POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Studies on Management of Emergency Service Systems

AKBAR KARIMI

Département de mathématiques et de génie industriel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*

Mathématiques

Août 2019

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Studies on Management of Emergency Service Systems

présentée par **Akbar KARIMI**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Antoine SAUCIER, président
Michel GENDREAU, membre et directeur de recherche
Vedat VERTER, membre et codirecteur de recherche
Fabian BASTIN, membre
Armann INGOLFSSON, membre externe

DEDICATION

to my family

ACKNOWLEDGEMENTS

First and foremost, I thank Michel Gendreau and Vedat Verter for giving me this opportunity and for the extended financial support over the years. They both have been extremely and undeservedly nice and accommodating to me, for which I feel equal parts fortunate and grateful.

The financial support for the research conducted by the author, which is partly reported in this thesis, has been jointly provided, with no specific order, by the Interuniversity Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), the Department of Industrial Engineering and Applied Mathematics of the Polytechnique Montréal, and the CREATE Program on Healthcare Operations and Information Management. Their support is greatly appreciated.

I am grateful to the kind, friendly and supportive staff at CIRRELT, the Polytechnique Montréal, the department of Orthopedic Surgery at the Jewish General Hospital, and other medical care centers I have visited in the last few years; in particular, I wish to extend my special thanks to Lucie L'Heureux, Lucie-Nathalie Cournoyer, Guillaume Michaud, Delphine Périé-Curnier, Melisa Regalado, Suzanne Guindon, Diane Bernier, Amal Bennani, Dr. Peter Jarzem, Martine Dahan, Jason Skolar, and Émillie Fournier.

My transition to life abroad was greatly eased by the company of my little circle of good friends with whom I share memories I will cherish forever.

I wholeheartedly thank Mont Royal Park and Saint Joseph Oratory for being places where, if you go at the right time, polarities meet, singularities form and the whole physicality thing takes the back seat, at least for a few fleeting moments.

Last, but not least, I send my love to my parents, for doing what parents do best, and for going the extra mile of creating two cute siblings for me, so I wouldn't need to go out of the house to find kids, of either gender, to fight with—sometimes, to the point of inducing near death out-of-body experiences—and later on, rely on as unbounded sources of trust and compassion, when trust and compassion are as hard to come by as practical applications for the ideas presented in this manuscript.

RÉSUMÉ

Forts des outils de la théorie des files d'attente, de la géométrie stochastique et des extensions développées en cours de route, nous présentons des modèles descriptifs de systèmes de services d'urgence organisés en fonction du potentiel de limitation explicite des distances de dispatching avec une fidélité accrue du modèle et une stratégie de dispatching pour atteindre des performances maximales avec des ressources limitées. En utilisant le terme «sauvegardes partielles» pour faire référence à des règles d'expédition avec des limites explicites sur les distances d'expédition, nous étendons d'abord le modèle classique de mise en file d'attente hypercube pour inclure des sauvegardes partielles avec des priorités. La procédure étendue pourra représenter les systèmes de services d'urgence dans lesquels le sous-ensemble de serveurs pouvant être envoyés à une demande d'intervention d'urgence dépend de l'origine et du niveau de service demandé. Cela permet de développer des modèles d'optimisation dans lesquels le concepteur du système laisse le choix des unités de réponse pouvant être envoyées dans chaque zone de demande et peut être intégré à l'espace de la solution avec d'autres variables de décision d'emplacement ou d'allocation. La nouvelle méthode descriptive et les modèles d'optimisation sur lesquels reposent les plans de répartition et de répartition optimaux correspondants devraient indiscutablement améliorer les performances et mieux refléter le comportement réel des répartiteurs lorsque la configuration instantanée du système constitue un facteur majeur dans la prise de décision. Par la suite, nous étendons notre analyse. des déploiements statiques couverts par le premier modèle vers des systèmes à relocalisation dynamique. En faisant des hypothèses d'uniformité sur les origines des demandes de service et les emplacements des unités d'intervention, nous développons un cadre théorique pour une évaluation rapide et aléatoire de la performance du système avec une politique de sauvegarde partielle donnée et des résultats donnés spécifiés en fonction du temps de réponse. Le modèle général permet de révéler tout potentiel théorique d'amélioration des performances du système en utilisant des stratégies de dispatching de secours partielles aux stratégies tactiques ou opérationnelles, sans connaître les détails de la méthode de relocalisation dynamique utilisée ni même de la distribution de la demande au-delà du taux total d'arrivée et de la densité. Nous présentons des résultats auxiliaires et des outils à l'appui de notre traitement des systèmes de service d'urgence avec sauvegardes partielles, notamment des notes sur les distributions de distance avec des effets liés et quelques lois de conservation du débit liées aux situations de file d'attente rencontrées dans le cadre de ce travail.

ABSTRACT

Armed with tools in queuing theory, stochastic geometry, and extensions developed along the way, we present descriptive models of emergency service systems organized around and emphasizing the potential of explicitly limiting dispatch distances in increasing model fidelity and as a dispatching strategy to achieve maximal performance with limited resources.

Borrowing the term "partial backups" to refer to dispatch policies with explicit limits on the dispatch distances, we first extend the classic hypercube queuing model to incorporate partial backups with priorities. The extended procedure will be able to represent emergency service systems where the subset of servers that can be dispatched to a request for emergency intervention depend on the origin and level of service requested. This allows for development of optimization models where the choice of response units eligible for dispatch to each demand zone is left to the system designer and can be integrated into the solution space along with other location or allocation decision variables. The new descriptive method and thus the optimization models built upon and the corresponding optimal location and dispatch plans, should arguably lead to better performance and better reflect the actual dispatchers' behavior where the instantaneous system configuration constitutes a major factor in making assignment decisions.

We next extend our analysis of static deployments covered by the first model to systems with dynamic relocation. Making uniformity assumptions on the origins of service requests and locations of the response units, we develop a theoretical framework for quick and dirty evaluation of the system performance with a given partial backup policy and a given outcome specified as a function of response time. The general model, makes it possible to reveal any theoretical potential to improve system performance by employing partial backup dispatching strategies at tactical or operational, without knowing the details of the dynamic relocation method used or even the demand distribution beyond the total arrival rate and the density per area.

Finally, auxiliary results and tools supporting our treatment of emergency service systems with partial backups are presented, which include notes on distance distributions with boundary effects and a few rate conservation laws related to the queuing situations we encountered in this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

ESS	Emergency Service System
EMS	Emergency Medical Services
FCFS	First Come, First Served
LCFS	Last Come, First Served
ASTA	Arrivals See Time Averages
PASTA	Poisson Arrivals See Time Averages
PPP	Poisson Point Process
u.i.d	Uniformly and Independently Distributed
i.i.d	Identically and Independently Distributed

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

The primary goal of an Emergency Service System (ESS), quite obviously, is to provide emergency care to customers in immediate need for it. The ability of the system to provide such services is thus a natural measure of performance. To achieve adequate performance, the system needs to be well designed and adequately resourced. The optimal performance, however, cannot be achieved without reliable tools to assess the suitability of a given tentative design candidate. Simulation models remain the gold standard to assess complex systems and allow the designer to evaluate design and operation scenarios with great flexibility in terms of the desired level of detail and accuracy. The downside to simulation, however, is the computational expense of scenario analysis which in some applications can be quite prohibitive. For example, consider a dynamically relocating emergency service system in which the dispatchers will send the available response units to new waiting stations every time time the number of free vehicles changes. In many cases, such relocation decisions are based on a real-time evaluation of different and potentially many candidate relocation plans of which the best ones are selected. The time needed for evaluation of these candidate plans can arguably become prohibitively long if a simulation model is used, unless it is carried out on an unusually capable computing platform. In these cases, mathematical models with their faster computation times come to our aid allowing a wider system optimization applications such as real-time dynamic relocation planning and, in general, less computationally demanding analyses that can be performed on less powerful computing systems or in shorter times. The popularity and range of various descriptive mathematical models incorporated into prescriptive optimization models proposed in the literature attests to the effectiveness of such tools in helping the system analysts in making better management decisions at tactical, operational, and strategic levels. If desired, detailed simulation models can be employed in further evaluation of an initial set of decisions obtained using these primary optimization models. We will briefly compare simulation and analytical approaches and their respective straights at the end of this chapter.

Regardless of the objectives and constraints of an optimization model, the quality of the solutions obtained directly depends on the validity of the underlying descriptive model used in representing the system at hand as closely as possible. Therefore, it will always pay off to further improve a given descriptive model so that it better reflects the realities of the system or operation we seek to optimize or analyze. The hypercube queuing model of Larson (1974) and its approximate form developed by, again, Larson (1975), is the classic method of modelling a spatially distributed queuing system, and in particular, an ESS. This

model, and its numerous extensions, have been widely used by researchers and practitioners alike in development of optimization models of various emergency service operations. The original version of the method relies on several restricting and unrealistic assumptions that can limit the range of situations in which the model can reasonably act as a proxy to the system considered. Over the years, several attempts have been made to relax some of these assumptions; however, the assumption that the closest free response vehicle will always be sent to an incoming request for service regardless of the origin of the call, and in particular, its distance from the dispatched vehicle, remains to this day. This is basically the assumption of *full backups* as opposed to a policy of *partial backups* in which each response vehicle is responsible only for providing service to an arbitrary subset of demand zones or service region. The assumption of full backups in most practical applications will be in disagreement with how the actual system is operated. In some cases, this might be due to the strict zoning or allocation strategies in place; for example, a fleet of police patrol cars where each patrol car is responsible for a well-defined sector of the service region. A full backup assumption is clearly not valid for this case. Physical limitations of the response units may as well impose a limit on the maximum travel distance, rendering the full backup assumption inaccurate. For instance, an ESS operation comprised of aerial drones will be best modelled as a partial backup system since the maximum flight range of virtually all drones available today is limited by the capacity of the batteries powering them. Finally, the full backup assumption may fail to reflect dispatchers' behavior in assigning response vehicles to incoming calls. To see this clearly, suppose a request for service is received where the closest vehicle is busy and the second closest response unit is free but is located much farther from the call origin. Now, if the dispatcher estimates the close-by vehicle to finish its current job in less time than it would take the second vehicle to arrive at the scene, then he or she will most probably wait for that busy vehicle to become free and then get immediately dispatched to the new call. Under the assumption of full backups, however, the farther vehicle will be selected for dispatch regardless of the unreasonably long travel distance and response time it results in. The assumption of full backups in this situation will of course be less realistic and valid when the outcome of the intervention is more strongly impacted by the response time. Responding to highly urgent requests such as incidents of Out-of-Hospital-Cardiac Arrests (OHCA) presents a prime example of a highly time-critical operation for which dispatching a vehicle over distances over a certain limit will result in very low probability of patients surviving. These observations establish the importance of relaxing the full backup assumption in analysis of emergency service systems and motivate us to develop new descriptive models or modify the existing methods in which the more realistic partial backup policy is explicitly incorporated into the model. The notion of partial backups is thus the main theme of the present work

and forms the basis of the models we develop in subsequent chapters.

To provide a more comprehensive treatment of emergency service systems, we consider both *static* and *dynamically relocating* deployments and try to develop models for performance approximation of the system with these respective deployment mechanisms. In a *static deployment*, the response units are assigned fixed waiting stations from which they are dispatched to locations of emergency incidents and travel back to when the service is completed. In an ideal static deployment, the response units are arranged such that the best possible coverage is provided to the entire service region. The notion of adequate or good coverage, however, can be interpreted in many ways and encompass different and often competing objectives. For instance, with the maximum coverage or outcome as the main objective, deployments in which response units are placed closer to areas with higher demand concentrations will typically yield the best results. On the other hand, if a measure of equity in service provision is instead selected as the performance objective, then the optimal deployments will normally have the response units more uniformly distributed over the service region to guarantee sufficiently reliable access to services independent of location. In a deployment with *periodic relocation*, the planning horizon is divided in several time periods each with their own set of optimal locations for the waiting stations. Each of these static deployments is optimized for the corresponding demand configuration observed or predicted for that period. The number of vehicles deployed in each period may also be different. The system then sequentially switches through these set of static deployments in an attempt to adapt to varying operating conditions over the planning horizon. The geographic distribution of demand or traffic conditions are examples of operating conditions that change with time and can be effectively addressed through periodic deployments. Periodic deployment plans can be easily constructed by repetitive applications of the static deployment models and thus, for the most part, do not need special mathematical devices beyond what is needed to account for the costs and constraints associated with transitions between the periods.

A deployment with *dynamic relocation* tries to maintain an adequate coverage of the service area by relocating the free response units to new positions whenever the number of available vehicles changes; that is, each time a vehicle finishes its current job and becomes available thus increasing the number of free units, or when a vehicle gets dispatched to a new incoming call thus decreasing the number of free units. Unlike the periodic redeployment which reacts to the changes in the underlying operating conditions, the goal of dynamic relocation is to continuously maintain adequate coverage of the service area with an ever-changing number of available server at any given moment. These systems can also be treated with static deployment models by first obtaining a pre-calculated set or sets of optimal static deployments for fleets of size $n = 1, 2, \cdots, N$ with N the actual number of deployable response units, and

then relocating the vehicles in real-time according to the optimal static deployment with n units whenever the number of free vehicles drops or increases to n. This set of pre-determined deployments is commonly known as a *compliance table* which is basically a look-up table of acceptable arrangements of response units for every number of available vehicles. Whenever a relocation is required, the dispatcher will re-position the currently available vehicles to new waiting stations according to a relocation plan he or she deems most appropriate for the situation among the candidates given in the compliance table. This approach does not require any real-time computation as the compliance tables are obtained offline.

A dynamic relocation deployment, however, can also be implemented in a more sophisticated fashion in which relocation decisions are computed in real-time whenever the number of available units changes. This approach has the clear advantage of taking into account the exact instantaneous state of the system in calculating new waiting spots for each of the free vehicles. This is important since other factors besides maintaining adequate coverage should be taken into consideration while making relocation decisions. Most notably, a good relocation decision should preferably minimize both the number of vehicles that should be relocated and also the corresponding travel distances. Compared to compliance tables, real-time computation of the new waiting spots provides a greater flexibility for incorporating all these contributing aspects in generating best relocation decisions. Although the real-time calculation of the optimal locations requires considerably higher computational power compared to the compliance table method, the actual problems to be solved can still be considered as instances of static deployments with appropriate constraints and objectives reflecting what the manager considers acceptable and desired in a good relocation plan.

As we see, the ability to solve static deployment problems is all we need for analysis and design of static, periodic, compliance table and real-time dynamic deployments. This highlights the importance and value of developing more accurate mathematical methods such as the hypercube queuing model for evaluation of static deployment scenarios. As stated earlier, the quality of the location and relocation plans and compliance tables obtained will be directly dependent on the quality of the underlying mathematical model used to describe the steady-state behavior of the system for a given static deployment. In the first main contribution of the thesis, presented in Chapter 4, we take a step towards this very objective of increasing the accuracy and applicability of the approximate hypercube queuing model by relaxing the assumption of full backups and allowing for partial backups with priorities. As we will see, the original version of the approximate hypercube queuing model will lead to large approximation errors when used to estimate the steady-state behavior of a system with partial backups. The proposed extension of the hypercube model, however, performs well in predicting the equilibrium behavior of the system with priorities and partial backups. This opens a door to

new optimization models in which we augment instances of system deployments by specifying the subset of demand zones covered by each response unit in addition to the location in which each unit is deployed. This more refined view of the system combining the location and allocation decisions, allows for a more realistic and detailed description of the actual deployment scenarios encountered in practice. Armed with the extended hypercube model with partial backups, one can now predict the steady-state behavior of the system for a given set of tentative location and allocation decisions. Evaluation of many such sets of combined decisions within an optimization loop with appropriate objectives and constraints will then give us an optimal location and allocation decisions we need for a static deployment.

While the extended hypercube model developed in Chapter 4 aims at accurate prediction of the performance of system deployments with partial backups, the mathematical model presented in Chapter 6 tries to quantify the impact of partial backup policies on the performance of systems with real-time dynamic relocation. Making simplifying assumptions, including the uniformity of the distribution of call locations and of the response units, we build an abstract model that can be used to obtain the expected performance of the system operating under a given partial backup dispatch policy. The partial backup policy in this case is defined as the maximum number of neighboring response units considered for dispatch alongside a limit on the maximum dispatch distance for the rest of the units. Finally, Chapters 5 and 7 contain some supporting results used in development of the main models described above.

1.1 Basic concepts and definitions

In this section, we briefly go over some basic concepts related to the topics discussed in this thesis. These introductory sections are mainly aimed at a reader not familiar with the basics of queuing theory and related topics. We also clarify the terminology used in the text. We first give a basic overview of queuing theory including an introduction to Kendall's notation for queuing systems followed by an introduction of spatially distributed service systems and the tools we have for studying these systems. We then highlight the value and applications of analytical and simulation models in studying service systems. Finally, we provide a concise statement of the purpose and objective of the dissertation together with a plan of the thesis.

1.1.1 Queuing theory

The main contributions of this thesis rely heavily on queuing theory, a discipline within the mathematical theory of probability, devoted to the study and prediction of waiting lines and waiting times. Queuing theory results are most often used in operations research to aid in

making decisions about the resources required to provide a service with a certain level of quality. Besides operations research and industrial engineering, queuing theory ideas have been widely applied in telecommunication, traffic engineering, computing, forestry and many more design and management problems.

A queuing station or node (or simply a queue) can be viewed as a black box representing a service station where customers arrive, stay for a while to receive service, and depart once the service is completed. A supermarket checkout line is a cliched example of a queuing node (or system). Customers arrive, possibly wait in a line for their turn, eventually get processed by the cashier and then leave.

Usually, we have some information regarding the inside of a queuing model; at the very least, the number of servers, that is, for example, the number of cashiers in the supermarket line, should be specified. In a multi-server queue, that is a queue with more than one servers, any server can process an arriving customer and can start working on a new arriving or waiting customer after the current service is completed. This is the most usual situation and can be used, for example, to represent a single waiting line in a bank processed by multiple tellers. In other multi-server models, however, we might have different customer and server types with policies specifying the types of customers that can be processed by each server type. Call centers are a good example of this kind of queuing systems where agents are trained in one or more specific areas and thus can only accept callers who request a certain type or types of services. This is generally known as a *skill-based routing* as customers are assigned to suitable servers based on their matching skills. In a multilingual call center, for example, a skill-based routing may be used based on the languages spoken by the agents and the customers. Queues with skill-based routing are remarkably difficult to analyze with limited relevant theoretical results published in the literature. We note that the queuing system with partial service introduced in Chapter 4 and revisited in Chapter 6, can be considered queuing models with skill-based routing. In this case, the skill-based routing is determined by the distances between customer and server locations. In fact, as soon as we employ a partial backup dispatch policy in any emergency service system, and put an upper limit on the allowable distance between a call for service and the server dispatched to the call, a skill-based routing is observed since only a certain types of calls, in this case calls close enough to each server, can be processed by each server. The queuing system with partial service can then be used to model the actual system.

In addition to the number of servers and the type of routing, we also need to know what happens to customers who find no free (available) servers upon arrival. In a *loss system* or a queuing system with no waiting area, that is with zero queue capacity, customers who find

no available servers upon arrival will leave without receiving service. We will refer to these customers as *lost* customers or calls. The significance of loss systems in practical application will be discussed later in this chapter. In a system with a queue discipline, sometimes simply referred to as a *queuing system*, customers who find no free servers upon arrival will join a queue of waiting customers who will later enter service one at a time as servers finish their current jobs and become available again. The order in which these waiting customers are processed by the servers is also specified in most cases as the *scheduling policy* or the *service discipline*. For instance, with a First-come-first-served (FCFS) service policy, the waiting customer with the earliest arrival time (hence the longest waiting time) will enter service first. This is of course the exact opposite of a First-come-last-served (FCLS) policy. Other common scheduling policies include Service-in-random-order (SIRO) and shortest-job-first (SJF). In all of these service disciplines, a server can process one job at a time; however, a service discipline may allow for servers to work on multiple jobs at the same time; for example, in a *processor sharing* policy, all arriving customers enter service immediately with the total service capacity equally shared between them. A computer CPU working on several computing jobs with equal priorities is an example of a system with processor sharing policy.

A *priority queue* is a queuing system with multiple customer types (or classes) each with its own priority level in entering service. The scheduling policy within each priority can be also specified. Priority queues can be either *pre-emptive* or *non-preemptive*. In a pre-emptive priority queue, a job in service can be interrupted by an arriving customer with a higher priority. In a non-preemptive priority, on the other hand, higher priority customers cannot interrupt any ongoing jobs even those with lower priorities. In either case, it is often assumed that no work is wasted as the service to the lower priority job is presumed from where it was left once the higher priority job is finished. Pre-emptive priority queues are most often encountered in computer systems, for example, when lower priority computing tasks are processed only in time periods when no higher priority jobs exist. The priority emergency service systems we consider in this thesis, however, are all of the non-preemptive type. It should be easy to see how dropping a lower priority patient or customer in favor of a more critical call can be impractical, unreasonable and wasteful of system resources in an actual emergency service system.

Kendall's notation proposed by D. G. Kendall in 1953 is a standard method of specifying details of a single queuing node. The original Kendall's notation describes the queuing node using three elements written as A/S/c, where A describes the distribution of times between successive job arrivals, S describes the distribution of service times (time to complete a single job), and c is the number of servers. The possible values for A and C pertinent to our work here are: M, specifying a Markovian or memory-less process which basically

indicates an exponential distribution for the times between successive arrivals (that is A=M) or successive job completions of each server (that is S=M); D, specifying a deterministic time between successive job arrivals (that is A=D) or deterministic service times (that is S=D); and G, specifying a general distribution for the inter-arrival times (that is A=G) or service times (that is S=G). This basic notation has been extended since its introduction; in particular, we will use the extended notation A/S/c/K where K is the capacity of the system defined as the maximum number of customers in service plus those in the waiting area; in particular, a loss system where queues are not allowed is represented as A/S/c/c, indicating a maximum of c customers in system and hence zero waiting line capacity, and a queuing system with no limit on the number of waiting customers is denoted by A/S/c/∞.

Throughout the text, we will deal with basic M/M/N/N and M/M/N/∞ models that indicate exponential inter-arrival times, exponential service times, N servers, and a maximum system capacity of either N (implying a zero waiting line capacity; that is, a loss system) or infinity (indicating a queuing system with unlimited number of waiting customers). We note that an exponentially distributed time between successive arrivals with a mean value of $1/\lambda$ indicates that the arrival of customers is governed by a Poisson process with intensity parameter λ that is also equal to the average number of arrivals per unit time. Likewise, an exponentially distributed service time with a mean value of $1/\mu$ indicates a service rate of μ defined as the average number of jobs a server can complete per unit time when working non-stop during a *busy period*. A busy period, as the name suggests, is defined as a period of time during which the servers are busy.

For the M/M/N/N system, also known as the Erlang-B model, the probability of n servers being busy is given as

$$p(n) = \frac{(\lambda/\mu)^n n!}{\sum_{i=0}^{N} (\lambda/\mu)^i i!}, \quad n = 0, 1, \cdots, N. \tag{1.1}$$

Since this is a loss system and no waiting queues are allowed, the probability that $n = 0, 1, \cdots, N$ customers are in the system (receiving service) is also given by (1.1). The *blocking probability* or the *loss probability* is the probability of an arriving customer finding all servers simultaneously busy and thus not being able to receive service, and is given by the Erlang-B formula; that is

$$P_B = \frac{(\lambda/\mu)^N N!}{\sum_{i=0}^{N} (\lambda/\mu)^i i!}.$$

The probability of $k = 0, 1, \cdots$ customer being in an M/M/N/∞ queue, also known as the

Erlang-C model, is obtained as

$$p(k) = \begin{cases} \frac{(\lambda/\mu)^k}{k!} p_0 & 0 < k < N\,, \\ \frac{(\lambda/\mu)^k N^{N-k}}{N!} p_0 & N \leq k\,, \end{cases} \qquad k = 0, 1, \cdots,$$

with p_0 the probability of no customers in service (all servers free) given by

$$p_0 = \left[\frac{(\lambda/\mu)^N}{N!} \frac{1}{1 - \lambda/N\mu} + \sum_{i=0}^{N-1} \frac{(\lambda/\mu)^i}{i!} \right]^{-1}.$$

The number of customers in system includes those in service and those waiting in queue; therefore, $k > N$ customers in system indicates a queue of $k - N$ waiting customers and N customers in service. The *queue probability* or the probability of an arriving customer finding all servers occupied and thus being forced to join the queue is given by the Erlang-C formula; that is

$$P_Q = \left[1 + (1 - \frac{\lambda}{N\mu}) \frac{N!}{(\lambda/\mu)^N} \sum_{i=0}^{N-1} \frac{(\lambda/\mu)^i}{i!} \right]^{-1}.$$

Finally, the average time a customer spends in system is

$$T_{\text{system}} = \frac{P_B}{N\mu - \lambda} + \frac{1}{\mu}\,,$$

where the first and second terms are the the average time spent in queue and in service, respectively.

