers is $\sum_{w\in W} y_w = s = k$. Therefore, z_{wj11} is a solution to k-median with an objective value less than t, leading to a *yes* instance of k-median. Hence, the transformation from k-median to the Base model is valid and of polynomial time. Therefore, the Base model is NP-complete in the strong sense.

A.2 Model with Load Balancing and Contiguity (LBCM)

Here, we show that identifying districts that balance the workload is NP-complete in the strong sense using a transformation from the bin-packing problem. Bin-packing can be expressed as follows.

`Instance:` A set of n items, S, with their sizes $a_1, a_2, ..., a_n$, a bin size V and an integer

k where $0 < k \leq n$.

`Question:` Can we partition the items into k disjoint sets B_i $(1 \leq i \leq k)$ such that $\sum_{l \in B_i} a_l \leq V$ for all $1 \leq i \leq k$?

The decision version of LBCM is formed by removing the objective function and fixing the ambulance locations, leading to a *yes-no* model that checks the feasibility of a solution to LBCM. Therefore, we are interested in showing that finding a feasible way to form the preference lists is NP-complete.

The transformation from bin-packing to LBCM sets $J = S$ and $W = 1, ..., k$, the set of k bins. Moreover, the high-priority demands, λ_j^H, are set to be equal to the item sizes a_j, $\forall j \in J$. Furthermore, the load imbalance threshold is set equal to the bin size, $\delta = V$. The rest of the parameters are set as $s = k$, $\lambda_j^L = 0$, $\tau_{wj} = 1$, $r = 0$, $R_{wj} = 1$, $N_{wj} = \emptyset$, $c_w = 1$, and g_w are set arbitrarily for $\forall w \in W,\ j \in J$. The transformation requires $\mathcal{O}(nk)$ time, and it is therefore polynomial.

Now, we need to show that a *yes* instance for bin-packing translates into a *yes* instance for LBCM. Assume a solution to bin-packing and set $y_w = 1$ for $\forall w \in W$, and $z_{wj11} = 1$ if item $j \in B_w$ and $z_{wj11} = 0$ otherwise, $\forall w \in W,\ j \in J$. As in the transformation from k-median to the Base model, the remaining z_{wjp1} variables when $p \neq 1$ are set such that z_{wjp1} forms a preference list satisfying (2.23)-(2.25). Likewise, (2.21)-(2.26) are satisfied since $c_w = 1$ for $\forall w \in W$. Also, constraint (2.27) is satisfied since $\sum_{w \in W} y_w = |W| = k = s$, and (2.31) is relaxed since $N_{wj} = \emptyset$. Knowing that $q_{j11} = 1$ and $r^{p-1} = 0^{p-1} = 0$ when $p > 1$, constraint (2.28) simplifies to $o_w = \sum_{j \in J} \lambda_j^H z_{wj11} = \sum_{j \in J} a_j z_{wj11}$ for $\forall w \in W$. Hence, (2.29) and (2.30) can be expressed as $-V \leq \sum_{j \in J} a_j z_{wj11} \leq V$. Since we started with a solution to bin-packing, we have $\sum_{j \in J} a_j z_{wj11} = \sum_{l \in B_w} a_l \leq V$ for $\forall w \in W$. Also, $\sum_{j \in J} a_j z_{wj11} \geq 0$ since $a_j \geq 0$ for $\forall j \in J$. Consequently, $-V \leq 0 \leq \sum_{j \in J} a_j z_{wj11} \leq V$ and therefore, (2.28)-(2.30) are satisfied. All in all, a *yes* instance for bin-packing transforms into a *yes* instance for LBCM.

Lastly, we need to show that a *yes* instance for LBCM corresponds to a *yes* instance for bin-packing. A solution to LBCM can be transformed into a solution to bin-packing by assigning an item j to a bin B_w when $z_{wj11} = 1$. This transformation is valid since the first priority assignment of nodes in J to stations in W is a also a partition of nodes into s disjoint sets, corresponding to the k bins. A *yes* instance for LBCM leads to a *yes* instance for bin-packing since (2.28)-(2.30) imply that $-V \leq \sum_{j \in J} \lambda_j^H z_{wj11} = \sum_{j \in J} a_j z_{wj11} = \sum_{l \in B_w} a_l \leq V$, satisfying the capacity constraint in the bin-packing. Therefore, the transformation is valid, and LBCM is NP-complete in the strong sense.