1.1.2 Spatially distributed service systems

The emergency service systems we consider in this thesis belong to the class of spatially distributed service systems. Besides emergency service systems (such as Emergency Medical Services (EMS), fire and police), other examples of specially distributed service systems in an urban environment include door-to-door pickup and delivery services (for example, mail delivery, waste collection), community service centers (for example, libraries, outpatient clinics, social work centers), and transportation services (for example, bus and subway services, taxicab services). Similar to regular queuing systems, congestion is likely to arise in spatially distributed service systems because of the uncertainties in demand, service requirements and the finite amount of resources allocated to providing those services. For the spatially distributed systems, however, uncertainties appear not only in the arrival time and the duration of requested services, but also in the location of the demand for service as well; therefore, the geometric structure of the service area, which is the city or part of the city, will have a sig-

nificant impact on the system performance. While the spatial nature of demand distribution usually requires tools from geometric probability, the congestion inherent in these systems calls for a queuing theory type of analysis. These systems are therefore sometimes referred to as spatially distributed queues.

There are two main types of spatially distributed service systems. In a *server-to-customer* system, servers travel to customers' locations to provide the requested services. Emergency service systems and mail delivery are examples of this category. In a *customer-to-server* system, customers have to travel to server locations such as a public library or a clinic, to receive service. The emergency service systems we focus on in this thesis are server-to-customer systems, where servers are the mobile *response units* or *emergency vehicles* distributed over the service area (or service region) each with its own specified waiting location (or station). Requests (or demand, or calls) for service arrive from within the service area with a given distribution. These are the customers of the service system. The distribution of the demand over the service area is usually represented as a set of demand points (or nodes, zones, atoms) scattered over the service region each with a given arrival rate. This is a discrete approximation of the actual distribution of demand and can be constructed based on the historical data of the actual arrivals to the service system over a period of time. The total arrival rate is then equal to the sum of the arrivals from the demand zones.

In response to a call arrival, the dispatcher selects a response unit (most probably the closest to the call location) and assigns (dispatches) it to the call. Service time is measured from the moment the server is assigned to a call until the call is released either at the scene of the emergency or at the hospital and the server is available again. Depending on the system, service time may be broken down into several components or steps. In the most general case, this includes getting ready to start the travel to the call location (becoming *en-route*), travelling to the call location, providing the on-scene care, transporting the patient to a nearby hospital or care center, if needed, and a turnaround stage where the vehicle prepares to get back in service again. The service time is thus equal to the sum of the times spent in each of these steps. Information about the distributions of the service time components may be available from observations of the real system or from empirical models developed for this purpose. Figure 1.1 shows a typical break-down of service time into its components.

Since the system is spatially distributed, the distribution of service time components may depend on the locations of the server and the customer (in addition to service-to-service variations captured by the distribution). The most obviously location-dependent components of the service time are the travel and transport times. In the regular queuing models, servers are *indistinguishable*; that is, they do not possess any characteristics beside being available

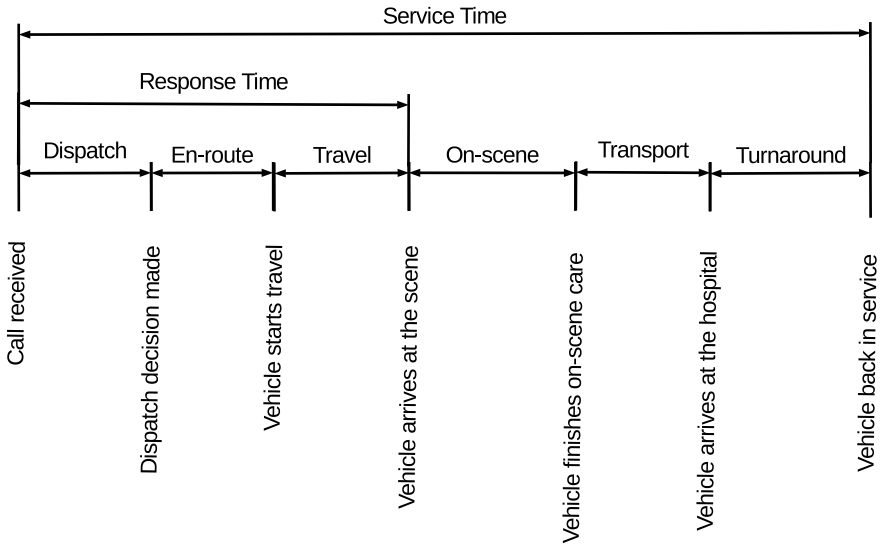


Figure 1.1 Service time components

or free, to identify them from one another. In a spatially distributed queue, however, servers are indeed *distinguishable* by their individual characteristics such as different average service times and workloads. The average service times will be different because of the different *dispatch rates* which are the long-run average frequency with which each server responds to calls from each demand zone. Different dispatch rates lead to different average travel and transport times, and consequently different average service times. For example, a server located in a part of service region with higher demand density will have to travel shorter distances and thus will have a lower average service time than another server located in an area with lower demand density. The average service time of a given server multiplied by the average number of dispatches sending that server to respond to calls gives the average fraction of time the server is busy. This is called the *utilization* or the average *workload* of the server. Therefore, servers may have different workloads in addition to different average service times. The main goal of the analysis of a spatially distributed system with a given configuration (set of demand locations with intensities and the set of servers) therefore is to predict the dispatch rates of each server to different demand locations. The server-specific parameters (such as the average service time, the average travel time, and the average workload) and the system performance measure can be then easily computed once the dispatch rates are known. The performance of emergency service systems are usually, and not surprisingly, a measure of the random response time, which is defined as the time period from the moment a call is received until the emergency vehicle arrives at the call location. Typically, desired performance objectives for emergency systems are specified as a minimum fraction of

dispatches with response times below a given threshold.

Emergency service systems may operate as loss or queuing systems. The queuing system is suitable when the system under consideration is the only one capable of providing the required services and the nature of the emergency allows for delays in service caused by possible queuing. An example of this scenario would be a fleet of roadside emergency repair vehicles deployed along a highway where occasional delays in the arrival of the repair vehicles would not result in critically negative outcomes. The loss system, on the other hand, is typically used to represent a primary service provider backed up by one or more supporting systems operating in parallel. This might be the case, for example, for the main ambulance service operating in a city with the additional support from fire engines and police patrol cars. In this scenario, any urgent call that the ambulance service is not able to respond to immediately, will be handed off to the backup service, either the police or fire, which will then try to respond to the call in the shortest time possible. Therefore, from the perspective of the main system, the call is in fact lost, although in reality, it has been transferred to another system to avoid queuing delays and improve the response time. Service systems dealing with highly critical emergencies, such as EMS dealing with incidents of cardiac arrests, are best modelled as loss systems.

1.1.3 Discrete event simulation

In a discrete event simulation, we model the operations of a real system as a sequence of events each happening in a specific instant in time (occurrence time) and potentially triggering a change of the system state. The state of the system, regardless of its definition, will stay the same between consecutive events. This allows us to jump from each event to the next while keeping track of the evolution of the system state over the entire simulation time. More specifically, starting from an initial system state and a sequence of upcoming events, we keep moving the simulation time to the next upcoming event while updating the system state and the sequence of upcoming events accordingly and following the protocols and rules governing the real system being simulated. For the type of emergency service systems considered in Chapter 4, the state of the system is comprised of the server locations, busy statuses of each server, and a list of waiting customers with their locations and priority levels. As mentioned earlier, the service time is often broken into a sequence of components which typically include getting ready to travel to the call location, travelling to the call location, delivering on-scene care, transporting the patient to a hospital if needed, and traveling back to the station. The events in this case are therefore *a new call arriving* or a *busy server finishing a step* of an ongoing service and moving on to the next stage. A busy server becomes idle when

it finishes the last step of its current service. Every time the simulation clock (simulated time) is moved to the occurrence time of the next event, we generate a new event of that type and place it in the sequence of upcoming events. Random number generators are used to obtain the occurrence time and other properties of these newly generated events. For instance, if the current event is a new call arrival, we construct a new call arrival event by randomly generating values for its location, priority and arrival time and then place this newly constructed event in the sequence of upcoming events. In addition, if the new arrival is compatible with a server, then we assign the server to the call and construct a *server becomes en-route* event with an occurrence time determined by adding the randomly generated *en-route time* (based on the distribution and the average value given for the *en-route time*) to the current simulation time, and add this new event to the sequence of upcoming events. Similarly, if the current event is a server finishing a service step while there are still more steps to perform, then we assign a randomly generated value to the duration of the next service step based on the parameters given for that specific component of service time (specified distribution, average value, etc) and add the corresponding event to the sequence of upcoming events. For instance, if the current event is a *server starting travel to call location*, then we compute the travel time based on a travel time estimation model and the distance between the current location of the server (usually the waiting station) and the call location. A *server arrives at the call location* event with the occurrence time equal to the current time plus the travel time, is then added to the sequence of upcoming events. With the sequence of upcoming events updated, we will have the time of the next event. The next step is to update the system state for the time period between the current and the next event. For example, for an event of a new call arrival, the closest compatible server will change status from *free* to *assigned but not yet en route*. If the new call arrival is not compatible with any server, then neither the system state nor the sequence of upcoming events will change in this period. Beside this basic framework, we may have other data structures to keep track of the simulation time, the number of iterations or to compute the desired outputs either as average values or distributions. This simulation approach in which we jump directly to the time of the next event is called a *next-event time progression* which is used in the simulation experiments of Chapter 4. The general framework of a discrete event simulation is depicted in Figure 1.2.

There is, however, another approach called *fixed-increment time progression* in which the simulation time advances in small constant steps with the system state updated based on the events happening during each time step. This approach does not allow for jumps to the occurrence time of the next event and hence can be considerably slower than the simulation with a next-event time progression; however, if the simulation of the system requires detailed

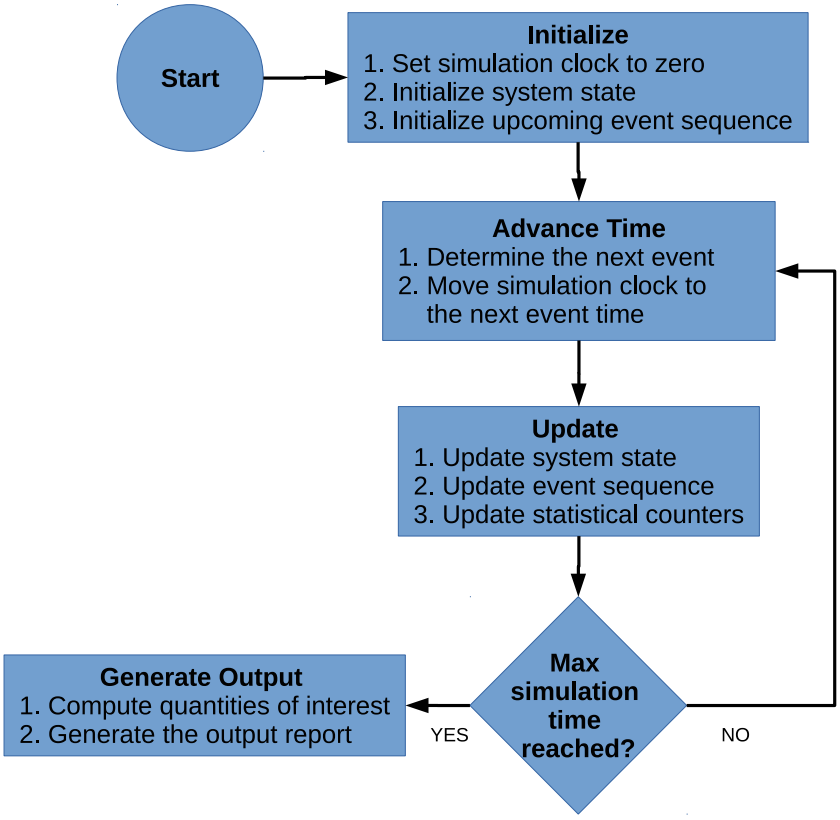


Figure 1.2 General flowchart of discrete event simulation

instantaneous information on the processes or activities happening within a given time period, then a fixed-increment time progression might be needed to simulate these activities in an accurate fashion. For example, one can use a fixed-increment time simulation to study the traffic in a network of roads. In this case, the system state will be the set of vehicles in the network and their locations and travel velocities or accelerations which should be updated in small enough time increments to provide an accurate representation of the dynamics and flow of vehicles moving around the network. In Chapter 6, we treat emergency service systems with dynamic relocation in which the dispatching decisions are based on the instantaneous locations of the moving response units and their distances from the random arrivals to the system. Therefore, we use a fixed-increment time progression to keep track of the current location of any of the moving response units along their respective travel paths. On the other hand, we still need to make system state updates at times where events such as call arrivals and service completions happen. This leads to a simulation model with hybrid time progression in which the simulation time follows a next-event progression when no vehicles are moving, and a fixed-step time increment is used between consecutive events if at least one response unit is relocating. This allows us to minimize the computational expense of the simulation by applying the fixed-increment time progression only when it is needed.

1.1.4 Analytical versus simulation models

Virtually, every real-world system can be studied through an appropriate simulation model developed to replicate the inner mechanisms of the system with a desired level of detail. The wide applicability and great flexibility of simulation models are the two attractive features of analysis through simulation. The usual downside of simulation studies is the high computational expense which naturally increases with increasing level of detail.

We can also develop or use existing mathematical models, such as those proposed in this thesis, that also aim at describing a real system and replicating or approximating its behavior under different operational conditions and parameters. These analytical models are usually applicable in a much more limited range of situations and typically offer far less approximation accuracy with equally limited room for extended realism. As is often the case, the very existence of an analytical model for describing a real-world system or operation relies on making simplifying assumptions about the system at hand that can be potentially quite unrealistic. For example, the assumptions of Poisson arrivals and exponentially distributed service times are both frequently encountered in queuing theory; while the former can be a reasonable approximation of the arrival process in most applications, this is not the case for the latter as exponential distributions can rarely be a realistic approximation of service

times in real-world systems. On the plus side, the computational effort needed for analysing a system with analytical models is usually much less than performing the same analysis using simulation. This trade-off between the accuracy and flexibility of modelling and the associated computational expense, justifies the importance of these approaches as individual tools in analysis and also points towards hybrid analytical-simulation approaches in which we exploit the advantages of each method.

In many application scenarios in which a highly accurate and detailed representation of a system is desired, a simulation model may be the only viable option. For example, a one-off evaluation of a fixed design or set of decision variables for a given problem can very well be performed via an arbitrarily detailed simulation model; in this case, the high level of accuracy and the potentially long run times will be well justified without posing a significant problem. On the other hand, analytical models have the unique ability of immediately revealing the mathematical relationships between the key system parameters and thus will be the obvious choice if gaining fundamental insight into the problem at hand is desired. It is still possible to obtain this kind of knowledge about the interplay of system parameters through simulation as well; however, this makes for a less rigorous and elegant approach and usually involve a large set of numerical experiments that can become prohibitive depending on the problem scale, level of detail, and the number of experiments required.

In addition to the exclusive applications of simulation and analytical models, the accuracy of the first approach and the low computational cost of the second can be taken advantage of through a standard hybrid method of solving optimization problems. This solution framework consists of two major steps; in the first step, we solve the optimization (prescriptive) problem based on an analytical descriptive model of the system at hand. The solutions obtained in the first step are then used to construct a restricted search space which is considerably smaller than the original search space and hopefully contains the optimal solution. A simulation-based optimization or scenario evaluation is then used to further explore this restricted solution space and improve upon the primary results. This hybrid approach is particularly useful in situations where the analytical approximation models used in the analysis do not adequately reflect the realities of the system under study and thus lead to solutions with questionable reliability. By using the hybrid analytical-simulation approach, we identify the set of promising solution candidates via the fast analytical approximation model and then use a detailed simulation study to screen this small set of tentative solutions and select the best one.

1.2 Research objectives

With the preliminaries given in the previous section, we are now in position to state the main purpose and objective of the current study. The models and tools proposed in the literature for the analysis of spatially distributed service systems rely on the fundamental assumption that each server can respond to calls from any demand location. We refer to this assumption as *full backups* as opposed to what we call *partial backups* where only a subset of servers are eligible for dispatch to calls from a given demand location.

We observe that the assumption of full backups can be unrealistic and ineffective in reflecting the behavior of dispatchers in real emergency systems and thus limit the range of applications of the analytical models based on this assumption. This motivates us to close this gap and develop analytical models for effective prediction of the behavior of static emergency system deployments with partial backup dispatching policies. As discussed before, analytical models of static deployments will enable us to analyze non-static deployments as well.

The basic reason why the assumption of full backups does not generally agree with real-life emergency service operations directs us towards our second objective. The key observation here is that human dispatchers do not tend to send servers over long distances simply because it may lead to a waste of resources and a degradation of service quality. In other words, placing upper limits on travel distances as reflected in partial backup policies should be seen as a naturally and intuitively efficient way of using system resources. We are thus interested in quantifying the relationship between the system performance and dispatch policies that follow partial backups and put limits on travel distances. To this end, we develop an analytical descriptive model of an emergency service system with dynamic real-time relocation and partial backup dispatching policies. Using this model, we can then take the basic configuration of the system along with the expected outcome as a function of response time, and determine if and how the system performance can be improved by employing a partial backup dispatch policy. The goal here is to immediately reveal any opportunities for performance optimization via partial backups rather than an accurate and detailed representation of the system.

The above models complement each other as first steps towards our general goal of more efficient and realistic study of emergency service systems and consequently better management strategies for achieving the best possible performance.

1.3 Plan of the thesis

The thesis is organized as follows. A brief literature review is given in Chapter 4.2 followed by a synthesis of the work as a whole provided in Chapter 3. In Chapter 4, we give our first mathematical tool for priority emergency service systems with static deployments and partial backups. We consider emergency service systems with real-time dynamic relocation and partial backup dispatch policies in Chapter 6 and present the our second analytical model. Chapter 5 and Chapter 7 provide supplementary results that we use in our development of the above mentioned models. Finally, a general discussion of the results is given in Chapter 8 followed by concluding remarks in Chapter 9.

We note that the content of Chapter 4 has been published as Karimi et al. (2018).

CHAPTER 2 LITERATURE REVIEW

The relevant literature has been cited throughout the text. Therefore, to avoid repetition, we suffice to briefly summarize the survey papers covering topics discussed in this thesis.We start by papers dealing with the general topic of this thesis, which is the management of emergency service system or in particular emergency medical systems. A recent survey of EMS optimization models is presented by Bélanger et al. (2019) which focuses on the location, relocation, and dispatching decisions and the interaction between these important aspects of EMS management. Aringhieri et al. (2017) provide an extensive and integrated review of the literature covering not only the location, relocation, and dispatching decisions, but also other segments of the emergency management and care pathway such as demand forecasting, routing policies, emergency department management, and work-flow planning. Earlier reviews of the EMS management can be found in Ingolfsson (2013), Bélanger et al. (2012), and Brotcorne et al. (2003). These survey papers focus on the planning and management challenges surrounding EMS operations or the operations research tools developed to address those challenges. Simulation models as an important tool, however, are usually not treated with the same detail as the mathematical models. Fortunately, Aboueljinane et al. (2013) cover this front and provide an overview of the simulation models applied to EMS operations.

In Chapter 4 we extend the approximate form of the hypercube queuing model. The hypercube model was originally described by Larson (1974) to replace simulation models in estimation of the steady state behavior of spatially distributed systems. He also proposed an approximation of this model to overcome the prohibitive computational costs of the full model (Larson 1975). Excellent reviews of the published work on the development of this descriptive model and its approximate form is presented in Larson (2013) and Galvao and Morabito (2008). In Chapter 4 we cover the literature related to the approximate form of the hypercube model in more detail and put our contribution in perspective.

In Chapter 5 we deal with distances between random points which is an extremely vast topic with an extensive literature scattered across many different disciplines ranging from physics to engineering and operations research. As a result, many of the results have been rediscovered by different authors. Fortunately, Moltchanov (2012) makes an attempt at providing a unifying overview of the main concepts and the basic results. Tong et al. (2017) gives a more recent discussion of the state-of-the-art approaches in modelling of distance distributions, their applications in ad-hoc network problems, and open challenges.

In Chapter 6 we consider emergency service systems with dynamic relocation and study the effects of operation under a partial backup dispatch policy instead of a the usual full-backup policy. To the best of our knowledge this concept has not been proposed in the open literature. Finally, extensions of the Little's law for queuing systems operating under certain conditions are given in Chapter 7 that connect the number of waiting customers and the duration of the queuing delay conditional on the system state observed upon arrival or while waiting. We refer the reader to Little (2011) which covers the major developments of the law alongside a clear demonstration of the concept and applications. The main result we derive in Chapter 7 is a form of distributional Little's law between the queue length and waiting times conditional on the system states upon arrival or during the wait. The original distributional Little's law was proposed by Haji and Newell (1971) while its importance and applications are discussed in detail by Keilson and Servi (1988) and Bertsimas and Nakazato (1995).

CHAPTER 3 SYNTHESIS OF THE WORK AS A WHOLE

As stated in the introduction, the main theme of this thesis is the potential performance enhancements possible by adopting an operation policy in which a maximum limit is imposed on the distance between the location of an incoming request for emergency services and the response units that a dispatcher will consider for dispatch. This is in contrast to the traditional full backup policy in which all response units will be dispatch candidates regardless of their distance from the scene of the incident. The chapters in this text are thus organized around this central idea and supporting ideas which are used as intermediate steps towards the developments of the main mathematical models presented in Chapter 4 and 6 in which we look into mathematical models incorporating partial backup dispatch policies in emergency service systems with static deployments and systems with dynamic relocation, respectively.

The hypercube queuing model of Larson (1974) and its less computationally expensive approximate form proposed by Larson (1975) himself, remains the standard method of modelling emergency systems with static deployments. A natural approach to incorporate partial backups into systems with static deployments, will be o extend the hypercube queuing model to allow for partial backups. Now, the hypercube queuing model uses an M/M/[N] queuing model as its core element to estimate the distribution of the number of busy response units; therefore, our extension of the hypercube queuing model naturally leads to an extension of the M/M/[N] queuing model to incorporate the partial backup dispatch policy. This has been done in Chapter 4 by introducing the queuing systems with partial service which replaces the M/M/[N] model of the hypercube model. To analyze the steady state properties of the queuing model with partial service, we used the theory of skill-based queues which allow product-form solutions for the state probabilities. Therefore, Chapter 4 is self-contained and does not depend on the material presented elsewhere in the thesis. However, in earlier versions of the model presented in Chapter 4, to derive the state probabilities of the queuing model with partial service, we used an alternative approach based on the extensions of the Little's law presented in Chapter 7. Beside providing an alternative method of obtaining the state probabilities of the queuing model with partial service, we consider the results given in Chapter 7 to be of independent theoretical interest. In addition, we conjecture that these results may be used to derive an approximation of the skill-based queuing systems with general configurations, although we do not pursue this possibility in the current version of this thesis.

In Chapter 6 the central idea of partial backups is extended to emergency systems with

dynamic relocation. The mathematical framework we develop for analysis of these systems is significantly more theoretical than the model presented in Chapter 4 and is based on simplifying assumptions we make; in particular, uniformity of the distribution of the call locations and the response units. This model requires closed-form expressions for the distribution of distances neighbors. Simple and classic expressions for these distributions obtained through a Poisson point process assumption already exist in the literature and we do use these equations in our derivation of the model. However, we still opt to obtain and use more accurate and theoretically valid distance distributions alongside the expression based on the Poisson point process. This is due to the fact that the model developed for systems with dynamic relocation, similar to the previous model in Chapter 4, is based on the queuing model with partial service and our numerical experiments shows that distance distributions with boundary effects obtained in Chapter 5 leads to significantly better approximations to the input parameters to the queue with partial service. We note that, unlike the steps in which we resort to Poisson-based expressions, the use of the fairly cumbersome edge-corrected expressions in this fashion constitutes a purely numerical step and thus not complicate the overall algorithm. Therefore, we can consider the material in Chapter 5 as a pre-requisite to Chapter 6 within the context of this thesis. However, a quick survey of the literature will reveal the general interest in and wide applications of results related to inter-node distance distributions in many different field s of science and engineering. Although the most interest in the topic has been drawn by applications in wireless communication and ad-hoc random networks, the work presented in Chapter 6 can be considered an application of such results within the field of operations research.

CHAPTER 4 ARTICLE 1: PERFORMANCE APPROXIMATION OF EMERGENCY SERVICE SYSTEMS WITH PRIORITIES AND PARTIAL BACKUPS [1]

Abstract

An extension of Larson's approximate hypercube procedure is presented for priority emergency service systems with partial backups where requests for service with a given origin and priority can be responded to by a certain subset of response units. We introduce and analyze the family of queuing and loss systems with partial service and use them to approximate the distribution of the number of busy response units which in turn enables us to estimate the average server workloads, immediate and delayed dispatch rates, and waiting delays as system performance measures. We consider systems with zero and infinite queue capacities and let the dispatching policies and service times depend on call priority and customer and server locations. The validity of the approximation model and its computational aspects are studied through numerous tests and a realistic application of locating a fleet of ambulances in downtown Montreal, Canada.
keywords: emergency service systems, partial backup, hypercube queuing model, spatially distributed systems, server-to-customer systems, priority queues

4.1 Introduction

Server-to-Customer systems represent an important class of spatially distributed queuing systems due to their strong presence in modern urban settings ranging from Emergency Service Systems (ESS) including ambulance, fire, police, and repair, to non-emergency applications such as non-scheduled home visits, demand-responsive delivery operations, and dial-a-ride transport systems. The nature of ESSs, which involves risks to human lives, has made them a focal point of study by the operations research community over the past few decades as evidenced by the substantial research effort targeted at modelling, designing, and analyzing such systems in different application settings.

An important aspect of the strategic or operational design of ESSs is the ability to accurately predict the equilibrium behavior and hence the expected performance of the system with a

[1] Published as Karimi, Akbar, Michel Gendreau, and Vedat Verter. "Performance Approximation of Emergency Service Systems with Priorities and Partial Backups." Transportation Science 52.5 (2018): 1235-1252.

given configuration. Discrete-event simulation can be used for this purpose and offers great flexibility in detailed modelling of the system at hand; however, its computational cost can be prohibitive in many practical applications. Fortunately, there exist analytical alternatives to simulation, with reduced computational overhead and reasonable accuracy.

Larson (1974) was the first to view ESS in urban settings as spatially distributed queues, and developed the hypercube queuing model as a descriptive tool for evaluating their performance. For an ESS with N servers, the model sets up a system of 2^N linear equations the solution of which gives the equilibrium probabilities of system states and makes it possible to compute a host of region-wide and server-specific performance measures, such as individual server workloads and the rates at which servers are dispatched to different demand locations. However, the exponential growth of the computational expense of the hypercube model with the fleet size may hinder its application to real life systems with large numbers of response units. This has motivated researchers to develop more tractable approximate alternatives. Larson (1975) was also the first to propose such an approximate procedure where an iterative algorithm is used to solve a system of N nonlinear equations for the individual server workloads (instead of the 2^N state probabilities of the exact model). The development is based on approximating the spatially distributed system with an M/M/N queue or an M/M/N/N loss system (depending on whether queues are allowed or not), and unlike the exact model, assumes identical mean service rates. Server dependency has also been approximately taken into account through a set of *correction factors*.

In this paper, we develop approximation algorithms for both loss and queuing ESSs with two main features. First, we allow multiple call types and priority processing of the waiting requests for the case where queues are allowed. Second, we relax the assumption of full backups, in which any server can travel to any demand location; instead, we allow arbitrary partial backups where each server can only be dispatched to calls from its own predetermined and priority-specific subset of demand locations. The service times in our models can depend on the customer and server locations and also on the priority level of the request for service. We believe the relaxation of these restricting assumptions represents important steps toward increased realism and applicability of such approximation algorithms.

The assumption of full or total backups, rarely reflects the complex dispatching protocols of the real systems and hence can substantially limit the realism of the descriptive modelling, and as shown by our experiments, may lead to unreliable performance approximations. The paramount importance of the quickness of service delivery in ESSs naturally imposes an upper limit on the customer proximity to the responding server beyond which the quality or outcome of the delivered service can be severely degraded. This is reflected in the notion of coverage

thresholds in location science literature and also in the behaviour of real systems where the distance between the customers and servers is often a crucial factor in making dispatching decisions. To the best of our knowledge, there exists no published general approximation method to deal with partial backups in server-to-customer systems or priorities in the waiting lines.

The original hypercube model and its approximation can handle a specific type of priority through the procedure of *layering*, in which demand zones are split into separate sub-nodes each generating a certain type of request and with their own list of preferred servers. However, this basic scheme only considers immediate dispatches and cannot be used to adequately represent the real life ESSs where the waiting customers are processed according to their priority levels. A number of studies mentioned in Section 4.2 present special applications of the exact hypercube model with partial backups, priorities in the queue, or both; however, our work is the first to provide the same extension in the approximate model.

The concurrent relaxation of the assumption of full backups and FCFS queue discipline in the approximate hypercube context, not only provides unique analytical and computational challenges, but also paves the way for more sophisticated and realistic modelling and analysis of ESSs for which these simplifying assumptions clearly do not hold. A typical EMS operation with a heterogeneous fleet, consisting of basic life support (BLS) vehicles and advanced life support (ALS) ambulances is an example of such a system where the ALS vehicles only cover higher priority service requests residing within a given distance while the BLS units can respond to any call within a certain (and possibly different) distance threshold. The ALS units might be given a priority over BLS units in responding to demand points covered by both vehicle types; alternatively, unit types and distances from the call location can be used together to determine optimal dispatching orders for the best service quality.

The paper is organized as follows: Section 4.2 gives a brief review of the relevant literature followed by the formulation of our approximation model in Section 4.3. The approximation algorithm is outlined in Section 4.4 and in Section 4.5, we describe our experimental setup and observations. Concluding remarks are given in Section 4.6 along with possible directions for future development.

4.2 Literature Review

We first review the notable assumptions underlying Larson's hypercube queuing model and its approximate alternative. Both procedures assume that demand for service is distributed over a given region broken down into a number of *geographic atoms* from which requests for

service arrive at known mean rates following a Poisson process and independently of any other atom. There are N response units that can be dispatched to calls coming from any atom (full backup assumption). The waiting locations of the servers (when they are idle) as well as the mean travel times between each waiting location and atom are assumed to be, at least probabilistically, known. In response to each incoming call for service, exactly one server is dispatched according to a fixed-preference dispatch procedure, where the first available server in a given atom-specific ordered list of all response units is dispatched. If there are no free units, the call is assumed to be added to a waiting line of infinite capacity, which in case of the exact model is depleted according to a queue discipline that does not depend on the expected service time or the call origin, for example First-Come, First-Served (FCFS), Last-Come-First-Served (LCFS), or random. The approximate model is more restricting in this regard as it only allows for an FCFS queue discipline. Negative exponential service times with known average values are assumed that include the travel time, the on-scene time, and the follow-up time. It is also assumed that the service time variation is mostly caused by on-scene and follow-up times rather than travel times. The exact hypercube model allows for server-specific mean service rates whereas the approximate procedure assumes identical mean service rates for all response units, although the contribution of travel times to the service rates can be accounted for through an iterative process called *mean service time calibration*.

Both the exact and approximate hypercube models have been employed, modified, and extended in various ways. Reviews of the relevant developments can be found in Larson (2013) and Galvao and Morabito (2008). We first briefly mention related applications or extensions of the exact model and then review the notable extensions upon the original approximate procedure. Atkinson et al. (2006) used a loss hypercube model to describe deployment of an Emergency Medical Services (EMS) along a highway where calls can only be serviced by the two neighboring ambulances. The resulting exact model with 2^N states was then approximately solved by heuristic algorithms to compute the probability of a call being lost as a result of both neighboring units being busy. This model was updated in Atkinson et al. (2008) to have each demand location covered by two primary and two secondary ambulances with different service times leading to an exact formulation with 3^N states which was again solved heuristically to approximate the loss probability. Iannoni and Morabito (2007) presents another example of an EMS on highways where only the first and second closest ambulances are allowed to respond to a call, and a request for service will be lost if both of these ambulances are busy at the arrival moment. A recent paper by de Souza et al. (2015) extends the exact hypercube model to account for priorities in the queue of waiting customers. This work has been recently extended in Rodrigues et al. (2017) to incorporate both partial backups and queue priorities in the exact hypercube model. These studies reveal that, despite its

attractive simplicity, the exact hypercube model can be computationally impractical even for small-sized systems, motivating researchers to develop general approximate alternatives as we review next.

Larson and Mcknew (1982) developed extensions of the original exact and approximate hypercube queuing models in a police patrol context and allowed three server states: patrol, busy with patrol-initiated activity, and busy with a call for service. They allowed the mean service times to depend on both servers and customer types in their exact model and only on customer types in their approximate version. Jarvis (1985) proposed an extended model for loss systems where the mean service times were allowed to depend on the server and customer locations. His formulation also allowed general service time distributions since these loss systems with distinguishable servers were shown to be relatively insensitive to the shape of the service time distribution beyond its mean. Birge and Pollock (1989) proposed a decomposition procedure in which the large linear system of equations corresponding to state probabilities of service systems is replaced by a set of non-linear equations and then solved iteratively. Their approach is approximate in that the servers are assumed independent. Goldberg and Szidarovszky (1991) presented detailed convergence results for two fixed-point iteration methods assuming independent servers; they considered loss systems with server and customer dependent service times and showed that the independence assumption can lead to biased performance measures especially in high system utilization.

As for the dispatch preference ties, which most often happen in case of co-located servers sharing a waiting station, there are two works to note: the work of Burwell et al. (1993) who extended the loss models of Jarvis (1985) and Larson (1975) to explicitly incorporate general dispatch ties through their *internal stacking method* and its slightly modified alternative; these algorithms were aimed to alleviate the storage and coding complexity issues involved in the so-called *stacking* procedure of accommodating dispatch ties in the hypercube models through software. Their formulation is based on individual servers and hence employs Larson correction factors to approximate server dependency. Budge et al. (2009), however, instead of general dispatch ties, extends the loss model of Jarvis (1985) to allow for multiple servers in each waiting station using a formulation based of station-specific average workload and dispatch rates instead of server-specific measures. They derived a modified set of correction factors to account for server cooperation and also gave a theoretical convergence guarantee for a special case of their procedure.

4.3 Formulation

In this section, we present our approximation model and the necessary formulation. The problem definition and preliminary assumptions are given first followed by the approximation of the distribution of the number of busy servers and the effects of cooperation and dependency among them. We then estimate the immediate and delayed dispatch rates of servers to demand locations.

We assume that the area of interest is partitioned into M distinct demand zones, each generating calls that can be classified into K priority levels. Requests for service with priority p from demand zone i arrive as Poisson streams with known average rates λ_{ip}. There are N mobile servers which respond to calls according to a fixed dispatch policy. For any given priority level p and demand zone i, there is a predetermined subset of servers which can be considered for dispatch. Binary variables b_{ijp} specify whether server j can respond to priority p calls from demand zone i ($b_{ijp} = 1$) or not ($b_{ijp} = 0$). If $b_{ijp} = 1$, we may say server j *p-covers* demand zone i, or equivalently, server j covers *sub-queue* (i, p). Letting $\mathcal{I}$, $\mathcal{J}$, and $\mathcal{P}$ respectively be the set of demand locations, servers, and priority classes, and denoting the number of servers covering sub-queue (i, p) by c_{ip} ($\leq N$) we have

$$c_{ip} = \sum_{j \in \mathcal{J}} b_{ijp} \qquad i \in \mathcal{I},\ p \in \mathcal{P}\,,$$

and the total covered demand λ given by

$$\lambda = \sum_{\substack{i \in \mathcal{I}, p \in \mathcal{P} \\ c_{ip} > 0}} \lambda_{ip}\,.$$

We assume that in response to an incoming call, the dispatcher considers the covering servers according to a fixed preference order and assigns the first available unit. The k-th preferred server to dispatch to a priority p call from demand zone i is denoted by $l(i, k, p)$ with $k = 1, ..., c_{ip}$. Auxiliary variables $r(i, j, p)$ are defined to denote the dispatch preference order of the j-th server when responding to a priority p call coming from demand zone i. By convention, we set $r(i, j, p) = 0$ if server j does not p-cover demand zone i, that is $b_{ijp} = 0$. Similarly, we set $l(i, k, p) = 0$ for $k > c_{ip}$. Thus, we will have $l(i, k, p) = j \Leftrightarrow r(i, j, p) = k$ for $k \leq c_{ip}$. As an example, consider a system with three demand zones ($M = 3$), three servers ($N = 3$), and two priority levels ($K = 2$). If the dispatch preference lists for the priority one calls coming from demand zones 1, 2, and 3 are respectively given as $\{2, 3, 1\}$, $\{3, 2\}$ and $\{1\}$, then we will have $l(1, 1, 1) = 2$, $l(1, 2, 1) = 3$, $l(1, 3, 1) = 1$, $l(2, 1, 1) = 3$, $l(2, 2, 1) = 2$,

$l(2,3,1) = 0$, $l(3,1,1) = 1$, $l(3,2,1) = 0$, and $l(3,3,1) = 0$, whereas $r(1,1,1) = 3$, $r(1,2,1) = 1$, $r(1,3,1) = 2$, $r(2,1,1) = 0$, $r(2,2,1) = 2$, $r(2,3,1) = 1$, $r(3,1,1) = 1$, $r(3,2,1) = 0$, and $r(3,3,1) = 0$.

We consider both loss and queuing systems; in the former case, calls arriving while all the covering servers are occupied will be lost, and in the latter case, they will join the queue of waiting customers. Because of partial backups, overtaking can happen in this queue when considered as one single line; however, the service discipline within each sub-queue will be strictly FCFS. Upon finishing its current job, a server will be dispatched to the longest waiting among the highest priority queued customers covered by that server, if one exists.

The rates at which priority p calls from demand location i are immediately assigned to a server or queued (lost in the loss system) are respectively denoted by λ^I_{ip} and λ^D_{ip}, with a^I_{ijp} and a^D_{ijp} the corresponding immediate and delayed dispatch rates to server j. We then have $\lambda_{ip} = \lambda^I_{ip} + \lambda^D_{ip}$, $\lambda^I_{ip} = \sum_{j\in\mathcal{J}} a^I_{ijp}$, and the total (delayed or immediate) dispatch rate of server j to priority p customers from location i as $a_{ijp} = a^I_{ijp} + a^D_{ijp}$. For the system with queues, we also have $\lambda^D_{ip} = \sum_{j\in\mathcal{J}} a^D_{ijp}$ and $\lambda_{ip} = \sum_{j\in\mathcal{J}} a_{ijp}$, whereas for loss systems, we interpret λ^D_{ip} as the mean rate at which calls are lost (rather than queued) and we set $a^D_{ijp} = 0$ since no customers will be dispatched from the queue.

4.3.1 Distribution of the Number of Busy Servers

The approximation will rely on the distribution of the number of busy servers $P_n = \Pr\{S_n\}$, $n = 0, \cdots, N$, with S_n the event of exactly n servers being busy. In the literature, a simplified queuing model with identical servers, that is M/M/N or M/M/N/N is usually employed to approximate the P_n of the spatially distributed system. Our experiments, however, show that these models are not adequate enough to approximate a system with partial backups operating at moderate to high workloads. To overcome this issue, we introduce *queuing (or loss) systems with partial service* defined below to act as a more accurate proxy to the EMS with partial backups.

Definition 1. *The queuing (loss) system with* partial service*, denoted by M/M/[N]/∞ (M/M/[N]/LOSS) has N identical servers with i.i.d exponential service times with rate μ and N customer classes arriving as Poisson streams with rates λ_c, $c = 1, 2, \cdots, N$, where each class c customer can receive service from exactly c servers randomly selected upon arrival, independently of other arrivals or properties of the system.*

We give the distribution of the number of busy servers of the queuing and loss systems with partial service in the following theorems which form the basis of our subsequent analysis.

Theorem 1. *The queuing system with partial service (M/M/[N]/∞) will reach steady state if* $N\mu > \sum_{c=1}^{N} \lambda_c$ *with the distribution of the number of busy servers given by*

$$P_n = P_0 \frac{N!}{(N-n)!n!} \prod_{j=0}^{n-1} \sum_{c=1}^{N} \lambda_c \sum_{\substack{h=max\{0,\\ c+j-N))\}}}^{min\{c-1,j\}} \frac{c!(N-c)!}{(N+h-c-j)!(j-h)!(c-h)!h!} \times \prod_{j\in\mathcal{J}} \frac{j!(N-j)!}{N!j\mu - j!\sum_{c=1}^{j} \lambda_c \frac{N-c}{j-c}} \quad n = 1, \cdots, N\,, \quad (4.1)$$

where

$$P_0 = \left[1 + \sum_{n=1}^{N} \frac{N!}{(N-n)!n!} \prod_{j=0}^{n-1} \sum_{c=1}^{N} \lambda_c \sum_{\substack{h=max\{0,\\ c+j-N))\}}}^{min\{c-1,j\}} \frac{c!(N-c)!}{(N+h-c-j)!(j-h)!(c-h)!h!} \times \prod_{j\in\mathcal{J}} \frac{j!(N-j)!}{N!j\mu - j!\sum_{c=1}^{j} \lambda_c \frac{N-c}{j-c}} \right]^{-1}. \quad (4.2)$$

Theorem 2. *The steady state distribution of the number of busy servers in the loss system with partial service is given by*

$$P_n = P_0 \prod_{k=1}^{n} \frac{\sum_{c=1}^{k} \lambda_c (1 - \frac{k!(N-c)!}{N!(k-c)!}) + \sum_{c=k+1}^{N} \lambda_c}{k\mu}, \quad n = 1, \cdots, N \quad (4.3)$$

with

$$P_0 = \left[1 + \sum_{n=1}^{N} \prod_{k=1}^{n} \frac{\sum_{c=1}^{k} \lambda_c (1 - \frac{k!(N-c)!}{N!(k-c)!}) + \sum_{c=k+1}^{N} \lambda_c}{k\mu} \right]^{-1}. \quad (4.4)$$

The Proofs of Theorems 1 and 2 will be given in Appendix A.

We approximate the distribution of the number of busy servers in our priority EMS with partial backups using an M/M/[N] model with input parameters

$$\lambda_c = \sum_{i\in\mathcal{I}} \sum_{\substack{p\in\mathcal{P}\\ c_{ip}=c}} \lambda_{ip}, \quad c = 1, \cdots, N\,, \quad (4.5)$$

and

$$\mu = \frac{\sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{P}} a_{ijp}}{\sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{P}} a_{ijp} t_{ijp}}\,, \quad (4.6)$$

whenever known or estimated values for the total dispatch rates a_{ijp} and individual service times t_{ijp} are available. In constructing λ_c in (4.5), we sum up all the demand (λ_{ip})

from any priority that is covered by exactly c servers. This ignoring of priorities represents another approximation because unlike the M/M/N queue, the state probabilities of the prioritized M/M/[N] can differ from the non-priority version; the difference, however, is negligible and does not affect the accuracy of our calculations in any practical way. To illustrate this, in Figure 4.1, we compare the state probabilities of a large set of randomly generated M/M/[N] queues obtained by simulation with those of the equivalent non-priority M/M/N and M/M/[N] models. Noting an order of magnitude difference in the vertical axes, it is clear that the non-priority M/M/[N] model approximates the priority M/M/[N] significantly better than the non-priority M/M/N model with average errors often around 0.5% and rarely exceeding 1% which we deem quite negligible. This allows us to confidently use the state probabilities obtained from the M/M/[N] model as an approximation to the distribution of the number of busy servers in the ESS. Finally, we point out that λ_c may be zero for some c; in particular, in a system where each demand location is assigned to exactly c' servers ($c' = N$ for full backups), we have $\lambda_{c'} = \lambda$ and $\lambda_c = 0$ for $c \neq c'$.

Similar to the original approximate hypercube model, the accuracy of P_n estimation decreases when the individual server workloads

$$\rho_j = \sum_{i \in \mathcal{I}} \sum_{p \in \mathcal{P}} a_{ijp} t_{ijp}, \quad j \in \mathcal{J}\,,$$

are not close enough to the average server workload

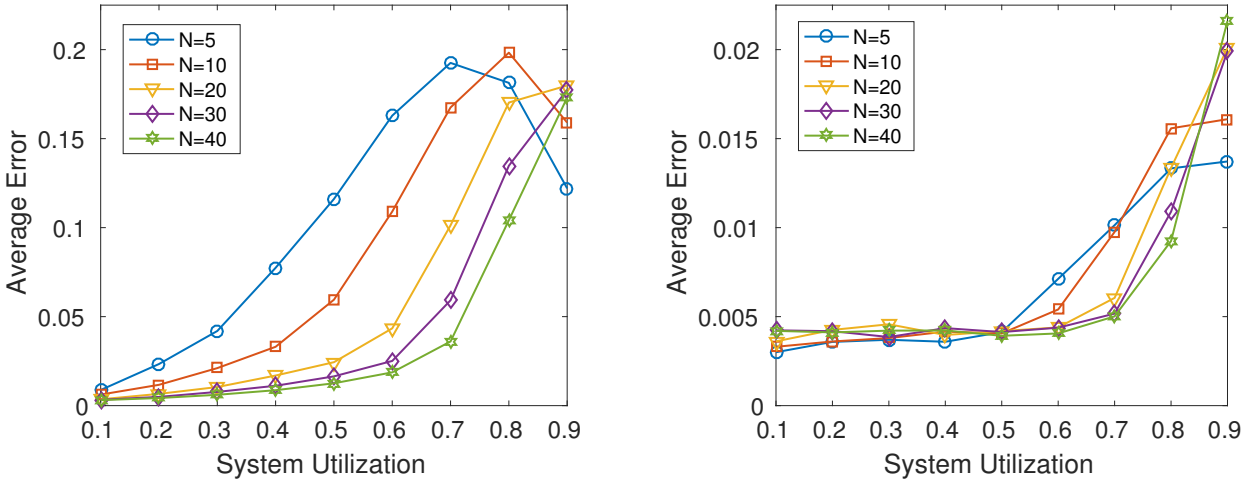
$$\rho = \frac{1}{N} \sum_{j \in \mathcal{J}} \rho_j = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{p \in \mathcal{P}} a_{ijp} t_{ijp}\,;$$

therefore, we propose an empirical approach detailed in Appendix A.3 to further improve the accuracy of the estimation by correcting the λ_c and μ_c values obtained from (4.5) and (4.6) based on a measure of discrepancy between server workloads.

4.3.2 Immediate Dispatches

In this section, we consider the calls who find at least one available covering server upon arrival, and hence do not experience any queuing delays. We are interested in the steady-state rates at which these immediately serviced calls are assigned to each of their covering servers.

To approximately account for the effects of server cooperation, we adopt an approach similar to Larson (1975) where P_n is now obtained from an M/M/[N] model. First, assume we randomly sample servers without replacement from the system operating at steady state



(a) M/M/N queue (b) M/M/[N] queue

Figure 4.1 Comparison of errors in approximating the state probabilities of a priority partial service queue by the corresponding non-priority version (M/M/[N]) and an M/M/N queue. Reported are the mean absolute errors for a system with three priority levels and different numbers of servers.

until we find an idle one. Letting B_j and F_j be the events of the j-th sampled server being respectively busy and free, we write the probability of the $j+1$-th sampled server being the first available one as

$$\Pr\{B_1 B_2 \cdots B_j F_{j+1}\} = \sum_{n=j}^{N} \Pr\{S_n\} \Pr\{B_1 B_2 \cdots B_j F_{j+1} | S_n\}\,;$$

moreover,

$$\Pr\{B_1 B_2 \cdots B_j F_{j+1} | S_n\} = \Pr\{F_{j+1} | B_1 B_2 \cdots B_j S_k\} \Pr\{B_j | B_1 B_2 \cdots B_{j-1} S_k\} \cdots \Pr\{B_1 | S_n\}\,.$$

The probabilities on the right-hand side can be expressed as

$$\Pr\{B_i | B_1 B_2 \cdots B_{i-1} S_n\} = [n-(i-1)]/[N-(i-1)]\,, \quad i = 1, 2, \cdots, n+1\,,$$

and

$$\Pr\{F_{j+1} | B_1 B_2 \cdots B_j S_n\} = (N-n)/(N-j)\,, \quad j = 0, 1, \cdots, n\,.$$

Combining the results above we have the desired probability

$$\Pr\{B_1B_2\cdots B_jF_{j+1}\} = \sum_{n=j}^{N-1} \frac{n}{N}\frac{n-1}{N-1}\cdots\frac{n-(j-1)}{N-(j-1)}\frac{N-n}{N-j}P_n\,, \quad j=0,1,\cdots,N-1\,,$$

or

$$\Pr\{B_1B_2\cdots B_jF_{j+1}\} = \sum_{n=j}^{N-1} \frac{N-n}{N-j}P_n\prod_{k=0}^{j-1}\frac{n-k}{N-k}\,, \quad j=0,1,\cdots,N-1\,.$$

Re-writing the above expression as

$$\Pr\{B_1B_2\cdots B_jF_{j+1}\} = Z(N,\mu,[\lambda_c],j)'\rho^j(1-\rho)\,,$$

we obtain our correction factors as

$$Z(N,\mu,[\lambda_c],j) = \sum_{n=j}^{N-1} \frac{n!(N-j-1)!(N-n)}{(n-j)!N!(1-\rho)\rho^j}P_n\,, \quad j=0,1,\cdots,N-1\,.$$

For the M/M/[N]/∞ system, we have $\rho=\lambda/(N\mu)$ with $\lambda=\sum_{c=1}^{N}\lambda_c$ and thus the Z factors will be given as

$$Z(N,\mu,[\lambda_c],j) = \sum_{n=j}^{N-1} \frac{n!(N-j-1)!(N-n)}{(n-j)!N!(\frac{\lambda}{N\mu})^j(1-\frac{\lambda}{N\mu})}P_n\,, \quad j=0,1,\cdots,N-1\,, \tag{4.7}$$

with P_n given by (4.1). Class c customers arriving to the loss system in state S_n will be lost at an average rate of

$$q_c(n) = \frac{\binom{n}{c}}{\binom{N}{c}} = \frac{n!(N-c)!}{N!(n-c)!}\,, \tag{4.8}$$

resulting in an average server workload of

$$\rho = \frac{\sum_{c=1}^{N}\lambda_c(1-\sum_{n=0}^{N}q_c(n)P_n)}{N\mu}\,,$$

leading to the loss system Z factors of the form

$$Z(N,\mu,[\lambda_c],j) = \left[\frac{1}{N\mu}\sum_{c=1}^{N}\lambda_c\left(1-\sum_{n=0}^{N}\frac{n!(N-c)!}{N!(n-c)!}P_n\right)\right]^j\left[1-\frac{1}{N\mu}\sum_{c=1}^{N}\lambda_c\left(1-\sum_{n=0}^{N}\frac{n!(N-c)!}{N!(n-c)!}P_n\right)\right]$$
$$\times\sum_{n=j}^{N-1}\frac{n!(N-j-1)!(N-n)}{(n-j)!N!}P_n\,, \quad j=0,1,\cdots,N-1\,,$$

with P_n given by (4.3).