Appendix B

Proofs of Theorems 1 and 2

B.1 Proof of Theorem 1

Consider demand node j. Without loss of generality, we assume that station 1 is the closest station to j, station 2 is the second closest station to j, and so on. As a result, $R_{1j} > R_{2j} > \cdots > R_{sj}$ since $R_{wj} = f(D_{wj})$, a monotonically decreasing function of distance D_{wj}. The effect of each demand node's preference list on the objective function is independent from the effects of preference lists for the other demand nodes since the terms in (2.20) do not interact with each other. Therefore, the contribution of node j's preference list to the objective is

$$\max \sum_{w \in W} \sum_{p=1}^{s} \sum_{m=1}^{\kappa_{wp}} q_{jpm}(1 - r^m) r^{p-1} \lambda_j^H R_{wj} z_{wjpm}, \qquad \forall j \in J \tag{B.1}$$

We can simplify the above as

$$\max \sum_{w \in W} \sum_{p=1}^{s} f_{wj} \lambda_j^H R_{wj} z_{wjp1}, \qquad \forall j \in J \tag{B.2}$$

where $f_{wj} = q_{jp1}(1 - r) r^{p-1}$ are the dispatch probabilities. Note that $m = 1$ since there is no more than one server located at every station. Therefore, the Base model objective function reduces to (B.2) for individual j's. Note that the correction factors (2.3) reduce to the Hypercube correction factors in Larson (1975) when there is no more than one server located at every station, and therefore, f_{wj}'s are monotonically decreasing with priority, i.e., $f_{wj} > f_{w'j}$ if j prefers w over w'. The objective function contribution from j in (B.2) is largest when $f_{1j} > f_{2j} > \cdots > f_{sj}$ since $R_{1j} > R_{2j} > \cdots > R_{sj}$ Hardy et al. (1952).As a result, we conclude that in an optimal solution, station 1 is the first priority responder to calls from j, station 2 is the second, and so on. Therefore, the demand nodes whose first priority responder is station w are those that are closest to w. Hence, we conclude that the first priority districts for the Base model are the same as 'send-the-closest-server' when there is no more than one server located at every station.

B.2 Proof of Theorem 2

Consider a demand node j and a station w that is the closest to j. We denote the distance between two points by D_{wj}. Consider A to be the set of all demand nodes whose distances to j is less than or equal to d_{wj}. Note that there will be no open station located within any of the nodes in A except for w, since w is closest to j.

Assume a second demand node, j', along the line between j and w such that j' is closer to w than j. Let B be the set of demand nodes whose distances to j' is less than or equal to $d_{wj'}$. The triangle inequality implies that for any node j'' within B, $d_{j''j} \leq d_{j''j'} + d_{j'j}$. Knowing that $d_{j''j'} \leq d_{wj'}$ and $d_{wj} = d_{wj'} + d_{j'j}$, it turns out that $d_{j''j} \leq d_{wj'} + d_{j'j} = d_{wj}$. Therefore, B is a subset of A and thus, there is no open station located within B and w is the closest station to j' too. As a result, w will be the first priority responder to the calls from j'. That means, if j's first priority responder is w, the first priority responder for any demand node along the line from j to w is also w. Hence follows the contiguity of the first priority districts.

Appendix C

Demand across different time periods

Eight time periods are considered in the computational examples. Figure 2.3 shows the aggregate demand across all time periods. Figure C.1 and C.2 illustrate the variations in the demand level between different time periods for weekdays and weekends, respectively. The red color indicates an increase in the call volume relative to the last time period while the blue color indicates a decrease. These figures show that during the weekdays (aside from the 12am6am period) the demand is more concentrated in the Southern region of the county, whereas the demand is more spread out during the weekend. This shift in demand necessitates different server locations and districts at different time periods.

Figure C.1a shows that there is an overall increase in the EMS calls in the mornings during the week, while the largest increase is in the middle of the county and around *Ashland* and *Mechanicsville*. Similarly, the demand rises in the middle of the county throughout the afternoon as shown in Figure C.1b. The opposite pattern starts taking place from 6pm and continues until 6am, as illustrated in Figure C.1c and C.1d. The overall demand starts decreasing throughout the county with the largest decrease happening in the middle of the county.

Similar to the weekdays, the overall call volume increases in the morning throughout the county in the weekends. However, this change, shown in Figure C.2a, in not concentrated in the center of the county as much as it is during the weekdays. The *Doswell* area, whose call volume starts increasing from 6am, faces a sharp increase in the demand at 12pm (Figure C.2b). This in fact is due to an amusement park that is located in this area whose peak times are at weekend afternoons. The call volume starts decreasing all over the county from 6pm on.