Similar to the Q factors derived by Larson (1975) for M/M/N queues, our Z factors represent the extent to which the term $\rho^j(1-\rho)$, which assumes independent servers, should be *corrected* to reflect the probability of sampling exactly j servers before finding an available server in an M/M/[N] queue operating at steady state.

Values of P_n and $Z(\cdot, j)$ for the queue and loss versions of an M/M/[10] system with different $[\lambda_c]$ but identical total demand are plotted in Figure 4.2, which clearly shows the strong dependence of both quantities on the distribution of total demand into the customer classes. We note that scenario A in this example is equivalent to a regular M/M/N queue and therefore the corresponding Z factors are identical to the Q factors of Larson (1975). It is particularly interesting to observe how shifting the concentration of the demand intensity towards the more openly received customer classes (higher c) increases the effects of server cooperation; this is highlighted by the most aggressive correction factors associated with scenario A which represents the extreme case with the highest server cooperation. Case B is the other extreme where each arrival is covered by a single server and hence we have zero cooperation and perfect independence reflected in correction factors equal to unity. We also need an expression for λ_{ip}^D the rate at which priority p calls from location i get queued or lost. Denoting the event of having no free servers to cover a priority p call from demand location i by B_{ip}, we have $B_{ip} \equiv \bigcap_{j=1}^{c_{ip}} B_{l(i,p,j)}$. Conditioning on the system state S_n we have $\lambda_{ip}^D = \lambda_{ip} \Pr\{B_{ip}\}$ with

$$\Pr\{B_{ip}\} = \lambda_{ip} \sum_{n=0}^{N} P_n \Pr\{B_{ip} \mid S_n\}\,.$$

We see that $\Pr\{B_{ip} \mid S_n\} = 0$ for $n < c_{ip}$ and

$$\Pr\{B_{ip} \mid S_n\} = \frac{\binom{N - c_{ip}}{n - c_{ip}}}{\binom{N}{n}} = \frac{(N - c_{ip})!n!}{(n - c_{ip})!N!}\,,$$

for $n \geq c_{ip}$. Thus

$$\lambda_{ip}^D = \lambda_{ip} \sum_{n=c_{ip}}^{N} P_n \frac{(N - c_{ip})!n!}{(n - c_{ip})!N!}\,. \tag{4.9}$$

Given a set of steady-state server workloads ρ_j, $j \in \mathcal{J}$, one can obtain a set of tentative values for the immediate dispatch rates a_{ijp}^I as

$$a_{ijp}^I = b_{ijp}\lambda_{ip} Z(N, \mu, [\lambda_c], r(i,j,p) - 1)(1 - \rho_j) \prod_{k=1}^{r(i,j,p)-1} \rho_{l(i,k,p)}\,, \tag{4.10}$$

which can then be normalized using any of the normalization schemes discussed in Section 4.4 to satisfy the balance equation

$$\sum_{j \in \mathcal{J}} a^I_{ijp} = \lambda_{ip} - \lambda^D_{ip} \,. \tag{4.11}$$

We can also approximate a^I_{ijp} and λ^D_{ip} using an alternative two-pass scheme where we make a better use of the individual server workloads ρ_j; in the first pass, values for the dispatch rates assuming full backups denoted by $\hat{a}^I_{ijp}$ are calculated as

$$\hat{a}^I_{ijp} = \lambda_{ip} Z(N, \mu, [\lambda_c], r(i,j,p) - 1)(1 - \rho_j) \prod_{k=1}^{r(i,j,p)-1} \rho_{l(i,k,p)} \,, \tag{4.12}$$

and normalized according to the balance equation

$$\sum_{j \in \mathcal{J}} \hat{a}^I_{ijp} = \lambda_{ip} \left(1 - P_N\right) \,, \tag{4.13}$$

where P_N is given by (4.1) or (4.3). In the second pass, the a^I_{ijp} values are obtained by recognizing that the normalized full-backup dispatches to non-covering servers would indeed be queued or lost in the actual partial-backup system; that is

$$a^I_{ijp} = b_{ijp} \hat{a}^I_{ijp} \,. \tag{4.14}$$

Finally, the rate of calls being queued (or lost) is given by

$$\lambda^D_{ip} = \lambda_{ip} P_N + \sum_{j \in \mathcal{J}} \left(1 - b_{ijp}\right) \hat{a}^I_{ijp} \,, \tag{4.15}$$

where the first part of the right-hand side represents the customers who arrive when all servers of the hypothetical full-backup system are busy and the second term adds the contribution of partial backups by summing up the hypothetical dispatches to the non-covering servers. Our numerical tests show that, on average, using the two-pass scheme defined by (4.12), (4.13), (4.14) and (4.15) instead of (4.11), (4.9) and (4.10), indeed leads to around 50% reduction in server workload and immediate dispatch estimation errors with an even higher 80% improvement in the estimation of delayed dispatch rates and waiting times discussed in the next section. We hence use this method in our experiments.

4.3.3 Delayed dispatches

The delayed dispatches in a system with full backups will be uniformly distributed among the N servers when service times are identical; that is $a^D_{ijp} = \lambda^D_{ijp}/N$ if $t_{ijp} = 1/\mu$, $i \in \mathcal{I}$, $j \in \mathcal{J}$, $p \in \mathcal{P}$. With location or priority dependent service times, some unevenness will be introduced in this distribution proportional in magnitude to the extent of differences in t_{ijp} values. This, however, will generally be of secondary importance compared to the asymmetry of the case with partial backups where servers covering a given sub-queue will have to finish work on different subsets of waiting customers before being able to receive the next delayed customer from that sub-queue. In this section, we are interested in quantifying this asymmetry and deriving approximate expressions for the steady-state dispatch rates of waiting customers and the average delays incurred in the queue.

Let B^D_{jp} be the event of server j being busy with a delayed priority p call and ρ^D_{ijp} the probability of server j being busy with a delayed call with priority p *or higher* (that is with priority $p' \leq p$) given that all servers p-covering the demand location i are also busy (the event B_{ip}); that is $\rho^D_{ijp} = \Pr\left\{\cup^p_{p'=1} B^D_{jp'} \mid B_{ip}\right\}$. Although obtaining exact expressions for ρ^D_{ijp} for the general case is mathematically intractable, we propose the following approximation

$$\rho^D_{ijp} = \sum_{p'=1}^{p} \sum_{i' \in \mathcal{I}} \kappa^{ip}_{i'p'} \lambda_{ip} \bar{a}^D_{i'jp'} t_{i'jp'} \,, \tag{4.16}$$

with $\bar{a}^D_{i'jp'} = a^D_{i'jp'}/\lambda^D_{ip}$ and the *coupling factor* $\kappa^{ip}_{i'p'}$ defined as the probability of all servers p'-covering demand location i' being simultaneously busy given that all servers p-covering demand location i are busy; or

$$\kappa^{ip}_{i'p'} = \Pr\{B_{i'p'} \mid B_{ip}\}\,, \quad i, i' \in \mathcal{I}, p, p' \in \mathcal{P}\,.$$

The coupling factors can be approximated assuming either indistinguishable or independent servers. With the assumption of indistinguishable dependent servers, we readily observe that

$$\kappa^{ip}_{i'p'} = \Pr\{B_{i'p'} \mid B_{ip}\} = \frac{\Pr\{B_{i'p'} \cap B_{ip}\}}{\Pr\{B_{ip}\}}\,,$$

and also that, conditioning on the system state S_n

$$\Pr\{B_{i'p'} \cap B_{ip}\} = \sum_{n=0}^{N} P_n \Pr\{B_{i'p'} \cap B_{ip} \mid S_n\}\,.$$

Defining $h^{ip}_{i'p'}$ as the number of servers that either p-cover demand location i or p'-cover

demand location i', that is $h^{ip}_{i'p'} = \sum_{j\in\mathcal{J}} 1-(1-b_{ijp})(1-b_{i'jp'})$, we have

$$\Pr\{B_{i'p'} \cap B_{ip} \mid S_n\} = \binom{N-h^{ip}_{i'p'}}{n-h^{ip}_{i'p'}}\binom{N}{n}^{-1},$$

if $h^{ip}_{i'p'} \leq n$ and $\Pr\{B_{i'p'} \cap B_{ip} \mid S_n\} = 0$ otherwise. The coupling factors are then obtained as

$$\kappa^{ip}_{i'p'} = \frac{\sum_{n=h^{ip}_{i'p'}}^{N} P_n (N-n)!^{-1} \prod_{k=n+1}^{N} (k-h^{ip}_{i'p'})}{\sum_{n=c_{ip}}^{N} P_n (N-n)!^{-1} \prod_{k=n+1}^{N} (k-c_{ip})}. \tag{4.17}$$

Equation (4.17) implies indistinguishable servers assumption as no individual server workloads are used. However, server dependency is taken into account through the conditioning on the system state S_n.

We can also assume independent distinguishable servers and approximate $\kappa^{ip}_{i'p'}$ as

$$\kappa^{ip}_{i'p'} = \prod_{\substack{j\in\mathcal{J} \\ b_{i'jp'}=1,\, b_{ijp}=0}} \rho_j \,. \tag{4.18}$$

As discussed in Section B.4 of the electronic companion, this approach can be a viable alternative to (4.17) and depending on the problem size and the implementation, may result in better or worse performance in terms of accuracy or efficiency.

Finally, we define *residual service rates* μ^D_{ijp} as the service rate delivered by server j to the end of the waiting line of priority p customers from demand location i and write the approximating expression

$$\mu^D_{ijp} = b_{ijp}\mu_j \left(1-\rho^D_{ijp}\right)\left(1-\rho^D_{ij(p-1)}\right), \tag{4.19}$$

with μ_j the average service rate of server j defined as

$$\mu_j = \frac{\sum_{p\in\mathcal{P}}\sum_{i\in\mathcal{I}} a_{ijp}}{\sum_{p\in\mathcal{P}}\sum_{i\in\mathcal{I}} a_{ijp}t_{ijp}}.$$

With μ^D_{ijp} known, we compute the normalized delayed dispatch rates as

$$\bar{a}^D_{ijp} = \frac{\mu^D_{ijp}}{\sum_{k=1}^{N}\mu^D_{ikp}} \tag{4.20}$$

and the average time spent in queue by the delayed customers from sub-queue (i, p) as

$$w_{ip}^D = \left(\sum_{j \in \mathcal{J}} \mu_{ijp}^D \right)^{-1} . \tag{4.21}$$

The delayed dispatch rates and the waiting times of *all arrivals* from sub-queue (i, p) then follow from

$$a_{ijp}^D = \lambda_{ip}^D \bar{a}_{ijp}^D , \tag{4.22}$$

and

$$w_{ip} = \frac{\lambda_{ip}^D}{\lambda_{ip}} w_{ip}^D . \tag{4.23}$$

Expression (4.19) has an intuitive interpretation. Imagine a priority p customer from demand location i enters the queue. A covering server j will finish its current task (possibly with an immediately dispatched customer) at rate μ_j; with probability $1 - \rho_{ijp}^D$ no other customer will be waiting before this newly arrived one in gaining access to server j; and with probability $1 - \rho_{ij(p-1)}^D$ no higher priority customer covered by server j will push this customer back while he is waiting. It can be verified that for the special case with full backups ($b_{ijp} = 1$, $i \in \mathcal{I}$, $j \in \mathcal{J}$, $p \in \mathcal{P}$), all coupling factors $\kappa_{i'p'}^{ip}$ will be equal to unity. If we also have identical service times (that is $t_{ijp} = 1/\mu$ for all $i \in \mathcal{I}$, $j \in \mathcal{J}$, and $p \in \mathcal{P}$) which are also negative exponentially distributed, then relations (4.16), (4.19), (4.22) and (4.21) will hold exactly and yield the well-known expression for the waiting time of the prioritized M/M/N/∞ queue. In the general case, however, the period between successive servings of queued customers from a given sub-queue by a specific server will depend on the detailed state of the system and will not be independently and exponentially distributed as suggested by these approximations, which, except for w_{ip}^D, are shown in Section 4.5 to provide good results even with non-exponential service times. The approximated w_{ip}^D from (4.21) seems to be sensitive to the service time distribution, which can perhaps be attributed to the fact that unlike the $\bar{a}_{ijp}^D$ from (4.20), which only depend on the ratios of the approximated μ_{ijp}^D values across the fleet, the w_{ip}^D given by (4.21) depend on the actual μ_{ijp}^D values thus making them more sensitive to the validity and accuracy of the approximation.

We now have all the ingredients to derive an iterative algorithm to approximate the steady-state values of the desired performance measures, namely the average server workloads, immediate and delayed dispatch rates, and waiting times.

4.4 Algorithm

In this section, we provide the layout of our approximation algorithm. All the assignments are assumed to be carried out for $i, i' \in I$, $j \in J$, and $p, p' \in P$. Here are the steps to follow:

1. Initialization
   (a) Set the iteration counter: $g = 0$;
   (b) Initialize the problem definition parameters N, M, K, λ_{ip}, t_{ijp}, $l(i,j,p)$, $r(i,j,p)$, c_{ip}, b_{ijp}, and $h^{ip}_{i'p'} \leq n$;
   (c) Starting from an empty system, initialize the dispatch rate and server workload variables to zero: $\rho_j^{(g)} = 0$, $a_{ijp}^{(g)} = 0$, and $\rho^{(g)} = 0$;
2. Pre-processing
   (a) Compute the state probabilities of the underlying M/M/[N] queue, $P_0, P_1, \cdots, P_N$ from (4.1) and (4.2) for the system with queues and from (4.3) and (4.4) for the loss system;
   (b) Compute Z correction factors from (4.7);
3. Immediate Dispatches
   (a) Compute the full-backup immediate dispatch rates $\hat{a}^I_{ijp}$ from (4.12);
   (b) Normalize $\hat{a}^I_{ijp}$ values obtained above so that $\sum_{j\in\mathcal{J}} \hat{a}^I_{ijp} = \lambda_{ip}(1 - P_N)$;
   (c) Update the immediate dispatch rates as $a^I_{ijp} = b_{ijp}\hat{a}^I_{ijp}$;
   (d) Compute the rates of delayed or lost calls from (4.15);
4. Delayed Dispatches
   (a) Compute the coupling factors $\kappa^{ip}_{i'p'}$ either from (4.17) or (4.18);
   (b) Compute updated values for ρ^D_{ijp} from (4.16);
   (c) Compute updated values for the residual service rates μ^D_{ijp} from (4.19);
   (d) Obtain the delayed dispatch rates as in (4.22): $a^D_{ijp} = \lambda^D_{ip}\mu^D_{ijp} / \sum_{k=1}^N \mu^D_{ikp}$;
   (e) Compute the expected waiting times from (4.23): $w_{ip} = \lambda^D_{ip}/\lambda_{ip} \sum_{j=1}^N \mu^D_{ijp}$;
5. Global Update and Termination

(a) Define and set auxiliary variables V_j as

$$V_j = \frac{\sum_{i=1}^{N} \sum_{p \in \mathcal{P}} a_{ijp} t_{ijp}}{1 - \rho_j^{(g)}} ;$$

(b) Update server workloads as $\rho_j^{(g+1)} = V_j/(1 + V_j)$. This update rule ensures that server workloads remain between 0 and 1 and helps with the stability of the algorithm. For details see Larson (1975), Jarvis (1985), Budge et al. (2009) and Goldberg and Szidarovszky (1991).

(c) Terminate the algorithm if a given convergence condition is satisfied such as

$$\max_{j \in \mathcal{J}} |\rho_j^{(g+1)} - \rho_j^{(g)}| \leq \epsilon \, ,$$

with ϵ a predefined convergence threshold; otherwise, set $g \leftarrow g + 1$ and start a new iteration from Step 2.

The normalization procedure in step 3b can be carried out in different ways. The simplest method is to scale all $\hat{a}_{ijp}^I$ for $j \in \mathcal{J}$ and a given sub-queue (i, p) by a single factor so that the balance equation is satisfied. Alternatively, we can retain the element corresponding to the primary (the most preferred) server (that is $\hat{a}_{i1p}^I$) and scale all the remaining values. In our experience this method slightly improves the approximation and is thus employed in our numerical experiments. We found more complicated approaches, such as the third normalization method in Larson (1975), to be generally too computationally costly for the systems and scales we consider here.

Finally, although the algorithm presented above starts from an empty system, based on our experiments, the method performs equally well with arbitrary initial conditions. An illustration of the algorithm is given next.

We apply the presented algorithm to a simple example consisting of five demand zones ($M = 5$) and three servers ($N = 3$) located as shown in Figure 4.3.a. We assume two priority levels ($K = 2$) with arrival rates and dispatch preference orders given as

$$[\lambda_{ip}] = \begin{bmatrix} 1.0 & 1.0 \\ 0.225 & 0.075 \\ 0.075 & 0.1 \\ 0.05 & 0.2 \\ 0.2 & 0.05 \end{bmatrix}, \; [b_{ij1}] = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \; [b_{ij2}] = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix},$$

$$[r_{ij1}] = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & 3 \\ 2 & 3 & 1 \\ 3 & 2 & 1 \\ 1 & 2 & 3 \end{bmatrix}, \; [r_{ij2}] = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \\ 3 & 2 & 1 \end{bmatrix},$$

with the matrix $[c_{ip}]$ implied from $[b_{ijp}]$ as

$$[c_{ip}] = \begin{bmatrix} 1 & 2 \\ 2 & 2 \\ 2 & 2 \\ 2 & 1 \\ 3 & 2 \end{bmatrix}.$$

We assume exponentially distributed service times with three components: travel time from the server location to the scene, on-scene time, and follow-up time. We set on-scene and follow-up times respectively to 20 and 30 minutes regardless of the call priority, and use the procedure detailed in Section 4.5 to approximate the mean travel times between each server and customer location and for each priority. Euclidean norms were used to measure travel distances. The resulting mean service times are

$$[t^{ser}_{ij1}] = \begin{bmatrix} 58.0 & 61.1 & 60.8 \\ 61.4 & 60.2 & 62.2 \\ 60.1 & 60.5 & 58.2 \\ 62.2 & 59.8 & 60.4 \\ 56.9 & 56.2 & 56.4 \end{bmatrix}, \quad [t^{ser}_{ij2}] = \begin{bmatrix} 61.6 & 66.1 & 65.7 \\ 66.6 & 64.8 & 67.8 \\ 64.7 & 65.3 & 61.9 \\ 67.8 & 64.3 & 65.2 \\ 59.9 & 59.0 & 59.3 \end{bmatrix}.$$

The plots in Figure 4.3.b show how server workloads converge to their final values as the algorithm progresses when applied to the loss and queuing versions of this example problem. As typically observed, the algorithm takes fewer iterations in the case of a loss system.

The outputs of the model when applied to the example problem as a queuing or loss system are compared with the corresponding values obtained from the simulation model in Tables 4.1, 4.2, 4.3 and 4.4. We observe accurate approximations to all performance measures in each priority and for both queuing and loss systems. We note the significant asymmetry in the distribution of immediate and delayed dispatches. This is caused by differing dispatch preference orders and partial backups in case of the immediate dispatches, and for the delayed dispatches, can be attributed mainly to partial backups and, probably to a much less degree, to varying service times.

Table 4.1 Comparison of server workloads estimated by the model and simulation.

	Loss			Queue		
	Server 1	Server 2	Server 3	Server 1	Server 2	Server 3
Simulation	0.2589	0.4513	0.2486	0.3323	0.5453	0.2922
Model	0.2574	0.4512	0.2482	0.3309	0.5467	0.2927

Table 4.2 Comparison of immediate dispatch rates estimated by the model and simulation (in parentheses).

	Priority 1			Priority 2		
Demand Zone	Server 1	Server 2	Server 3	Server 1	Server 2	Server 3
1	1 (1)	—	—	.8462 (.8543)	.1538 (.1457)	—
2	.4187 (.4179)	.5813 (.5821)	—	—	.5674 (.5634)	.4326 (.4366)
3	.1797 (.1892)	—	.8203 (.8108)	.1797 (.1891)	—	.8203 (.8109)
4	—	.5674 (.5659)	.4326 (.4341)	—	1 (1)	—
5	.1105 (.1097)	.5047 (.5018)	.3848 (.3885)	—	.5674 (.5634)	.4326 (.4366)

4.5 Numerical Experiments

We have conducted comprehensive numerical experiments to assess the efficiency of the proposed approximation algorithm and also to answer a few questions that may arise while applying the model to practical cases. A model representing the downtown area of the city of Montreal, Canada and its main EMS provider, has been developed and employed in our tests as a realistic example of a typical ESS in an urban setting. Because of the unavailability of real EMS history data, we used the census data provided by Statistics Canada (2016b,a) which gives the population densities of arbitrarily shaped subdivisions called *dissemination areas*. Based on these data we constructed a model comprising 447 uniformly distributed square shaped demand zones with intensities at different priority levels determined by the

Table 4.3 Comparison of delayed dispatch rates estimated by the model and simulation (in parentheses).

	Priority 1			Priority 2		
Demand Zone	Server 1	Server 2	Server 3	Server 1	Server 2	Server 3
1	1 (1)	—	—	.5297 (.5436)	.4703 (.4564)	—
2	.4800 (.4821)	.5200 (.5179)	—	—	.4090 (.4109)	.5910 (.5891)
3	.4667 (.4681)	—	.5333 (.5319)	.4331 (.4460)	—	.5669 (.5540)
4	—	0.4862 (.4877)	.5138 (.5123)	—	1 (1)	—
5	.3032 (.3043)	.3317 (.3297)	.3652 (.3660)	—	.4090 (.4116)	.5910 (.5884)

Table 4.4 Comparison of average waiting times (in minutes) and fractions of calls queued estimated by the model and simulation (in parentheses) for the system with queues.

	Waiting Times		Fraction of Calls Queued	
Demand Zone	Priority 1	Priority 2	Priority 1	Priority 2
1	23.5 (22.8)	12.9 (12.4)	.3309 (.3323)	.2093 (.2192)
2	8.3 (8.0)	9.6 (9.0)	.2201 (.2182)	.2010 (.1939)
3	4.8 (4.5)	6.3 (5.9)	.1378 (.1281)	.1378 (.1271)
4	6.7 (6.4)	54.6 (49.6)	.2010 (.1951)	.5467 (.5446)
5	2.6 (2.4)	9.6 (8.9)	.1017 (.0950)	.2010 (.1929)

population densities of the overlapping dissemination areas and the yearly per capita number of EMS requests reported by Urgences-santé the main EMS provider in the region. Irrespective of location, calls can be urgent (priority 1), less urgent (priority 2) or non-urgent (priority 3) with the respective ratios 0.44, 0.38, and 0.18. The total arrival rate from the region is 9.11 per hour. Conversion to a regular grid were motivated by possible future applications of the model in equitable EMS deployments where geographical locations should be equally represented irrespective of call volumes they generate. Figure 4.4 shows the distribution of the demand intensity and locations of the hospitals serving the area.

We assume the service times to include six components: (1) Dispatch Time: from the moment the dispatcher receives the call until a decision is made to assign an ambulance to the call or put it in the waiting queue; (2) Chute Time: from the moment the ambulance crew is notified of the dispatch until they become en route to the scene; (3) Travel Time: ambulance traveling time from the current location to the scene; (4) Scene Time: time spent on scene; (5) Transport Time: traveling time from the scene to the care center; (6) Turnaround Time: from the arrival of the ambulance to the care center until it is back in service which typically consists of transferring the patient to the care center and possibly ambulance clean-up or replenishing any depleted resources. Based on data we had on ambulance operations in Montreal and other Canadian metropolitan areas we estimated the values of the non-traveling components of the service time as in Table 4.5. For the travel and transport times, we use the log-normally distributed stochastic model in Budge et al. (2010) to approximate the random time t to travel a given distance d, that is

$$t = m(d)e^{c(d)z}\,, \tag{4.24}$$

$$m(d) = \begin{cases} 2\sqrt{d/a} & \text{if } d \leq 2d_c \\ v_c/a + d/v_c & \text{if } d > 2d_c \end{cases}\,, \tag{4.25}$$

Table 4.5 Components of the service time (in minutes).

	Priority 1	Priority 2	Priority 3
Dispatch Time	2	3	3
Chute Time	0.5	0.5	0.5
Scene Time (Patient Transported)	19	19	19
Scene Time (Patient not Transported)	22	22	22
Turnaround Time	40	50	50

$$c(d) = \frac{\sqrt{b_0(b_2+1) + b_1(b_2+1)m(d) + b_2 m(d)^2}}{m(d)}\,, \tag{4.26}$$

where $m(d)$ and $c(d)$ are respectively the median and coefficient of variation (CV) of the travel time distribution with z a random variable with a standard normal distribution. Here, b_0 represents variation at the beginning and end of a trip independent of the travel distance, such as inaccuracy in distance measurement; b_1 represents short-term variation of speed during a trip; and b_2 represents trip to trip variation of travel time caused by factors like traffic or weather conditions. The values we selected for the parameters of the model are given in Table 4.6 for each priority level. For the urgent calls, we used the values reported in Budge et al. (2010) for the Calgary EMS, and for the non-urgent calls, we simply modified some of the parameters to reflect the lower cruising speeds and accelerations and higher variability associated with driving in normal mode and abiding by all traffic laws. Values of d were computed as Manhattan norms tilted at 45 degrees to match the street layout in the region. Finally, the corresponding mean travel times needed by the approximation model can be obtained as

$$\bar{t} = m(d)e^{c(d)^2/2}\,. \tag{4.27}$$

Table 4.6 Values of the travel time model parameters.

Parameter	Priority 1	Priority 2	Priority 3
a (acceleration, km/h/min)	41	25	25
v_c (cruising speed, km/hr)	100	70	70
b_0	0.336	0.336	0.336
b_1	0.000058	0.000116	0.000116
b_2	0.0388	0.0776	0.0776

We have conducted tests using six different coverage threshold scenarios given in Table 4.7 with C1 representing the full backup scenario. For a given fleet size (N), we generated 100

Table 4.7 Coverage threshold scenarios considered in the experiments. In each scenario, the maximum coverage threshold for each priority level is given in kilometers.

Scenario	Priority1	Priority 2	Priority 3
C1*	∞	∞	∞
C2	6	6	6
C3	4	4	4
C4	6	4	2
C5	2	4	6
C6	2	2	2

*Full backups

random ambulance location sets where no ambulance is located farther than 500 meters from its closest demand zone (to exclude unrealistic deployment patterns). We considered a given test location set in both queuing and loss situations (designated by Queue and Loss in the tables) and in each case compared the outputs of the approximation model with those of a discrete event simulation which we developed to benchmark the performance of our algorithm. Both the approximation and simulation models were coded in MATLAB and shared the same input parameters. Various service time distributions were tested in the simulation model as discussed in Section B.1. To minimize the variability of the simulation outputs, tests were run for 5000000 simulation hours with 100000 hours of warm-up. Prior to each simulation run, the demand not covered by any server, were removed by settig every λ_{ip} for which $c_{ip} = 0$ to zero.

Denoting simulation outputs by hatted symbols, we use $E_\rho = \sum_{j=1}^{N} |\rho_j - \hat{\rho}_j|/\hat{\rho}_j$ as the measure of error in the estimation of server workloads for a given test problem. For total, immediate, and delayed dispatch rates, we take the average errors per dispatch given by $E_a = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{p\in\mathcal{P}} |a_{ijp} - \hat{a}_{ijp}|/\lambda$, $E_a^I = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{p\in\mathcal{P}} |a_{ijp}^I - \hat{a}_{ijp}^I|/\lambda$ and $E_a^D = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{p\in\mathcal{P}} |a_{ijp}^D - \hat{a}_{ijp}^D|/\lambda$ as error measures, respectively. We capture waiting time estimation errors by $E_w = \sum_{i\in\mathcal{I}}\sum_{p\in\mathcal{P}} |w_{ip} - \hat{w}_{ip}|/\lambda$ and also report the average waiting times from the simulation $\hat{w} = \sum_{i\in\mathcal{I}}\sum_{p\in\mathcal{P}} \hat{w}_{ip}/\lambda$ as a reference for E_w. We emphasize that the total covered demand λ will be different for each test problem in scenarios with finite coverage threshold. The measures computed for a specific priority p are denoted by E_{a^p}, $E_{a^p}^I$, $E_{a^p}^D$, E_{w^p} and $\hat{w}_p$ and of course normalized with the total priority p covered demand. Finally, we report in the subsequent tables the error measures averaged over all test problems in percentage format; that is, for instance $\bar{E}_\rho = (1/T)\sum_{m=1}^{T} E_\rho^m/100\%$ where E_ρ^m is E_ρ corresponding to the m-th out of the T number of test cases (here $T = 100$).

In this section, we test the validity of the full-backup assumption, assess the general effec-

tiveness of the algorithm and give a summary of the rest of our computational experiments presented in the electronic companion. We have repeated these tests for N=20, 30, 40, and 50 but only report the results for $N = 20$ for ease of exposition. For each value of N we test varying system workloads by scaling the total demand accordingly (See Section B.3). The error margins and other observations for the cases with larger fleet sizes were closely matching the results we report for $N = 20$. Most importantly, however, the computational expense of the approximation method was observed to grow almost linearly with N. This is important as it indicates that the procedure may be applied to even larger-scale problems without getting computationally prohibitive.

4.5.1 Validity of the Full-backup Assumption

This paper is focused on the approximation of ESSs with partial backups. It is then naturally interesting to see whether an approximation based on a full-backup assumption will be adequate or not in practical settings. In Table 4.8 we compare the performance measures obtained from the simulation model for different coverage threshold scenarios with the values predicted by the approximation algorithm assuming full-backups (that is the coverage scenario C1). It is readily observed that as the coverage thresholds become more stringent, the approximation model with a full-backup assumption becomes increasingly incapable of predicting any of the performance measures with reasonable accuracy, with estimation errors up to 100% and 90% for the server workload and dispatch rates. We note that the full-backup assumption yields an average waiting time of nearly zero resulting in 100% waiting time estimation errors for all coverage scenarios except C1. We also observed in our experiments that the discrepancy between the simulation and the full backup model, not surprisingly, grows with increasing system workloads and decreasing fleet sizes.

As mentioned earlier, in systems with partial backups, the values of performance measures corresponding to each server can be largely different and the assumption of full backups will not reveal these imbalances. However, there is another major drawback associated with assuming full backups in applications where the demand that is not covered by any server is considered lost. In these cases, each candidate positioning of the response units will lead to a different total covered demand, λ, and thus a different average server workload. The full backup assumption will also fail to reveal these effects. Therefore, we can conclude that assuming full backups may result in highly inaccurate approximations in many practical applications, especially with fairly moderate to high congestion levels and dispatch protocols not compatible with this assumption.

Table 4.8 Estimation errors assuming full-backups*.

	Loss						Queue					
	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6
$\bar{E}_\rho$	1.68	4.67	17.98	22.64	38.53	111.30	1.69	2.08	9.54	10.60	20.09	78.84
$\bar{E}_a$	5.43	8.95	23.87	26.31	40.10	65.68	5.42	11.97	31.75	31.21	50.15	87.35
$\bar{E}_w$	—	—	—	—	—	—	0.00	0.79	6.70	6.24	13.20	44.97

* $\bar{E}_\rho$ and $\bar{E}_a$ in percentages, $\bar{E}_w$ in minutes

4.5.2 Accuracy of the Approximation

Having established the importance of relaxing the full-backup assumption, we now evaluate the predictive accuracy of the algorithm. The average total and priority-specific estimation errors in each performance measure for different coverage scenarios and queue disciplines are given in Tables 4.9, 4.10, and 4.11. The error margins in prediction of server workloads and dispatch rates are capped at around 1.7% and 5.5% respectively, with considerably lower averages across the test scenarios. The average error in the estimation of waiting times in cases with significant average waiting time values is observed to be at most 10%. The server workload estimation errors are in agreement with the values reported by Larson (1975) and Budge et al. (2009). Overall, we deem the accuracy of the approximation well within the acceptable range in most practical applications. Comparing the error margins with those obtained with a full backup assumption in the previous subsection shows the effectiveness of the method in handling partial backups in the presence of priorities in the queues.

We mentioned before that because of potential equity related applications, we have used a regular grid for demand locations. This, however, comes at the cost of impractically long simulation run times as the convergence rate of simulation outputs corresponding to demand locations with very low intensities will be extremely slow. This issue is further aggravated by the relatively large scale of the model considered here ($M = 447$), and is naturally more pronounced with larger numbers of non-zero elements of b_{ijp}, that is less stringent coverage threshold scenarios. We consider this a major drawback of using simulation models in studying problems with hugely varying arrival rates; moreover, we assume the reported error margins to be, to some extent, overestimated because of this issue, particularly in less stringent coverage threshold scenarios. The noticeable increase in the error margins with increasing coverage thresholds can be attributed in part to this issue, and in part to the fact that in this application, the total demand faced by the system (λ) increases with increasing coverage thresholds, thus leading to bigger error magnitudes.

Table 4.9 Estimation errors for the loss system (in %)

	C1	C2	C3	C4	C5	C6
$\bar{E}_{\rho}$	1.68	1.51	1.50	1.31	1.09	0.70
$\bar{E}_{a^1}$	5.41	4.28	3.30	3.98	1.36	0.96
$\bar{E}_{a^2}$	5.43	4.31	3.30	3.04	2.85	0.94
$\bar{E}_{a^3}$	5.53	4.40	3.36	1.54	3.78	1.02
$\bar{E}_{a}$	5.43	4.31	3.32	3.23	2.47	0.96

Table 4.10 Estimation errors for the queuing system (in %)

	C1	C2	C3	C4	C5	C6
$\bar{E}_{\rho}$	1.69	1.59	1.26	1.34	1.14	0.46
$\bar{E}^{I}_{a^1}$	5.38	5.16	4.29	5.24	1.59	1.35
$\bar{E}^{I}_{a^2}$	5.40	5.16	4.29	3.64	3.83	1.38
$\bar{E}^{I}_{a^3}$	5.50	5.22	4.38	1.63	5.37	1.43
$\bar{E}^{D}_{a^1}$	0.09	1.16	2.25	0.91	1.27	1.13
$\bar{E}^{D}_{a^2}$	0.09	1.15	2.37	1.81	1.82	1.17
$\bar{E}^{D}_{a^3}$	0.13	1.17	2.47	1.31	0.83	1.29
$\bar{E}_{a}$	5.42	5.19	4.01	3.97	3.19	1.20

4.5.3 Complementary Computational Results

We now briefly summarize the extra computational experiments presented in the electronic companion to investigate important aspects of the algorithm.

We examine the sensitivity of the algorithm to the service time distribution in section B.1 and conclude that while the approximations of server workloads and dispatch rates remain fairly insensitive to the service time distribution, the waiting time estimations remain valid only if the service time distribution has a CV close to unity.

In Section B.2, we verify the importance of modeling location and priority dependent service

Table 4.11 Waiting time estimation errors with actual values from simulation in parentheses (in minutes)

	C1	C2	C3	C4	C5	C6
$\bar{E}_{w^1}$	0.00 (0.014)	0.13 (0.69)	0.46 (5.07)	0.10 (0.54)	0.67 (27.22)	0.56 (24.84)
$\bar{E}_{w^2}$	0.00 (0.024)	0.17 (0.85)	0.72 (7.16)	0.46 (4.81)	0.45 (6.03)	1.27 (44.59)
$\bar{E}_{w^3}$	0.01 (0.037)	0.21 (1.03)	1.43 (9.58)	2.33 (30.44)	0.15 (0.83)	2.47 (61.10)
$\bar{E}_{w}$	0.00 (0.021)	0.16 (0.80)	0.69 (6.72)	0.58 (6.26)	0.49 (13.22)	1.18 (40.35)

times through test cases in which service times are identical or depend only on location or priority. It is then clearly seen that letting the service times simultaneously depend on customer and server locations and call priority, significantly improves the approximation accuracy and hence is worth the extra modeling effort.

Repeating the numerical tests for varying load factors, we show in Section B.3 that regardless of the system workload, the approximations provided by the algorithm remain accurate in all performance measures and across all tested scenarios.

Finally, in Section B.4 we consider the computational expense of the algorithm and suggest optimal intervals to repeat time-consuming calculations so that the computational overhead is reduced without significantly affecting the accuracy.

4.6 Conclusions

We extended the EMS approximation procedure of Larson (1975) by letting each server be responsible for an arbitrary subset of the demand locations (partial backups) and allowing priorities in the queues. We considered queuing and loss systems and let service times depend on the locations of the customer and server and the call priority. The approximation is based on the queue or loss systems with partial service which we defined and analyzed in steady-state.

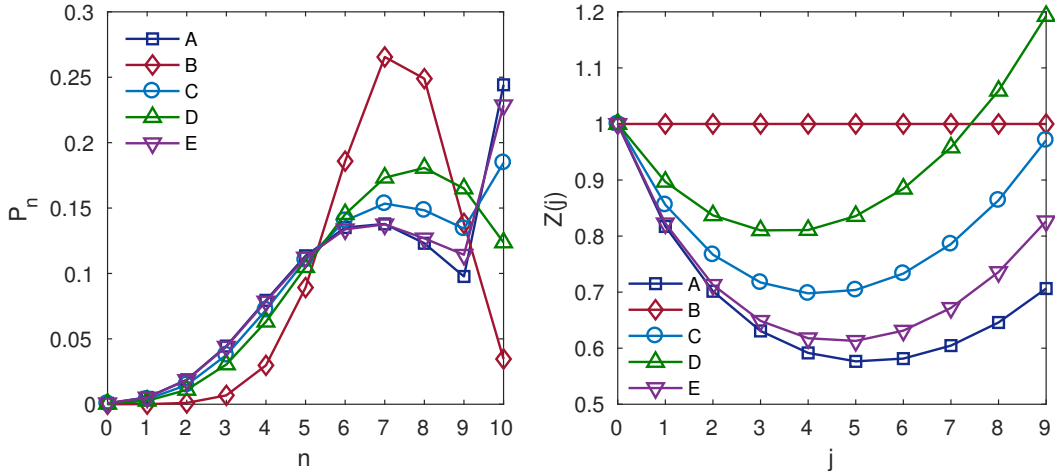
We conducted numerical experiments to validate the approximation model and demonstrated that it can accurately predict the performance measures of typical EMS systems with real scales and different operational scenarios.

The proposed method facilitates the more realistic description of emergency response systems where the assumptions of full backups and no priorities in the queues are often too simplistic to represent the dispatching policies of the actual system. In particular, the algorithm paves the way to efficient and reliable analysis and design of systems with multi-tier customers and heterogeneous fleet where dispatch decisions are made based on the customer and server types and only if an accordingly determined minimum response time is not expected to be exceeded.

An interesting line of research to follow will be to extend the current approximation procedure to model EMS systems that intentionally queue or reject lower priority requests to maintain a strategic reserve of available servers for upcoming higher priority calls. A similar extension to the exact hypercube model has been given by Iannoni et al. (2015).

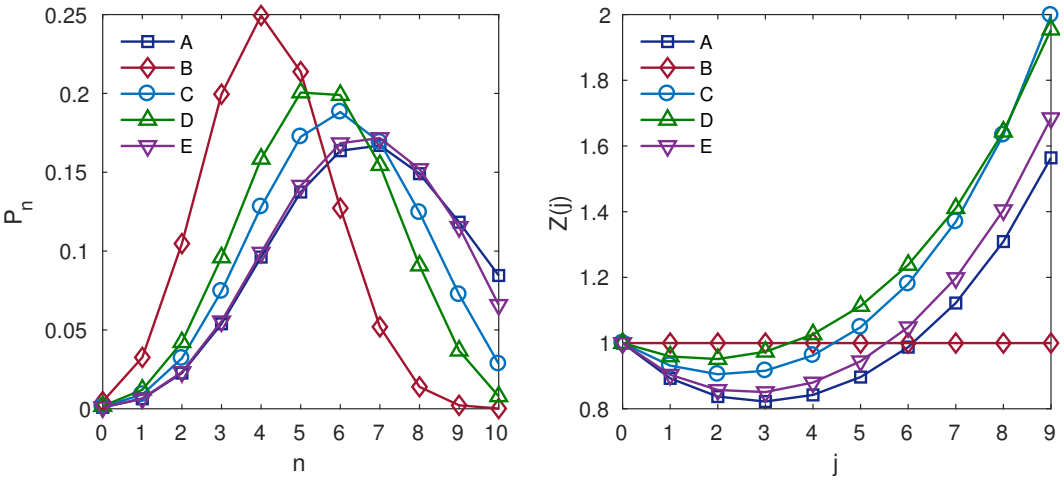
4.7 Complementary Computational Experiments

In Appendix B, we present extra computational experiments conducted to assess the performance of the approximation method.



(a) State probabilities for queued system

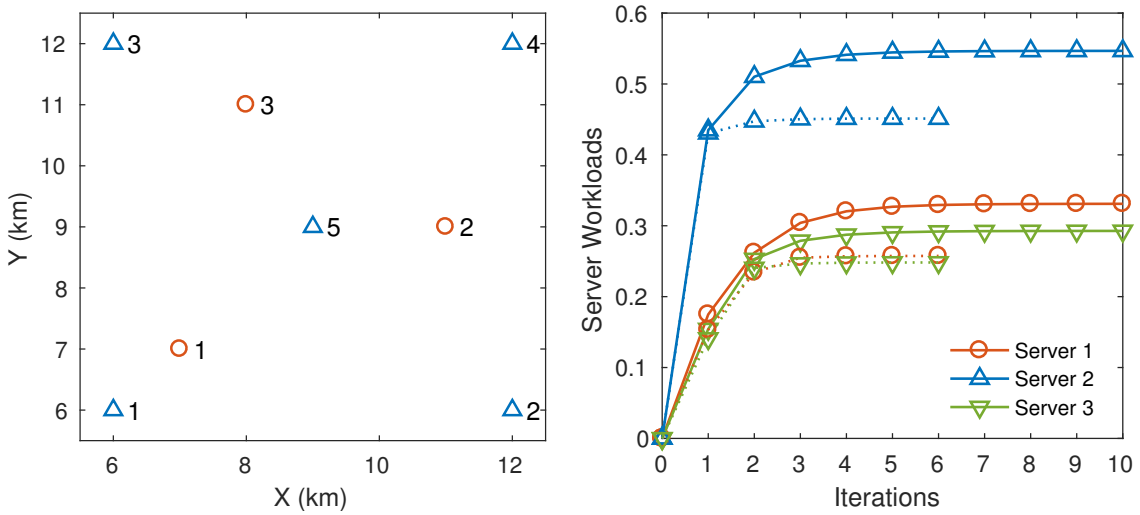
(b) Correction factors for queued system



(c) State probabilities for loss system

(d) Correction factors for loss system

Figure 4.2 Distribution of the number of busy servers and the Z correction factors for an example queuing system with $N = 10$, $\mu = 1.4$ and different arrival rate scenarios given by: A) $[\lambda_c] = [0, 0, 0, 0, 0, 0, 0, 0, 0, 10]$, B) $[\lambda_c] = [10, 0, 0, 0, 0, 0, 0, 0, 0, 0]$, C) $[\lambda_c] = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$, D) $[\lambda_c] = [2, 2, 2, 2, 2, 0, 0, 0, 0, 0]$, and E) $[\lambda_c] = [0, 0, 0, 0, 0, 2, 2, 2, 2, 2]$.



(a) Locations of the demand zones (triangles) and servers (circles) for the example problem.

(b) Convergence of server workloads for the queuing (solid) and loss (dotted) systems.

Figure 4.3 Illustrative Example

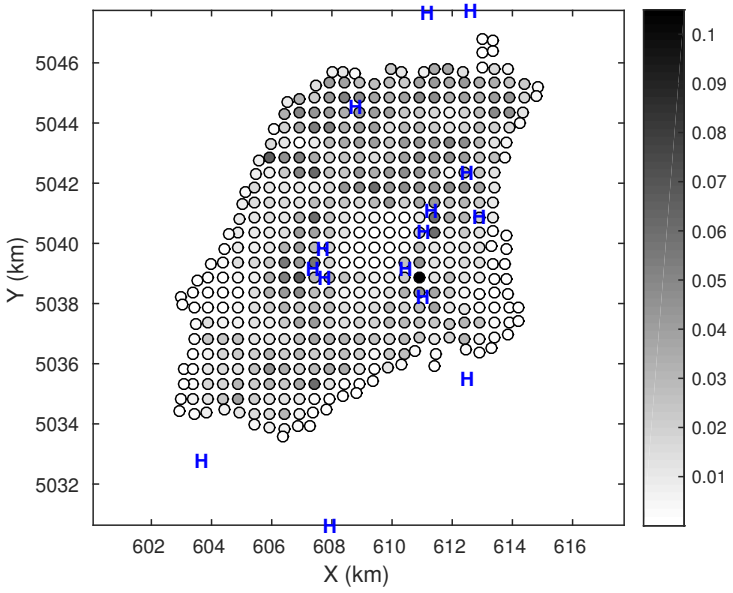


Figure 4.4 Demand distribution and hospital locations.

CHAPTER 5 DISTANCE DISTRIBUTIONS WITH BOUNDARY EFFECTS

The literature on expected distances and distance distributions is vast and scattered across multiple disciplines with identical results sometimes rediscovered by different authors as intermediate steps towards their respective and potentially unrelated research goals. This makes it very challenging to trace the developments and provide a well rounded review of the literature which dates back to as early as 1920s. Hence, no attempt is made here to give an exhaustive review, instead, we refer to Moltchanov (2012), Tong et al. (2017) and Tong et al. (2017) for excellent surveys of the literature and overview of the fundamental concepts. Here we are concerned with the distribution of the random Euclidean or Manhattan distance from a random uniformly distributed point to the n-th nearest neighbor out of N points which are also uniformly and independently distributed on a 2D region of a finite area. Making simplifying assumptions on the shape of the boundary of the region, we obtain expressions for the CDF, PDF and the moments of this distance, which are then used in Chapter 6.

Assuming a Poisson Point Process (PPP) with intensity σ, the PDF of R_n the random Euclidean distance from an arbitrary origin to its n-th neighbor is

$$f_{\mathrm{R}_n}(r) = \frac{2(\pi\sigma)^n}{(n-1)!} r^{2n-1} e^{-\pi\sigma r^2}, \quad r \geq 0,\ n = 1, 2, \cdots, \tag{5.1}$$

which, according to Dacey (1964), was first proposed by Hertz in 1909 for $n = 1$, with extensions to $n > 1$ given by Chandrasekhar (See Van Kampen (1992)), Skellam (1951), and Morisita (1954). These were extended by Thompson (1956) who derived the joint distribution of the distances to the first, up to the n-th neighbor.

From (5.1), the l-th moment of R_n is easily obtained as

$$\mathrm{E}[{\mathrm{R}_n}^l] = \frac{\Gamma(n + \frac{l}{2})}{(\pi\sigma)^{\frac{l}{2}}(n-1)!}. \tag{5.2}$$

For the Manhattan distance metric, (5.1) and (5.2) hold with every π replaced by 2.