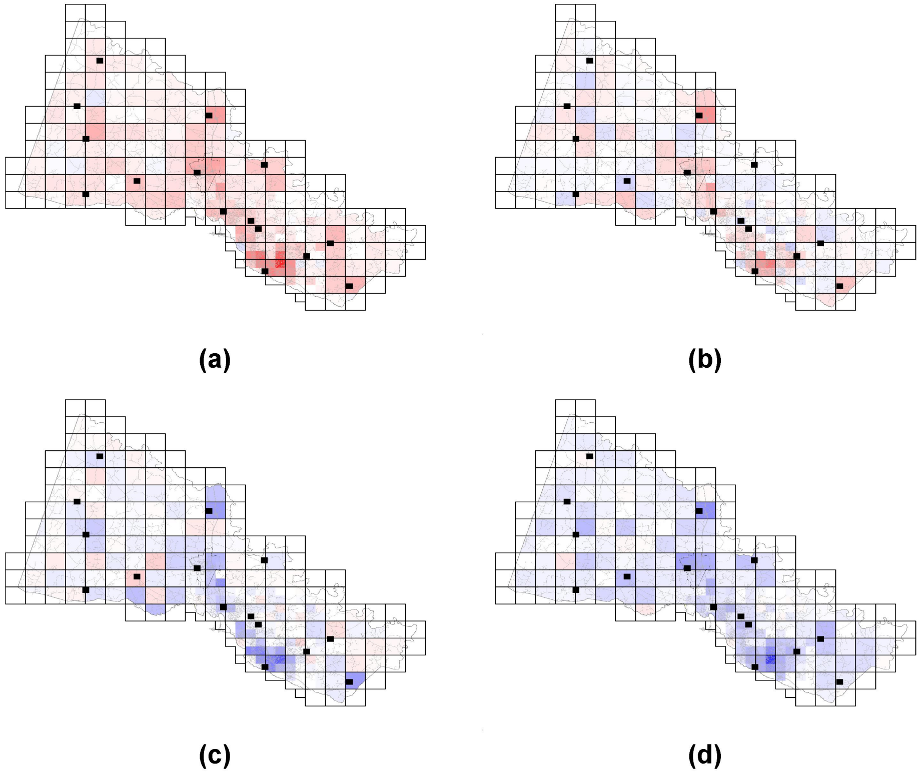


Figure C.1: The figure shows the variations of demand through different time periods during the weekdays. Red indicates an increase in the demand and blue indicates decreasing demands. (a) The amount of variation from 12am6am period to 6am12pm period. (b) The amount of variation from 6am12pm period to 12pm6pm period. (c) The amount of variation from 12pm6pm period to 6pm12am period. (d) The amount of variation from 6pm12am period to 12am6am period.

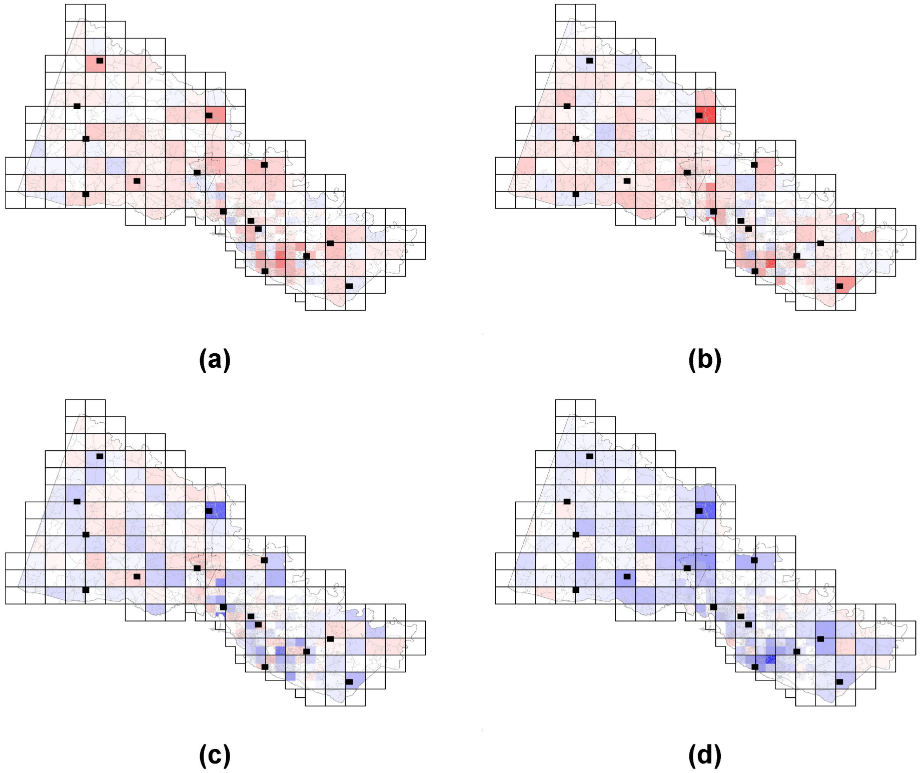


Figure C.2: The figure shows the variations of demand through different time periods during the weekends. Red indicates an increase in the demand and blue indicates decreasing demands. (a) The amount of variation from 12am6am period to 6am12pm period. (b) The amount of variation from 6am12pm period to 12pm6pm period. (c) The amount of variation from 12pm6pm period to 6pm12am period. (d) The amount of variation from 6pm12am period to 12am6am period.