The Poisson point process assumes an infinite number of points dispersed over an unbounded plane with the intensity σ. The number of points falling into a given subset of the plane is assumed purely random and independent of the number of points in another subset. These assumption rarely reflect the reality of many applications where a finite number of entities is distributed over a finite region and thus the number of points falling into different subsets of the region are obviously not independent. Moreover, with a bounded region, the choice of

the reference point from which the distances are measured becomes important as boundary effects will be stronger when the reference point is closer to the edges of the region. This is in contrast to the assumption of the PPP where there exist no bounds and hence every point, including a point of the process, can be taken as the reference with the same distribution to the neighbors as (5.1). Despite its limitations, the PPP and thus (5.1) are used in the vast majority of the literature because of it simplicity and mathematical tractability which, in many cases, allows for new closed form expressions for various performance metrics of a distributed system, for example the quality of a service received by a typical user of a wireless network. Whether explicitly mentioned or not, the use of (5.1) implies that the conditions under which PPP will be a good approximation to the actual distribution of nodes in the system are met. In particular, as N/A the density of the distribution of the points increases, the approximation of R_n by (5.1) with $\sigma = N/A$ improves. For a given N/A, the approximation is more accurate for a smaller n.

To make the analysis tractable, we make assumptions regarding the shape of the region; for the Euclidean metric, we assume the region to be a disk and for the Manhattan metric, we assume a square region and a tilt angle of 45 degrees between the travel directions and the sides of the region.

In view of the observations made by Larson and Odoni (1981) (Chapter 3, Section 7.1) regarding the relative insensitivity of the expected distances to the exact geometry of the region, we will assume that these expressions remain viable approximations to the "edge-corrected" nearest neighbor distances, even if the boundary shape assumptions are not strictly met, perhaps as long as the region under consideration is fairly compact and fairly convex.

5.1 Euclidean Metric

Let $p(R, r, x)$ be the probability of a randomly chosen point in a circular region of radius R being at Euclidean distance $\mathrm{D} < r$ from an arbitrary reference point p_0 which is at distance x from the center of the region. This probability is equal to the ratio of the overlapping area of the region and a circle of radius r with origin p_0 to the area of the entire region. For $r \leq R - x$ we have $p(R, r, x) = (r/R)^2$ and

$$p(R,r,x) = A^{-1}\left(r^2 \arccos(\frac{d^2+r^2-R^2}{2rd}) + R^2 \arccos(\frac{d^2+R^2-r^2}{2dR}) \right.$$
$$\left. -\frac{1}{2}\sqrt{((R+r-d)(d+r-R)(R+d-r)(R+r+d))}\right),$$

for $r > R - x$. We thus obtain the CDF of the distance to the n-th nearest neighbor of point p_0 conditional on x as

$$\begin{aligned} F_{R_n|x}(r) =& 1 - \sum_{i=0}^{n-1} \binom{N}{i} (p(R,r,x))^i (1 - p(R,r,x))^{N-i} \\ =& \frac{\beta_{p(R,r,x)}(n, N-n+1)}{B(n, N-n+1)} = I_{p(R,r,x)}(n, N-n+1)\,, \end{aligned} \tag{5.3}$$

where $B(a,b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$, $\beta_z(a,b) = \int_0^z t^{a-1}(1-t)^{b-1}dt$, and $I_z(a,b) = \beta_z(a,b)/B(a,b)$ are the beta, the incomplete beta, and the regularized incomplete beta functions, respectively.

Taking derivative with respect to r, we get the corresponding PDF as

$$f_{R_n|x}(r) = \frac{d}{dr} F_{R_n|x}(r) = \frac{dp(R,r,x)}{dr} \frac{p(R,r,x)^{N-n}(1 - p(R,r,x))^{n-1}}{\beta(n, N-n+1)}\,,$$

with

$$\frac{dp(R,r,x)}{dr} = \begin{cases} \frac{2r}{R^2} & r \le R - x\,, \\ \frac{2r}{\pi R^2} \operatorname{arcsec}(\frac{2dr}{d^2 + r^2 - R^2}) & r > R - x\,. \end{cases}$$

We can now integrate (5.3) to obtain $\tilde{F}_{R_n}(r)$ the CDF of the distance to the n-th neighbor from a uniformly randomly chosen reference point inside the region. We write

$$\begin{aligned} \tilde{F}_{R_n}(r) &= \int_0^{2R} \Pr\{X = x\} F_{R_n|x}(r) dx = \frac{2}{R^2} \int_0^{2R} x F_{R_n|x}(r) dx \\ &= \frac{2}{R^2} \int_0^{|R-r|} x I_{\min\{1, (\frac{r}{R})^2\}}(n, N-n+1)\, dx + \frac{2}{R^2} \int_{|R-r|}^{R} x I_{p(R,r,x)}(n, N-n+1) dx \\ &= \frac{(R-r)^2}{R^2} I_{\min\{1, (\frac{r}{R})^2)\}}(n, N-n+1) + \frac{2}{R^2} \int_{|R-r|}^{R} x I_{p(R,r,x)}(n, N-n+1)\, dx\,. \end{aligned} \tag{5.4}$$

The integral is not analytically solvable and needs to be evaluated numerically. Proceeding with the Gauss-Chebychev quadrature with K nodes, we write

$$\begin{aligned} \Phi(r) &= \frac{2}{R^2} \int_{|R-r|}^{R} x I_{p(R,r,x)}(n, N-n+1)\, dx \\ &\approx \frac{(R - |R-r|)\pi}{KR^2} \sum_{i=1}^{K} x_i \sqrt{1 - u_i^2}\, I_{p(R,r,x_i)}(n, N-n+1) + E_K\,, \end{aligned} \tag{5.5}$$

with

$$x_i = \frac{1}{2}((R - |R-r|)u_i + R + |R-r|)\,,$$

$$u_i = \cos(\frac{2i-1}{2K}\pi)\,, \tag{5.6}$$

and the quadrature error E_K tending to zero as $K \to \infty$. Assuming a large enough K we then get

$$\begin{aligned}\tilde{F}_{R_n}(r) \approx \frac{(R-r)^2}{R^2} I_{\min\{1,(\frac{r}{R})^2)\}}(n, N-n+1) \\ + \frac{(R-|R-r|)\pi}{KR^2} \sum_{i=1}^{K} x_i \sqrt{1-u_i^2}\, I_{p(R,r,x_i)}(n, N-n+1)\,.\end{aligned} \tag{5.7}$$

We obtain the PDF $\tilde{f}_{R_n}(r)$ by differentiating (5.4); we write

$$\begin{aligned}\tilde{f}_{R_n}(r) &= \frac{d}{dr}\left[\int_0^R \frac{2x}{R^2} F_{R_n|x}(r)dx\right] = \frac{d}{dr}\left[\int_0^R \frac{2x}{R^2} I_{p(R,r,x)}(n, N-n+1)dx\right] \\ &= \int_0^R \frac{d}{dr}\left[\frac{2x}{R^2} I_{p(R,r,x)}(n, N-n+1)dx\right] \\ &= \int_0^R \frac{2x}{R^2} \frac{dp(R,r,x)}{dr} \frac{p(R,r,x)^{n-1}(1-p(R,r,x))^{N-n}}{B(n, N-n+1)}\, dx \\ &= \int_0^{|R-r|} \frac{2x}{R^2}\frac{2r}{R^2} \frac{(\frac{r}{R})^{2(n-1)}(1-(\frac{r}{R})^2)^{N-n}}{B(n, N-n+1)}\, dx \\ &+ \int_{|R-r|}^R \frac{2x}{R^2}\frac{2r}{\pi R^2} \sec^{-1}(\frac{2rx}{r^2+x^2-R^2}) \frac{p(R,r,x)^{n-1}(1-p(R,r,x))^{N-n}}{B(n, N-n+1)}\, dx \\ &\approx 1(r<R) \frac{2r(R-r)^2(\frac{r}{R})^{2(N-n)}(1-\frac{r}{R})^{n-1}N!}{R^2(n-1)!(N-n)!} \\ &+ \frac{2r(R-|R-r|)}{KR^4} \sum_{i=1}^{K} \sin(\theta_i)\, x_i\, \sec^{-1}(\frac{2rx_i}{r^2+x_i^2-R^2}) \frac{p(R,r,x_i)^{n-1}(1-p(R,r,x_i))^{N-n}N!}{(n-1)!(N-n)!}\,,\end{aligned} \tag{5.8}$$

where $B(a,b)$ is the beta function.

We are now looking to obtain the moments of the distance to the n-th neighbor. We write

$$\begin{aligned}E[\tilde{R}_n^l] &= \int_0^{2R} r^l d\tilde{F}_{R_n}(r) = \int_0^{2R} r^l \frac{d}{dr}\left[\int_0^R \frac{2x}{R^2} F_{R_n|x}(r)dx\right] dr \\ &= \int_0^{2R} r^l \frac{d}{dr}\left[\int_0^R \frac{2x}{R^2} I_{p(R,r,x)}(n, N-n+1)dx\right] dr \\ &= \int_0^R r^l \frac{d}{dr}\left[\frac{(R-r)^2}{R^2} I_{(\frac{r}{R})^2}(n, N-n+1) + \int_{R-r}^R \frac{2x}{R^2} I_{p(R,r,x)}(n, N-n+1)dx\right] dr\end{aligned}$$

$$
\begin{aligned}
&+\int_R^{2R} r^l \frac{d}{dr}\left[\frac{(R-r)^2}{R^2}+\int_{r-R}^{R}\frac{2x}{R^2}I_{p(R,r,x)}(n,N-n+1)dx\right]dr\\
&=\int_0^{R} r^l \frac{d}{dr}\left[\frac{(R-r)^2}{R^2}I_{(\frac{r}{R})^2}(n,N-n+1)\right]dr+\int_R^{2R} r^l \frac{d}{dr}\left[\frac{(R-r)^2}{R^2}\right]dr\\
&+\int_0^{R} r^l \frac{d}{dr}\int_{R-r}^{R}\frac{2x}{R^2}I_{p(R,r,x)}(n,N-n+1)\,dx\,dr\\
&+\int_R^{2R} r^l \frac{d}{dr}\int_{r-R}^{R}\frac{2x}{R^2}I_{p(R,r,x)}(n,N-n+1)\,dx\,dr\,.
\end{aligned}
$$

We now evaluate each integral denoting them I_1, I_2, I_3 and I_4. We have for I_1

$$
\begin{aligned}
I_1&=\int_0^{R} r^l \frac{d}{dr}\left[\frac{(R-r)^2}{R^2}I_{(\frac{r}{R})^2}(n,N-n+1)\right]dr\\
&=\frac{r^l(R-r)^2 I_{(r/R)^2}(n,N-n+1)}{R^2}\bigg|_0^R-\int_0^R\frac{(R-r)^2 l r^{l-1}}{R^2}I_{(r/R)^2}(n,N-n+1)\,dr\\
&=-\int_0^R\frac{(R-r)^2 l r^{l-1}}{R^2}I_{(r/R)^2}(n,N-n+1)\,dr\\
&=-\int_0^R\frac{(R-r)^2 l r^{l-1}}{R^2 B(n,N-n+1)}\left[(\frac{r}{R})^{2n}\sum_{k=0}^{\infty}\frac{(n-N)_k}{k!(n+k)}(\frac{r}{R})^{2k}\right]dr\\
&=-\frac{l}{R^2B(n,N-n+1)}\sum_{k=0}^{N-n}\frac{(n-N)_k}{k!(n+k)}\int_0^R(\frac{r}{R})^{2(n+k)}(R-r)^2 r^{l-1}\,dr\\
&=-\frac{l}{R^2B(n,N-n+1)}\sum_{k=0}^{N-n}\frac{(n-N)_k}{k!(n+k)}\left(\frac{2R^{l+2}}{(2k+l+2n)_3}\right)\\
&=-\frac{2lR^l}{B(n,N-n+1)}\sum_{k=0}^{N-n}\frac{(n-N)_k}{k!(n+k)(2k+l+2n)_3}
\end{aligned}
$$

where $(x)_n=\Gamma(x+n)/\Gamma(x)$ is the Pochhammer symbol and we have used a series expansion of the incomplete beta function and the fact that $(n-N)_k=0$ for $k>N-n$.

For the next integral we have

$$
\begin{aligned}
I_2&=\int_R^{2R} r^l\frac{d}{dr}\frac{(R-r)^2}{R^2}\,dr=\frac{r^l(R-r)^2}{R^2}\bigg|_R^{2R}-\int_R^{2R}\frac{(R-r)^2 l r^{l-1}}{R^2}\,dr\\
&=(2R)^l-\left[r^l\left(lr\left(\frac{r}{R^2(l+2)}-\frac{2}{l(R+1)}\right)+1\right)\right]_R^{2R}\\
&=(2R)^l-\left[\frac{(2R)^l(l(l-1)+2)}{(l+1)(l+2)}-\frac{2R^l}{(l+1)(l+2)}\right]\\
&=\frac{2R^l(2^{l+1}l+1)}{(l+1)(l+2)}
\end{aligned}
$$

Combining I_3 and I_4 into $I_{34} = I_3 + I_4$, we write

$$\begin{aligned}
I_{34} &= \int_0^{2R} r^l \frac{d}{dr}\left[\int_{|r-R|}^{R} \frac{2x}{R^2} I_{p(R,r,x)}(n, N-n+1)dx\right] dr \\
&= \left[\int_{|r-R|}^{R} \frac{2x}{R^2} I_{p(R,r,x)}(n, N-n+1)dx\right]_{r=0}^{r=2R} - \int_0^{2R} l r^{l-1}\Phi(r)\, dr \\
&= 0 - R\int_{-1}^{1} l(R(1+u))^{l-1}\Phi(R(1+u))\, du \\
&= lR^l \int_{-1}^{1} (1+u)^{l-1}\Phi(R(1+u))\, du \\
&= \frac{\pi l R^l}{K} \sum_{j=1}^{K} \sqrt{1-u_j^2}(1+u_j)^{l-1}\Phi(R(1+u_j))\,,
\end{aligned}$$

where we have again used a Gauss-Chebychev quadrature with K nodes and with u_j defined as in (5.6). We remember that $\Phi(r)$ was obtained in (5.5) as a quadrature itself. We thus get the l-th moment of the distance as

$$\begin{aligned}
E[\tilde{R}_n^l] &= I_1 + I_2 + I_{34} \\
&= \frac{2R^l(2^{l+1}l+1)}{(l+1)(l+2)} - \frac{2lR^l N!}{(n-1)!(N-n)!} \sum_{k=0}^{N-n} \frac{(n-N+k-1)!(2k+2n+l-1)!}{k!(n+k)(n-N-1)!(2k+2n+l+2)!} \\
&+ \frac{\pi l R}{K} \sum_{j=1}^{K} \sin(\theta_j) r_j^{l-1}\Phi(r_j)\,,
\end{aligned}$$

with $r_j = R(1+\cos(\theta_j))$ and $\theta_j = \frac{2j-1}{2K}\pi$.

5.2 Manhattan Metric

We now repeat the analysis for the case with the Manhattan distance metric. In deriving these preliminary results, we will assume that the region is a square and the direction of travel is tilted at 45 degrees relative to the sides of the square. This allows us to obtain the integrals exactly without resorting to quadrature methods.

Let us divide the square region with half-diagonal R, that is $\mathcal{A} = \{(x,y)| -\frac{R}{\sqrt{2}} \le x, y \le \frac{R}{\sqrt{2}}\}$, into subregions $\mathcal{A}_1 = \{(x,y)\,|\,\frac{r-R}{\sqrt{2}} \le x, y \le \frac{R-r}{\sqrt{2}}\}$ and $\mathcal{A}_2 = \{(x,y)\,|\,\frac{R-r}{\sqrt{2}} \le x \le \frac{R}{\sqrt{2}}, \frac{r-R}{\sqrt{2}} \le y \le \frac{R-r}{\sqrt{2}}\}$, and $\mathcal{A}_3 = \{(x,y)\,|\,\frac{R-r}{\sqrt{2}} \le x, y \le \frac{R}{\sqrt{2}}\}$. Moreover, let $\mathcal{A}(r,x,y)$ be the square region of half-diagonal r centered at $(x,y) \in \mathcal{A}$. Because of the symmetry, we write

$$\tilde{F}_{R_n}(r) = \iint_A F_{R_n|(x,y)}(r)\frac{1}{2R^2}\, dx\, dy$$

$$= \frac{1}{2R^2}\left\{\iint_{A_1} F_{R_n|(x,y)}(r)\,dx\,dy + 4\iint_{A_2} F_{R_n|(x,y)}(r)\,dx\,dy + 4\iint_{A_3} F_{R_n|(x,y)}(r)\,dx\,dy\right\}, \tag{5.9}$$

where $F_{R_n|(x,y)}(r)\,dx\,dy = I_{p(R,r,x,y)}(n, N-n+1)$ and $p(R,r,x,y) = |\mathcal{A}\cap\mathcal{A}(r,x,y)|/|\mathcal{A}|$. We first consider the case $0 \le r \le R$. For $(x,y) \in \mathcal{A}_1$ and $0 \le r \le R$, we have $p(R,r,x,y) = (\frac{r}{R})^2$ and thus

$$\iint_{A_1} F_{R_n|(x,y)}(r)\,dx\,dy = |\mathcal{A}_1|\, I_{(\frac{r}{R})^2}(n, N-n+1) = 2(R-r)^2\, I_{(\frac{r}{R})^2}(n, N-n+1)\,.$$

For $(x,y) \in \mathcal{A}_2$ and $0 \le r \le R$, we have $p(R,r,x,y) = (2r^2 - rx_2\sqrt{2})/2R^2$ with $x' = x - \frac{R-r}{\sqrt{2}}$, and thus

$$\begin{aligned}
\iint_{A_2} F_{R_n|(x,y)}(r)\,dx\,dy &= \int_0^{\sqrt{2}(R-r)}\int_0^{\frac{r}{\sqrt{2}}} I_{(\frac{2r^2-rx'\sqrt{2}}{2R^2})}(n, N-n+1)\,dx'\,dy' \\
&= \int_0^{\sqrt{2}(R-r)} \frac{1}{B(n,N-n+1)}\left[\frac{\sqrt{2}R^2}{r}\beta_{\frac{r^2}{2R^2}}(n+1,N-n+1) - \frac{r}{\sqrt{2}}\beta_{\frac{r^2}{2R^2}}(n,N-n+1)\right. \\
&\quad \left. -\frac{\sqrt{2}R^2}{r}\beta_{(\frac{r}{R})^2}(n+1,N-n+1) + \sqrt{2}r\beta_{(\frac{r}{R})^2}(n,N-n+1)\right]\,dy' \\
&= \frac{\sqrt{2}(R-r)}{B(n,N-n+1)}\left[\frac{\sqrt{2}R^2}{r}\beta_{\frac{r^2}{2R^2}}(n+1,N-n+1) - \frac{r}{\sqrt{2}}\beta_{\frac{r^2}{2R^2}}(n,N-n+1)\right. \\
&\quad \left. -\frac{\sqrt{2}R^2}{r}\beta_{(\frac{r}{R})^2}(n+1,N-n+1) + \sqrt{2}r\beta_{(\frac{r}{R})^2}(n,N-n+1)\right] \\
&= \frac{(R-r)}{B(n,N-n+1)}\left[\frac{2R^2}{r}\beta_{\frac{r^2}{2R^2}}(n+1,N-n+1) - r\beta_{\frac{r^2}{2R^2}}(n,N-n+1)\right. \\
&\quad \left. -\frac{2R^2}{r}\beta_{(\frac{r}{R})^2}(n+1,N-n+1) + 2r\beta_{(\frac{r}{R})^2}(n,N-n+1)\right].
\end{aligned}$$

For $(x,y) \in \mathcal{A}_3$ and $0 \le r \le R$, we have $p(R,r,x,y) = (2r^2 - \sqrt{2}\,r(x'+y') + x'y')/2R^2$, and write

$$\begin{aligned}
\iint_{A_3} F_{R_n|(x,y)}(r)\,dx\,dy &= \int_0^{\frac{r}{\sqrt{2}}}\int_0^{\frac{r}{\sqrt{2}}} I_{\left(\frac{2r^2-\sqrt{2}\,r(x'+y')+x'y'}{2R^2}\right)}(n, N-n+1)\,dx'\,dy' \\
&= \int_0^{\frac{r}{\sqrt{2}}}\int_0^{\frac{r}{\sqrt{2}}} \frac{1}{B(n,N-n+1)}\sum_{k=0}^{N-n}\frac{(n-N)_k}{k!(n+k)}\left(\frac{2r^2-\sqrt{2}r(x'+y')+x'y'}{2R^2}\right)^{n+k} dx'\,dy' \\
&= \sum_{k=0}^{N-n}\frac{(n-N)_k}{k!(n+k)B(n,N-n+1)}\int_0^{\frac{r}{\sqrt{2}}}\int_0^{\frac{r}{\sqrt{2}}}\left(\frac{2r^2-\sqrt{2}r(x'+y')+x'y'}{2R^2}\right)^{n+k} dx'\,dy'
\end{aligned}$$

$$= \sum_{k=0}^{N-n} \frac{N!(n-N)_k}{k!(n-1)!(N-n)!(n+k)} \int_0^{\frac{r}{\sqrt{2}}} \left\{ \frac{r(2^{n+k+1}-1)}{\sqrt{2}(n+k+1)} \left(\frac{2r^2 - \sqrt{2}ry'}{4R^2} \right)^{n+k} \right\} dy'$$

$$= \sum_{k=0}^{N-n} \frac{N!(n-N)_k}{k!(n-1)!(N-n)!(n+k)} \left\{ \frac{(2^{n+k+1}-1)^2 r^{2n+2k+2}}{2(n+k+1)^2(2R)^{2n+2k}} \right\}$$

$$= \sum_{k=0}^{N-n} \frac{(-1)^k N!}{2k!(n-1)!(N-n-k)!(n+k)} \frac{(2^{n+k+1}-1)^2 r^{2n+2k+2}}{(n+k+1)^2(2R)^{2n+2k}} .$$

Plugging these results into (5.9) yields

$$\begin{aligned} \tilde{F}_{R_n}(r; 0 \le r \le R) = \frac{(R-r)^2}{R^2} I_{(\frac{r}{R})^2}(n, N-n+1) + \frac{2(R-r)N!}{R^2(n-1)!(N-n)!} \left[\frac{2R^2}{r} \beta_{\frac{r^2}{2R^2}}(n+1, N-n+1) \right. \\ \left. - r\beta_{\frac{r^2}{2R^2}}(n, N-n+1) - \frac{2R^2}{r} \beta_{(\frac{r}{R})^2}(n+1, N-n+1) + 2r\beta_{(\frac{r}{R})^2}(n, N-n+1) \right] \\ + \frac{N!}{R^2(n-1)!} \sum_{k=0}^{N-n} \frac{(-1)^k}{k!(N-n-k)!(n+k)(n+k+1)^2} \frac{(2^{n+k+1}-1)^2 r^{2n+2k+2}}{(2R)^{2n+2k}} . \quad (5.10) \end{aligned}$$

For $R < r \le 2R$, we follow a similar approach. For $(x, y) \in \mathcal{A}_1$ and $R < r \le 2R$, we have $p(R, r, x, y) = 1$ and

$$\iint_{A_1} F_{R_n|(x,y)}(r)\, dx\, dy = |\mathcal{A}_1|\, I_1(n, N-n+1) = 2(R-r)^2 .$$

For $(x, y) \in \mathcal{A}_2$ and $R < r \le 2R$, we have $p(R, r, x, y) = (2R^2 - Rx'\sqrt{2})/2R^2$ with $x' = x - \frac{r-R}{\sqrt{2}}$ and

$$\begin{aligned} \iint_{A_2} F_{R_n|(x,y)}(r)\, dx\, dy &= \int_0^{\sqrt{2}(r-R)} \int_0^{\frac{2R-r}{\sqrt{2}}} I_{(\frac{2R^2 - Rx'\sqrt{2}}{2R^2})}(n, N-n+1)\, dx'\, dy' \\ &= \int_0^{\sqrt{2}(r-R)} \frac{1}{B(n, N-n+1)} \left\{ \sqrt{2}R \left[(B(n, N-n+1) - B(n+1, N-n+1) \right] \right. \\ &\qquad \left. + \sqrt{2}R\beta_{\frac{r}{2R}}(n+1, N-n+1) - \frac{r}{\sqrt{2}} \beta_{\frac{r}{2R}}(n, N-n+1) \right\} dy' \\ &= \sqrt{2}(r-R) \left(\sqrt{2}R(1 - \frac{n}{N+1}) + \frac{N!}{(n-1)!(N-n)!} \left\{ \sqrt{2}R\beta_{\frac{r}{2R}}(n+1, N-n+1) - \right. \right. \\ &\qquad \left. \left. \frac{r}{\sqrt{2}} \beta_{\frac{r}{2R}}(n, N-n+1) \right\} \right) \\ &= \frac{2R(r-R)(N-n+1)}{N+1} + \frac{(r-R)N!}{(n-1)!(N-n)!} \left(2R\beta_{\frac{r}{2R}}(n+1, N-n+1) - r\beta_{\frac{r}{2R}}(n, N-n+1) \right) . \end{aligned}$$

For $(x,y) \in \mathcal{A}_3$ and $R < r \leq 2R$, we have $p(R,r,x,y) = (2R^2 - \sqrt{2}\,R(x'+y') + x'y')/2R^2$ and with the change of variables $x' = x - \frac{r-R}{\sqrt{2}}$ and $y' = y - \frac{r-R}{\sqrt{2}}$ we write

$$\begin{aligned}
&\iint_{A_3} F_{R_n|(x,y)}(r)\,dx\,dy = \int_0^{\frac{2R-r}{\sqrt{2}}} \int_0^{\frac{2R-r}{\sqrt{2}}} I_{\left(\frac{2R^2-\sqrt{2}\,R(x'+y')+x'y'}{2R^2}\right)}(n, N-n+1)\,dx'\,dy' \\
&= \int_0^{\frac{2R-r}{\sqrt{2}}} \int_0^{\frac{2R-r}{\sqrt{2}}} \frac{1}{B(n,N-n+1)} \sum_{k=0}^{N-n} \frac{(n-N)_k}{k!(n+k)} \left(\frac{2R^2 - \sqrt{2}R(x'+y') + x'y'}{2R^2}\right)^{n+k} dx'\,dy' \\
&= \sum_{k=0}^{N-n} \frac{(n-N)_k}{k!(n+k)B(n,N-n+1)} \int_0^{\frac{2R-r}{\sqrt{2}}} \int_0^{\frac{2R-r}{\sqrt{2}}} \left(\frac{2R^2 - \sqrt{2}R(x'+y') + x'y'}{2R^2}\right)^{n+k} dx'\,dy' \\
&= \sum_{k=0}^{N-n} \frac{N!(n-N)_k}{k!(n-1)!(N-n)!(n+k)} \int_0^{\frac{2R-r}{\sqrt{2}}} \left\{ \frac{(\sqrt{2}R-y)^{n+k}[(2R)^{n+k+1} - r^{n+k+1}]}{(n+k+1)2^{\frac{3}{2}k+\frac{3}{2}n+\frac{1}{2}}R^{2(n+k)}} \right\} dy' \\
&= \sum_{k=0}^{N-n} \frac{N!(n-N)_k}{k!(n-1)!(N-n)!(n+k)} \left\{ \frac{\left[(r^{n+k+1} - (2R)^{n+k+1}\right]^2}{2(n+k+1)^2(2R)^{2(n+k)}} \right\} \\
&= \sum_{k=0}^{N-n} \frac{(-1)^k N!}{2k!(n-1)!(N-n-k)!(n+k)} \frac{\left[(r^{n+k+1} - (2R)^{n+k+1}\right]^2}{(n+k+1)^2(2R)^{2(n+k)}}\,.
\end{aligned}$$

Plugging these back into (5.9), yields

$$\begin{aligned}
\tilde{F}_{R_n}(r; R \leq r \leq 2R) = \frac{(R-r)^2}{R^2} + \frac{4(r-R)(N-n+1)}{R(N+1)} + \\
\frac{2(r-R)N!}{(n-1)!(N-n)!} \left(\frac{2}{R} \beta_{\frac{r}{2R}}(n+1, N-n+1) - \frac{r}{R^2} \beta_{\frac{r}{2R}}(n, N-n+1) \right) \\
+ \frac{1}{R^2} \sum_{k=0}^{N-n} \frac{(-1)^k N!}{k!(n-1)!(N-n-k)!(n+k)} \frac{\left[(r^{n+k+1} - (2R)^{n+k+1}\right]^2}{(n+k+1)^2(2R)^{2(n+k)}}\,. \quad (5.11)
\end{aligned}$$

It is now possible to compute the PDF $\tilde{f}(r)$ as

$$\begin{aligned}
\tilde{f}_{R_n}(r;\, 0 \leq r \leq R) = \frac{2N!}{r^2R^2(n-1)!(N-n)!} \Big\{ r(r-R)^2 \left(1 - \frac{r^2}{R^2}\right)^{N-n} \left(\frac{r^2}{R^2}\right)^n \\
+ r^2(2r-R) B_{\frac{r^2}{2R^2}}(n, -n+N+1) - 2R^3 B_{\frac{r^2}{2R^2}}(n+1, -n+N+1) \\
+ r^2(R-3r) B_{\frac{r^2}{R^2}}(n, -n+N+1) + 2R^3 B_{\frac{r^2}{R^2}}(n+1, -n+N+1) \Big\} \\
+ \frac{2N!}{R^2(n-1)!} \sum_{k=0}^{N-n} \frac{(-1)^k}{k!(N-n-k)!(n+k)(n+k+1)} \frac{(2^{n+k+1}-1)^2 r^{2n+2k+1}}{(2R)^{2n+2k}}\,,
\end{aligned}$$

and

$$\tilde{f}_{R_n}(r;\, R<r\leq 2R)=\frac{2}{R^2}\left\{r+\frac{(N-2n+1)R}{N+1}+\frac{N!}{(n-1)!(N-n)!}\left[(R-2r)\beta_{\frac{r}{2R}}(n,N-n+1)\right.\right.$$
$$\left.\left.+2R\beta_{\frac{r}{2R}}(n+1,N-n+1)\right]+\sum_{k=0}^{N-n}\frac{(-1)^k N!}{k!(n-1)!(N-n-k)!(n+k)}\frac{r^{n+k}\left[(r^{n+k+1}-(2R)^{n+k+1}\right]}{(n+k+1)(2R)^{2(n+k)}}\right\}.$$

Using these relations the moments of R_n are obtained as

$$\mathrm{E}[R_n^l]=2R^l\left\{\frac{\left(2^{l+1}-1\right)(-2n+N+1)}{(l+1)(N+1)}+\frac{2^{l+2}-1}{l+2}+\frac{N!}{(n-1)!}\left[\frac{(l+n+N+1)\Gamma\left(\frac{l}{2}+n\right)}{2\Gamma\left(\frac{l}{2}+N+2\right)}\right.\right.$$
$$\left.\left.-\frac{\Gamma\left(\frac{l+1}{2}+n\right)}{\Gamma\left(\frac{l+3}{2}+N\right)}+\Phi\right]\right\},$$

where

$$\Phi=\sum_{k=0}^{N-n}\frac{-1^k}{k!(N-n-k)!}\left\{\frac{2^{-k-n}\left(l-2^{k+l+n+1}\left(k^2+k(l+2n+3)+l(n+3)+(n+1)(n+2)\right)\right)}{(k+n)(k+n+1)(k+l+n+1)(k+l+n+2)}+\right.$$
$$\frac{4\left(-2^{-k-n}+2^l+1\right)}{(k+n)(k+n+1)(2k+l+2n+2)}+\frac{2^{-k-n}\left(-2^{k+n}(4k+2l+4n+1)+2k+l+2n\right)}{(k+n)(2k+l+2n+1)(2k+l+2n+2)}+$$
$$\left.\frac{2^{-k-n+1}-2^{l+2}}{(k+n)(k+n+1)(k+l+n+1)}+\frac{2^{-k-n}\left(2^{k+n+1}-1\right)}{(k+n+1)(2k+l+2n+1)}\right\}.$$

5.2.1 Remarks

In Figure 5.1, we compare the Poisson and edge-corrected probability density functions of the Euclidean distance to the n-th nearest neighbor, given respectively by (5.1) and (5.8), for different values of n and the total number of points N. Under the assumption of a circular region, the PDF obtained from (5.8) is exact and hence the PDF obtained from the simulation is not plotted. We clearly see that as expected, while the approximation by the PPP seems acceptable for $n=1$ (the nearest neighbor), its accuracy rapidly deteriorates with increasing n for a given N. Also, the PPP nearest neighbor distance is more accurate when the number of total points is greater. The PDFs corresponding to the case with Manhattan metric are plotted in Figure 5.2 where the case with the travel directions parallel to the sides of the region, which we obtained from simulation, is also included. In addition to the similar trend for the accuracy of the PPP assumption, we see that even with the region rotated 45 degrees,

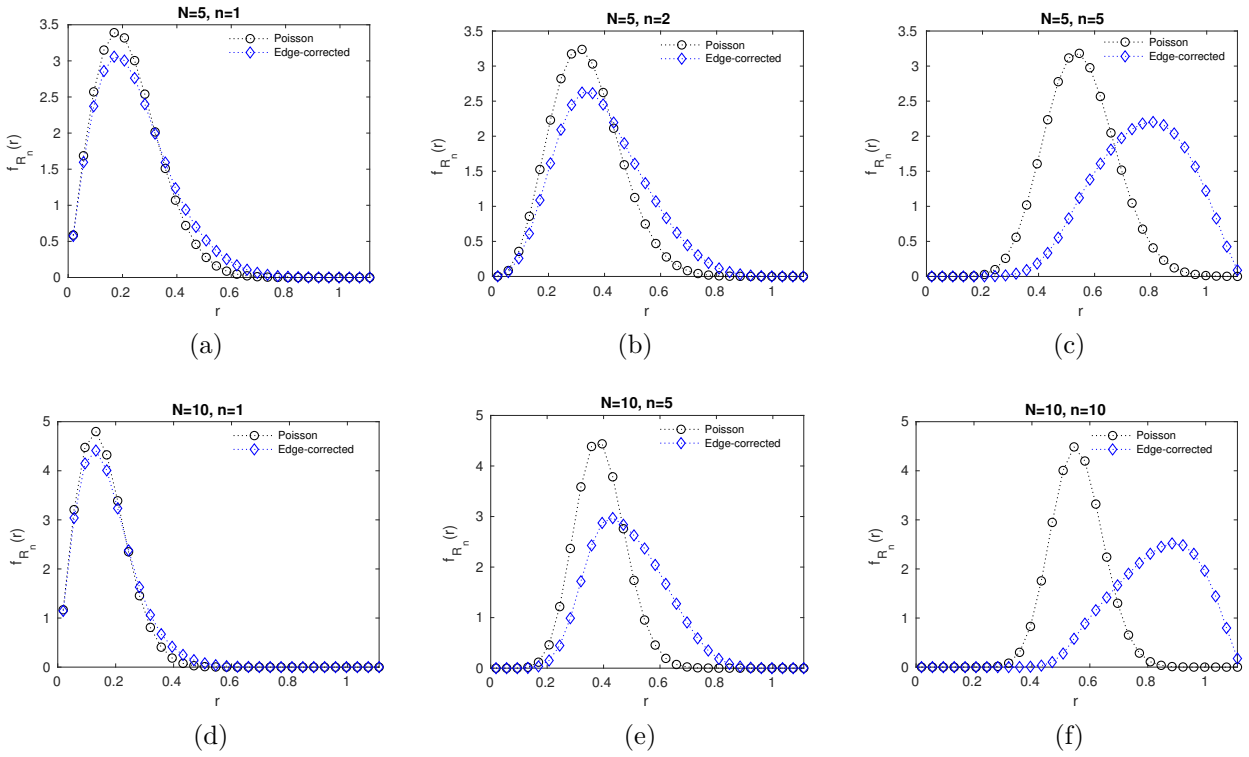


Figure 5.1 Distribution of the Euclidean distance to the n-th nearest neighbor out of N u.i.d random points.

the edge-corrected PDF remains a good approximation, particularly compared to the relation obtained through the Poisson assumption.

These observations are important to us, since in Chapter 6 we mostly deal with under-resourced (or over-loaded) emergency service systems where typically limited number of response units (or free response units) are deployed within a finite service area and the distribution of distances to every response unit is an important part of our analysis of such systems.

It might also be worth mentioning that the author has devoted considerable effort to obtain closed-form approximations of the form

$$F_{\mathrm{R}_n}(r) = \sum_{k=1}^{m} \beta_{g(r,N,n)}(n, N-n+1)$$

to the distribution of the distance to neighbors in the general case of a rectangular region of arbitrary proportions and a given metric; however, those efforts has not yet concluded at the time of writing this manuscript.

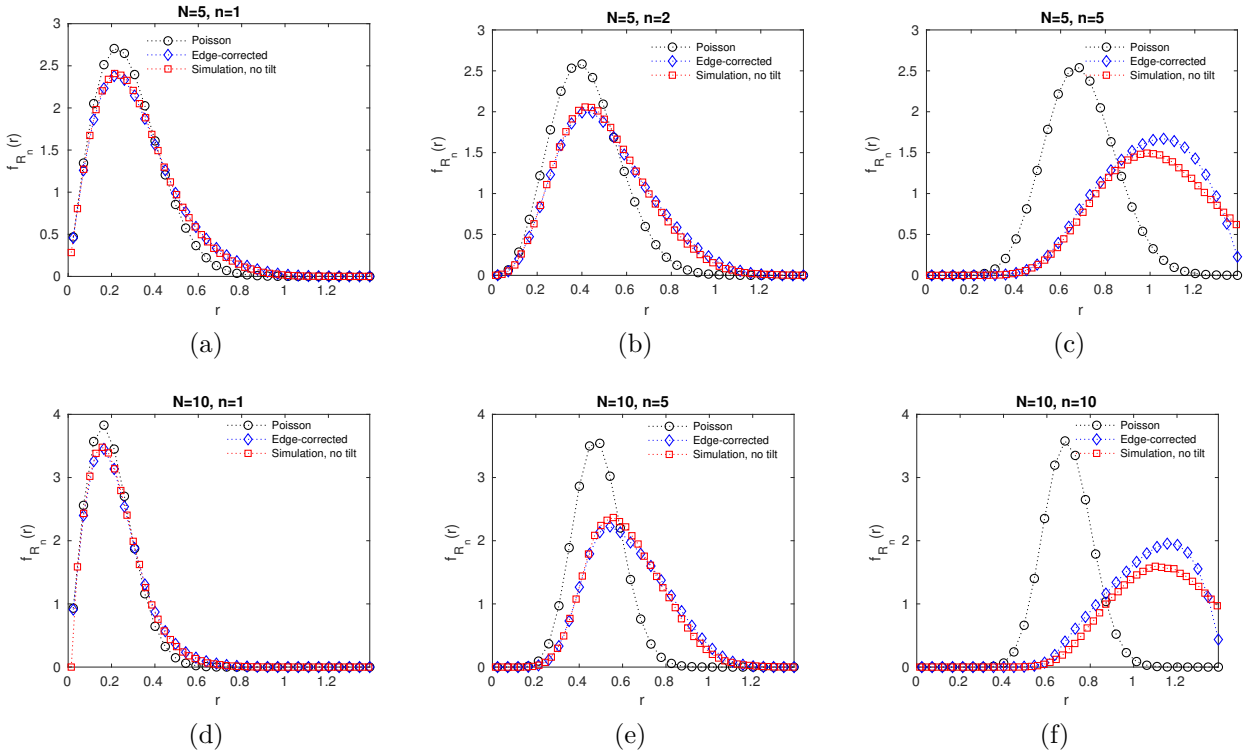


Figure 5.2 Distribution of the Manhattan distance to the n-th nearest neighbor out of N u.i.d random points.

CHAPTER 6 ON OPTIMAL DISPATCH POLICIES FOR EMERGENCY SERVICE SYSTEMS WITH DYNAMIC RELOCATION: A QUEUING THEORETICAL FRAMEWORK WITH APPLICATIONS

In Chapter 4, we presented an approximation algorithm based on queuing theory to describe the performance of an Emergency Service System (ESS) for a given set of tactical decisions on the locations of the servers, the subset of demand zones allocated to each server and the preference order of assigning incoming calls to potential candidate servers. Various prescriptive models can then be developed upon this descriptive tool by simply exploring the space of combined location and dispatch decisions via an optimization method of some sort to maximize a desirable performance metric such as the response time, intervention outcome, equity or a combination of such objectives. These optimization models will arguably be more realistic and reflective of the actual dispatchers' behavior than the models with the full backup assumption because they allow the dispatch preference lists of each priority and demand zone to have different lengths. In this text, however, we do not concern ourselves with the optimization models of this kind. Instead, recognizing the role of partial backup dispatch strategies in maximizing system performance, we present in this chapter, another general descriptive tool for studying the impact of a class of partial backup dispatch policies on the key system performance metrics in a general setting where the ESS and the demand it serves are minimally defined. As we will see, the simplifying assumptions we make, will lead us to deem the idealized situation considered here as a crude representation of a large class of ESSs with dynamic relocation.

As discussed in Chapter 4, the rationale behind partial backup dispatch policies is to improve the system performance by intentionally denying service to (losing) or queuing the calls that require travelling excessive distances, and more importantly, lead to potentially unsatisfactory response times. In an overloaded system and particularly when a service quality rapidly decays with response time, we expect the aggressive denial of immediate response to these relatively high-cost, low-return requests for service to improve the performance and decrease the workload imposed on the system at the same time. Besides potential performance improvements, the partial backup policy offers a more flexible and realistic way to model the complex behavior of a human dispatcher. To see this, suppose that a call arrives when the closest idle ambulance is 40 minutes in travel time away from the call location, but another ambulance that is 5 minutes away is currently busy and expected to complete its job and become available in 5 minutes. Obviously, the full-backup assumption does not reflect the typical decision of a dispatcher in situations like this, which is to queue the call and wait for

the closer ambulance to become free. The partial backup policy, however, will approximately model this aspect of dispatch decision making, by putting an upper limit on the permissible travel distance.

In their excellent review of the recent trends in location, relocation, and dispatching of emergency medical services, Bélanger et al. (2019) highlight the significant potential impact of ambulance relocation and dispatching decisions on the performance of EMS and the close relationship between the two, and call for more research effort in this direction. Furthermore, reviewing the published optimizing models involving dispatch decisions, they remark that the studies performed so far are sometimes too specific and attached to particular contexts to allow a clear and unbiased assessment of the impact of dispatching decisions on system performance. They also notice the shortcoming of current EMS optimization models in reflecting the fairly complex dispatcher behavior in general, and in particular, dispatchers' consideration of remaining time to job completion in making dispatch decisions, and envision more detailed models, perhaps based on simulation, to address these issues. As stated earlier, the model described in Chapter 4 combines the location and dispatch decisions while allowing for priority dependent partial backups, relaxing the unrealistic full backup assumption. Repeated solutions of an optimization model developed around this descriptive model, for different instances of the demand pattern, corresponding to different times during the planing horizon, will result in periodic relocation plans. Dynamic relocation policies can also be generated by solving the static deployment problem for N and then $N-1$, $N-2$, down to one available server, while the costs and constraints associated with each relocation are accounted for in the respective optimization stage. More sophisticated dynamic relocation policies that depend on system states beyond the number of available servers, can also be synthesized by solving the associated optimization model in real-time or by generating a collection of feasible solutions off-line and aggregating them into the compliance tables. In any case, the general approach to obtain static location plans or relocation policies remains exactly the same as those proposed in the literature; however, by incorporating a probabilistic approach in the core optimization model, and relaxing the full backup assumption by replacing the approximate hypercube model of Larson (1975) with the extended version proposed in Chapter 4, and provided that the computational requirements of searching the expanded solution space are met, the resulting location, relocation, and dispatching strategies should be of higher quality and more accurately reflect the operations of the real system. The model proposed in this chapter, on the other hand, which can be viewed as a stripped down version of the one in Chapter 4, allows one to measure the impact of a general family of partial backup dispatching policies on the performance of an ESS with an unspecified dynamic relocation strategy. Therefore, in light of the observations made by Bélanger et al. (2019), these new

descriptive models can be seen as steps in the right direction.

To put ideas in perspective, we recall that in the extension of the hypercube queuing model in Chapter 4, we introduced the queue with partial service and used it to obtain an approximation of the distribution of the number of busy servers in the actual spatially distributed service system with partial backups. We also remember that, in case of the system with a single priority and full backups, that is when any idle server can respond to an incoming call, the classic approximate hypercube model of Larson (1975) is obtained with an M/M/N/0 or M/M/N/∞ queue (depending on whether queues are allowed or not) to approximate the distribution of the number of busy servers. Now, if we keep the focus on the dispatching policy with partial backups, but ignore the exact geometry of the demand distribution, and of the distribution of the locations of the servers responding to those calls, then the queue with partial service will be a good candidate to represent the entire system. This way, not only the distribution of the number of busy servers, but also other performance metrics, such as the probability of loss or queue, and the distributions of the response times and queuing delays can be taken as approximates to their actual counterparts observed for the corresponding ESS. Such an admittedly abstract model, will allow us to study the impact of resource preserving partial backup dispatch policies, such as those considered here, on the performance of the system, without having to resort to detailed application-specific simulation models with potentially high cost of development and conducting the subsequent numerical experiments. In particular, for a given outcome function of an ESS response time, our descriptive model will reveal any opportunities to improve the expected outcome while possibly minimizing the workload imposed on the servers, by following a partial backup strategy in the sense specified shortly. If so desired, the general insight gained from this analysis can then be refined upon through subsequent detailed investigations, perhaps using simulation models or even empirical and field experiments. Alternatively, the results obtained from the model might be accepted and perhaps applied in practice as they are. At any rate, the decision to invest resources in conducting detailed complementary studies and the size and extent of decision spaces explored through such studies will be informed by this preliminary analysis.

Although compared to optimization models, the published work on descriptive models for emergency service systems appear to be few and far between, and mostly limited to extensions and modifications of the hypercube queuing model reviewed in Chapter 4, we wish to draw parallels between the work presented here and of Maxwell et al. (2014) who construct a lower bound on the fraction of late arrivals (defined based on a given response time threshold) in an EMS with a given shift schedule for the ambulances, without knowing how the ambulances will be dispatched or relocated around the city. Although, the objective and the methods used in their study to compute the aforementioned bound, which involves solving a series of integer

programs and then simulating a multi-sever queue, are fundamentally different from what we try to achieve here, the overall theme of trying to answer questions regarding the performance of an emergency system without full knowledge of its important operational details is shared with our present research question, which is "what are the potential performance gains to be had, if any, by introducing partial backup dispatching strategies, without knowing the relocation policies under which the system operates?".

Before proceeding, we give the definition of the Laplace-Stieltjes transform which is used later in this chapter. The Laplace-Stieltjes transform of a real-valued function f is given by the Laplace-Stieltjes integral of the form

$$\int e^{-sx}\, dg(x)$$

for s a complex number. Similar to the regular Laplace transform, we get a slightly different transform depending on the domain of integration. The bilateral Laplace-Stieltjes transform is given by

$$\{\mathcal{L}^* g\}(s) = \int_{-\infty}^{\infty} e^{-sx}\, dg(x)\,,$$

whereas

$$\{\mathcal{L}^* g\}(s) = \lim_{\epsilon \to 0^+} \int_{-\epsilon}^{\infty} e^{-sx}\, dg(x)$$

gives the unilateral transform. Here we will be working with the transforms of the CDFs of waiting times supported on $(0, \infty]$; therefore, the two definitions will lead to identical transforms. The Laplace-Stieljtes transform of a real-values function can be seen as a special case of the Laplace transform; we have

$$\mathcal{L}^* g = \mathcal{L}(dg).$$

The Laplace-Stieltjes transform appears naturally in applied and theoretical probability. In fact, if X is a random variable with F as its CDF, then the Laplace-Stieltjes transform of F is given by the expectation of the random variable e^{-X}; that is

$$\{\mathcal{L}^* F\}(s) = \mathrm{E}\left[e^{-sX}\right]\,,$$

which can also be recognized as the moment generating function of the random variable $Y = -X$.

We now present the model and then two example applications.

6.1 Mathematical Model

In this section, we give the basic definitions and underlying assumptions first, and then derive all the required relations that will eventually merge into an iterative algorithm outlined in the end.

We assume that calls arrive as a Poisson stream of rate λ originating from uniformly distributed random locations within a service region of total area A. The location of each of the N servers is also assumed to be uniformly and independently distributed in the service region independent of the state of the system.

Let Π_ξ^κ designate a partial backup dispatching policy in which the κ closest servers to the location of an incoming call or those that are less than ξ away from the call location are considered eligible for dispatch. Under this policy, the closest free and eligible server is always dispatched if one exists and, depending on the queue discipline, the call is queued or lost if all eligible servers are busy at the time of call arrival. Moreover, the call is lost to both the queuing and loss system if no eligible servers are found. The policy for the loss system will be denoted by $\bar{\Pi}_\xi^\kappa$. We will also consider two special cases of this policy as well. In the special case with $\xi = 0$, that is Π_0^κ, the closest κ response units to a call location will be considered for dispatch; therefore, we must have $\kappa \geq 1$ for this policy to be meaningful in practice. In the special case Π_ξ^0, on the other hand, there are no guaranteed backups and only the response units that happen to be within distance ξ from the call location upon arrival will be dispatch candidates. In this case, we must have $\xi \geq 0$ and set to a reasonably large value for the policy to be meaningful in practice.

6.1.1 Distribution of the number of busy servers

Let C be the random number of servers, busy or available, residing within distance ξ of the random call location. The probability of exactly c servers being within distance ξ from the call location, denoted by ζ_c, might be obtained from a binomial distribution as

$$\zeta_c = \Pr\{\mathrm{C} = c\} = \frac{N!}{(N-c)!c!}(\frac{\pi\xi^2}{A})^c(1 - \frac{\pi\xi^2}{A})^{N-c}, \quad c = 0, \cdots, N, \tag{6.1}$$

with every π replaced by 2 for the Manhattan metric; however, as discussed earlier, equation (6.1) does not take the boundary effects into account. We can obtain an edge-corrected ζ_c by first recognizing that

$$\Pr\{\mathrm{C} = 0\} = 1 - \Pr\{\mathrm{C} > 0\} = 1 - \Pr\{\mathrm{R}_1 \leq \xi\}\,,$$

$$\Pr\{\mathrm{C} = N\} = \Pr\{\mathrm{R}_N \le \xi\}\,,$$
$$\Pr\{\mathrm{C} = c\} = \Pr\{\mathrm{C} \ge c\} - \Pr\{\mathrm{C} \ge c+1\} = \Pr\{\mathrm{R}_c \le \xi\} - \Pr\{\mathrm{R}_{c+1} \le \xi\}\,,$$

and then arriving at

$$\zeta_c = \begin{cases} \tilde{F}_{\mathrm{R}_c}(\xi) - \tilde{F}_{\mathrm{R}_{c+1}}(\xi) & c = 1, \cdots, N-1, \\ 1 - \tilde{F}_{\mathrm{R}_1}(\xi) & c = 0, \\ \tilde{F}_{\mathrm{R}_N}(\xi) & c = N, \end{cases} \tag{6.2}$$

where $\tilde{F}_{\mathrm{R}_n}(r) = \Pr\{\mathrm{R}_n \le r\}$ is the CDF of the edge-corrected distance to the n-th neighbor we obtained as (5.7) for the Euclidean and as (5.10) and (5.11) for the Manhattan metric. Now, if we let ζ'_c denote the probability of $\{\mathrm{C} = c\}$ under a policy that guarantees at least κ covering servers, we have

$$\zeta'_c = \begin{cases} 0 & c < \kappa - 1\,, \\ \sum_{c'=0}^{\kappa} \zeta_{c'} & c = \kappa\,, \\ \zeta_c & c > \kappa\,, \end{cases}$$

from which the demand vector $[\lambda_c]$ immediately follows as

$$\lambda_c = \lambda \zeta'_c\,, \quad c = 1, \cdots, N\,.$$

We note that $\zeta_c = \zeta'_c$ if $\kappa = 0$ and that $\lambda = \sum_{c=1}^{N} \lambda_c$ only if $\kappa \ge 1$.

Finally, the distribution of the number of busy servers follows from the results we have for the queue or loss system with partial service with N servers and the demand vector $[\lambda_c]$. More specifically, P_n the probability of $n = 0, \cdots, N$ servers being busy is given by (4.1) and (4.2) for the system with queues and by (4.3) and (4.4) for the loss system.

6.1.2 Probability of Loss and Queue

Let P_{loss} be the probability of a random arrival leaving the system without receiving service. Fo r the customers who do receive service, let P_{q} be the probability of entering the waiting queue and hence experiencing a non-zero queuing delay, that is $P_{\mathrm{q}} = \Pr\{\mathrm{W}_{\mathrm{q}} > 0\}$ where W_{q} is the random queuing delay experienced by customers receiving service. We then obtain P_{q} and P_{loss} as follows.

Loss System

For the loss system we have

$$P_{\mathrm{q}} = 0\,,$$

by definition.

An arrival to a loss system will be lost if there are no free compatible servers to attend to the call; otherwise, one of the free compatible servers will be dispatched immediately. Conditioning on the number of covering and then busy servers, the loss probability P_{loss} under Π_{ξ}^{κ} is obtained as

$$P_{\mathrm{loss}} = \sum_{c=0}^{N} \zeta'_c \sum_{n=c}^{N} \frac{\binom{n}{c}}{\binom{N}{c}} P_n = \sum_{c=0}^{N} \zeta'_c \sum_{n=c}^{N} \frac{n!(N-c)!}{N!(n-c)!}\, P_n\,. \tag{6.3}$$

special case Π_{ξ}^{0}

In this case, it also holds that

$$P_{\mathrm{loss}} = \sum_{c=0}^{N} \zeta_c \sum_{n=c}^{N} \frac{n!(N-c)!}{N!(n-c)!}\, P_n\,,$$

since $\zeta'_c = \zeta_c$. The loss probability thus becomes

$$P_{\mathrm{loss}} = \sum_{n=\kappa}^{N} \frac{(N-\kappa)!n!}{N!(n-\kappa)!} P_n\,,$$

special case Π_{0}^{κ}

In this case, (6.3) simplifies to

$$P_{\mathrm{loss}} = \sum_{n=\kappa}^{N} \frac{n!(N-\kappa)!}{N!(n-\kappa)!}\, P_n\,,$$

since $\zeta'_{\kappa} = 1$ and $\zeta'_c = 0$ for $c \neq \kappa$.

Queueing System

In the system where queues are allowed, any customer that, upon arrival, finds at least one compatible server will eventually enter service either immediately or incurring a queuing

delay. Hence, the loss probability simply is

$$P_{\text{loss}} = \zeta'_0 \,,$$

and with probability $1 - P_{\text{loss}}$ a customer will enter service. The probability of getting queued for such a customer is then

$$P_{\text{q}} = \frac{1}{1 - P_{\text{loss}}} \sum_{c=1}^{N} \zeta'_c \sum_{n=c}^{N} \frac{\binom{N-c}{n-c}}{\binom{N}{n}} P_n = \frac{1}{1 - \zeta'_0} \sum_{c=1}^{N} \zeta'_c \sum_{n=c}^{N} \frac{(N-c)!n!}{(n-c)!N!} P_n \,. \tag{6.4}$$

We remember that P_{q} represents the fraction of queued customers to all *served* customers; this is reflected in the summation in (6.4) starting from $c = 1$ instead of $c = 0$ excluding the lost arrivals, and is accounted for via the division by $1 - P_{\text{loss}}$.

special case $\bar{\Pi}^0_\xi$

Again, since $\zeta'_c = \zeta_c$, we might use

$$P_{\text{q}} = \frac{1}{1 - \zeta_0} \sum_{c=1}^{N} \zeta_c \sum_{n=c}^{N} \frac{(N-c)!n!}{(n-c)!N!} P_n \,. \tag{6.5}$$

special case $\bar{\Pi}^\kappa_0$

The queuing probability becomes

$$P_{\text{q}} = \sum_{n=\kappa}^{N} P_n \frac{(N-\kappa)!n!}{N!(n-\kappa)!} \tag{6.6}$$

because $\zeta'_\kappa = 1$ and $\zeta'_c = 0$ for $c \neq \kappa$, and in particular $\zeta'_0 = 0$ since $\kappa \neq 0$ (Note $\kappa \geq 1$).

6.1.3 Queuing Delay

In this section, we characterize the delay incured by the customers who enter the waiting queue in a system where queues are allowed. In Chapter 3, we derived the distribution of the number of busy servers in a stable queuing system with partial service (for which $N\mu > \sum_{c=1}^{N} \lambda_c$). We have the following result for the waiting times and the number of waiting customers.

Theorem 3. *For class c customers, the Laplace-Stieltjes transform of the waiting time* W_c

and the generating function of the queue length L_c *will be*

$$\mathrm{E}\left[e^{-s\mathrm{W}_c}\right] = 1 - \sum_{n=c}^{N} P_n \frac{(N-c)!}{N!} \left(\frac{n!}{(n-c)!} - c \sum_{j=c}^{n} \frac{(j-1)!}{(j-c)!} \prod_{k=j}^{n} \frac{k\mu - \frac{k!}{N!}\sum_{c'=1}^{k} \lambda_{c'} \frac{(N-c')!}{(k-c')!}}{k\mu - \frac{k!}{N!}\sum_{c'=1}^{k} \lambda_{c'} \frac{(N-c')!}{(k-c')!} + s} \right)$$

and

$$\mathrm{E}[z^{\mathrm{L}_c}] = 1 - \sum_{n=c}^{N} P_n \left(1 - \prod_{j=c}^{n} \frac{j\mu - \sum_{c'=1}^{j} \lambda_{c'} \frac{j!(N-c')!}{N!(j-c')!}}{j\mu - \sum_{c'=1}^{j} \lambda_{c'} \frac{j!(N-c')!}{N!(j-c')!} + \lambda_c \frac{j!(N-c)!}{N!(j-c)!}(1-z)} \right),$$

with the corresponding expected values as

$$\mathrm{E}[\mathrm{W}_c] = \sum_{n=c}^{N} P_n \frac{(N-c)!}{N!} \sum_{j=c}^{n} \frac{j!/(j-c)!}{j\mu - (j!/N!)\sum_{c'=1}^{j} \lambda_{c'} \frac{(N-c')!}{(j-c')!}},$$

and

$$\mathrm{E}[\mathrm{L}_c] = \lambda_c \sum_{n=c}^{N} P_n \frac{(N-c)!}{N!} \sum_{j=c}^{n} \frac{j!/(j-c)!}{j\mu - (j!/N!)\sum_{c'=1}^{j} \lambda_{c'} \frac{(N-c')!}{(j-c')!}};$$

moreover, for the waiting time W *and queue length* L *of all customers we have*

$$\mathrm{E}[e^{-sW}] = \sum_{c=1}^{N} \frac{\lambda_c}{\lambda} \mathrm{E}[e^{-s\mathrm{W}_c}],$$

$$\mathrm{E}[z^L] = 1 - \sum_{n=1}^{N} P_n \left(1 - \prod_{j=1}^{n} \frac{j\mu - \sum_{c'=1}^{j} \lambda_{c'} \frac{j!(N-c')!}{N!(j-c')!}}{j\mu - \sum_{c'=1}^{j} \lambda_{c'} \frac{j!(N-c')!}{N!(j-c')!} + \lambda_c \frac{j!(N-c)!}{N!(j-c)!}(1-z)} \right),$$

$$\mathrm{E}[W] = \sum_{c=1}^{N} \frac{\lambda_c}{\lambda} \mathrm{E}[\mathrm{W}_c] = \sum_{c=1}^{N} \frac{\lambda_c}{\lambda} \sum_{n=c}^{N} P_n \frac{(N-c)!}{N!} \sum_{j=c}^{n} \frac{j!/(j-c)!}{j\mu - (j!/N!)\sum_{c'=1}^{j} \lambda_{c'} \frac{(N-c')!}{(j-c')!}},$$

$$\mathrm{E}[L] = \lambda \mathrm{E}[W].$$

Proof. Given the setup described in the proof of 1, we now proceed to find an expression for W_c the distribution of the waiting time of class c customers of the queue with partial service. We first note that this distribution is independent of the identities of the corresponding compatible servers since they are indistinguishable; that is $\mathrm{E}[z^{\mathrm{W}_c}] = \mathrm{E}[z^{W_\kappa}] = \mathrm{E}[z^{W_{\kappa'}}]$ for all $\kappa, \kappa' \in \mathcal{C}^c$ with $\mathcal{C}^c$ the class of customers that are compatible with exactly c servers. We can therefore specialize the product form

$$E[e^{-sW_\kappa}] = \sum_{n=0}^{N} \sum_{(\mathrm{M}_1, \cdots, \mathrm{M}_n) \in \mathcal{P}^n} \Pr(\mathrm{M}_1, \cdots, \mathrm{M}_n) \prod_{\substack{j=1 \\ \kappa \in \mathcal{U}(\{\mathrm{M}_1, \cdots, \mathrm{M}_j\})}}^{n} \frac{\mu_{\{\mathrm{M}_1, \cdots, \mathrm{M}_j\}} - \lambda_{\mathcal{U}(\{\mathrm{M}_1, \cdots, \mathrm{M}_j\})}}{\mu_{\{\mathrm{M}_1, \cdots, \mathrm{M}_j\}} - \lambda_{\mathcal{U}(\{\mathrm{M}_1, \cdots, \mathrm{M}_j\})} + s}. \tag{6.7}$$

for the queue with partial service to obtain the distribution of W_c in transform.

We first acknowledge that a class c customer observing the sequence of busy servers $(\mathrm{M}_1, \cdots, \mathrm{M}_n)$ upon arrival, will eventually be handled by the j-th server $(c \le j \le n)$ in the sequence with probability $\binom{j-1}{c-1}/\binom{N}{c}$; moreover, this customer will be compatible with at least one of the idle servers and hence experience zero waiting time with probability $1 - \binom{n}{c}/\binom{N}{c}$ if $c \le n$ (Note that $\sum_{j=c}^{n}\binom{j-1}{c-1} = \binom{n}{c}$). Therefore, conditioning on the receiving server j and substituting (A.4) and (A.5) in (6.7) implies

$$\mathrm{E}\left[e^{-s\mathrm{W}_c}|(\mathrm{M}_1, \cdots, \mathrm{M}_n), c \le n\right] = 1 - \frac{\binom{n}{c}}{\binom{N}{c}} + \sum_{j=c}^{n}\frac{\binom{j-1}{c-1}}{\binom{N}{c}}\prod_{k=j}^{n}\frac{k\mu - \sum_{c'=1}^{k}\lambda_{c'}\frac{\binom{k}{c'}}{\binom{N}{c'}}}{k\mu - \sum_{c'=1}^{k}\lambda_{c'}\frac{\binom{k}{c'}}{\binom{N}{c'}} + s},$$

which naturally (for a system with indistinguishable servers) depends only on the number of busy servers. Using $\mathrm{E}\left[e^{-s\mathrm{W}_c}|(\mathrm{M}_1, \cdots, \mathrm{M}_n), c > n\right] = 1$, summing over all possible $(\mathrm{M}_1, \cdots, \mathrm{M}_n) \in \mathcal{P}^n$ and simplifying yields

$$\mathrm{E}\left[e^{-s\mathrm{W}_c}\right] = 1 - \sum_{n=c}^{N} P_n \frac{(N-c)!}{N!}\left(\frac{n!}{(n-c)!} - c\sum_{j=c}^{n}\frac{(j-1)!}{(j-c)!}\prod_{k=j}^{n}\frac{k\mu - \frac{k!}{N!}\sum_{c'=1}^{k}\lambda_{c'}\frac{(N-c')!}{(k-c')!}}{k\mu - \frac{k!}{N!}\sum_{c'=1}^{k}\lambda_{c'}\frac{(N-c')!}{(k-c')!} + s}\right), \tag{6.8}$$

where P_n is given by (A.6). The expected waiting time of a class c customer is then obtained as

$$\mathrm{E}[\mathrm{W}_c] = -\left.\frac{d\mathrm{E}[e^{-s\mathrm{W}_c}]}{ds}\right|_{s=0} = \sum_{n=c}^{N} P_n \frac{(N-c)!}{N!}\sum_{j=c}^{n}\frac{j!/(j-c)!}{j\mu - (j!/N!)\sum_{c'=1}^{j}\lambda_{c'}\frac{(N-c')!}{(j-c')!}}. \tag{6.9}$$

We now calculate the distribution of the number of class c waiting customers in the queue with partial service. Letting L_κ be the random number of class κ customers in the skill-based queue we first note that the product form (6.7) has been obtained by first computing $\mathrm{E}[z^{L_\kappa}]$ and then using the relation

$$\mathrm{E}[z^{L_\kappa}] = \mathrm{E}[e^{-\lambda_\kappa W_\kappa(1-z)}], \tag{6.10}$$

which is a distributional form of Little's law for Poisson arrivals obtained by Keilson and Servi (1988) applied to class κ customers in the skill based queue discussed above (See Haji and Newell (1971) for the more general form). Unlike class κ customers in the skill based queue, class c customers in the queue with partial service do not satisfy the required FIFO condition since overtaking may happen within the class (A class c customer who enters service with a server will overtake all earlier class c waiting customers who are not compatible with that particular server). Thus, the distributional Little's law does not hold and hence we

cannot use an expression similar to (6.10) to obtain $\mathrm{E}[z^{\mathrm{L}_c}]$ by replacing $s = \lambda_c(1-z)$ in (6.8); instead, we directly derive this distribution following arguments similar to those used by Visschers et al. (2012) who make two observations: first, the product form stationary distribution implies a geometric distribution with parameter α_j for N_j the random number of customers waiting between servers M_j and M_{j+1}. Second, with independent Poisson arrivals and if $\kappa \in \mathcal{U}(\{\mathrm{M}_1, \cdots, \mathrm{M}_j\})$, the distribution of $\mathrm{N}_{\kappa,j}$ the number of class κ customers waiting between servers M_j and M_{j+1} conditional on N_j is binomially distributed with parameters N_j and $\theta_j = \lambda_\kappa / \lambda_{\mathcal{U}(\{\mathrm{M}_1,\cdots,\mathrm{M}_j\})}$. It is then shown that $\mathrm{N}_{\kappa,j}$ will be geometrically distributed with parameter

$$\eta_{\kappa,j} = \frac{\alpha_j \theta_j}{1 - \alpha_j(1-\theta_j)} = \frac{\lambda_\kappa}{\mu_{\{\mathrm{M}_1,\cdots,\mathrm{M}_j\}} - \lambda_{\mathcal{U}(\{\mathrm{M}_1,\cdots,\mathrm{M}_j\})} + \lambda_\kappa},$$

which, combined with the fact that the N_j and thus the $\mathrm{N}_{\kappa,j}$ are mutually independent leads to

$$\mathrm{E}[z^{L_\kappa} | (\mathrm{M}_1, \cdots, \mathrm{M}_i)] = \prod_{\substack{j=1 \\ \kappa \in \mathcal{U}(\{\mathrm{M}_1,\cdots,\mathrm{M}_j\})}}^{n} \frac{1-\eta_{c,j}}{1-\eta_{c,j} z}.$$

Replacing class κ with class c and noting that for $j \geq c$ a fraction $\binom{j}{c} / \binom{N}{c}$ of total class c arrivals will belong to $\mathcal{U}(\{\mathrm{M}_1, \cdots, \mathrm{M}_j\})$, we can argue following a similar line of reasoning that $\mathrm{N}_{c,j}$ will also be geometrically distributed with parameter

$$\tilde{\eta}_{c,j} = \frac{\tilde{\alpha}_j \tilde{\theta}_j}{1 - \tilde{\alpha}_j(1-\tilde{\theta}_j)},$$

with

$$\tilde{\theta}_j = \frac{\lambda_c \binom{j}{c} / \binom{N}{c}}{\tilde{\lambda}_{\mathcal{U}(\{\mathrm{M}_1,\cdots,\mathrm{M}_j\})}}.$$

Furthermore, from the mutual independence of $\mathrm{N}_{c,j}$, $j = 1, \cdots, N$, and noting that there will be no class c customers waiting between servers M_{k-1} and M_k for $k = 2, \cdots, c$, we conclude

$$\mathrm{E}[z^{\mathrm{L}_c} | (\mathrm{M}_1, \cdots, \mathrm{M}_n)] = \prod_{j=c}^{n} \frac{1-\tilde{\eta}_{c,j}}{1-\tilde{\eta}_{c,j} z}.$$

De-conditioning using the stationary distribution P_n and summing over all possible $(\mathrm{M}_1, \cdots, \mathrm{M}_n)$ we have

$$\mathrm{E}[z^{\mathrm{L}_c}] = \sum_{n=0}^{c-1} P_n + \sum_{n=c}^{N} P_n \prod_{j=c}^{n} \frac{1-\tilde{\eta}_{c,j}}{1-\tilde{\eta}_{c,j} z},$$

which after expanding the expression for $\tilde{\eta}_{c,j}$ and simplifying becomes

$$\mathrm{E}[z^{\mathrm{L}_c}] = 1 - \sum_{n=c}^{N} P_n \left(1 - \prod_{j=c}^{n} \frac{j\mu - \sum_{c'=1}^{j} \lambda_{c'} \frac{j!(N-c')!}{N!(j-c')!}}{j\mu - \sum_{c'=1}^{j} \lambda_{c'} \frac{j!(N-c')!}{N!(j-c')!} + \lambda_c \frac{j!(N-c)!}{N!(j-c)!}(1-z)} \right) .$$

One can then verify that the average queue length of class c customers will be

$$\mathrm{E}[\mathrm{L}_c] = \left. \frac{d\mathrm{E}[z^{\mathrm{L}_c}]}{dz} \right|_{z=1} = \lambda_c \sum_{n=c}^{N} P_n \frac{(N-c)!}{N!} \sum_{j=c}^{n} \frac{j!/(j-c)!}{j\mu - (j!/N!) \sum_{c'=1}^{j} \lambda_{c'} \frac{(N-c')!}{(j-c')!}} . \tag{6.13}$$

□

Remark 1. *Equation* (6.13) *can also be obtained from* (6.9) *and Little's law for class c customers* $\mathrm{E}[\mathrm{L}_c] = \lambda_c \mathrm{E}[\mathrm{W}_c]$.

We have the following corollary for the waiting time of customers who actually experience a nonzero queuing delay (delayed dispatches).

Corollary 1. *The delay experienced by class c customers that enter the queue in a queuing system with partial service, that is* $\mathrm{W}_c^D = \mathrm{W}_c | \mathrm{W}_c > 0$*, satisfies*

$$\mathrm{E}\left[e^{-s\mathrm{W}_c^D}\right] = \frac{c \sum_{n=c}^{N} P_n \sum_{j=c}^{n} \frac{(j-1)!}{(j-c)!} \prod_{k=j}^{n} \frac{k\mu - \frac{k!}{N!} \sum_{c'=1}^{k} \lambda_{c'} \frac{(N-c')!}{(k-c')!}}{k\mu - \frac{k!}{N!} \sum_{c'=1}^{k} \lambda_{c'} \frac{(N-c')!}{(k-c')!} + s}}{\sum_{n=c}^{N} \frac{n!}{(n-c)!} P_n} , \tag{6.14}$$

with the expected value

$$\mathrm{E}[\mathrm{W}_c^D] = \frac{\sum_{n=c}^{N} P_n \sum_{j=c}^{n} \frac{j!/(j-c)!}{j\mu - (j!/N!) \sum_{c'=1}^{j} \lambda_{c'} \frac{(N-c')!}{(j-c')!}}}{\sum_{n=c}^{N} \frac{n!}{(n-c)!} P_n} .$$

The form of the Laplace transform of W_c^D in (6.14) implies a mixture of sums of independent exponentially distributed random variables. Now for n exponentially distributed random variables X_i , $i = 1, \cdots, n$, with rate parameters μ_i, the PDF of the sum $S_n = \sum_{i=1}^{n} X_n$ is given by

$$f_{S_n}(x) = \sum_{i=1}^{n} f_i(x) \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{\mu_j}{\mu_j - \mu_i} , \tag{6.15}$$

with $f_i(x) = \mu e^{-\mu x}$ the PDF of the exponential distribution with rate parameter μ. For this and other results related to the convolution of exponential distributions see Bibinger (2013), Sen and Balakrishnan (1999), Lomonosov (1974), Kordecki (1997), Johnson et al. (1994),

Jasiulewicz and Kordecki (2003). With (6.15) and (6.14) we get the following result for the PDF of W_c^D.

Corollary 2. *The PDF of the delay experienced by class c customers that enter the queue in a queuing system with partial service is given by*

$$f_{\mathrm{W}_c^D}(x) = \frac{c\sum_{n=c}^{N} P_n \sum_{j=c}^{n} \frac{(j-1)!}{(j-c)!} \sum_{k=j}^{n} \mu_k e^{-\mu_k x} \prod_{\substack{m=j \\ m\neq k}}^{n} \frac{\mu_m}{\mu_m - \mu_k}}{\sum_{n=c}^{N} \frac{n!}{(n-c)!} P_n}\,, \tag{6.16}$$

with

$$\mu_k = k\mu - \frac{k!}{N!} \sum_{c'=1}^{k} \lambda_{c'} \frac{(N-c')!}{(k-c')!}\,. \tag{6.17}$$

To obtain the PDF of W^D the queuing delay for *all served calls*, we let $P_{q,c}$ be the probability of a class c customer getting queued and recognize that

$$f_{\mathrm{W}^D}(x) = \frac{\sum_{c=1}^{N} \lambda_c P_{q,c} f_{\mathrm{W}_c^D}(x)}{\sum_{c=1}^{N} \lambda_c P_{q,c}}\,,$$

where

$$P_{q,c} = \sum_{n=c}^{N} P_n \frac{n!(N-c)!}{N!(n-c)!}\,.$$

We thus obtain $f_{\mathrm{W}^D}(x)$ as the following.

Corollary 3. *The PDF of the delay experienced by the queued customers in a queuing system with partial service is given by*

$$f_{\mathrm{W}^D}(x) = \frac{\sum_{c=1}^{N} \frac{c(N-c)!}{N!} \lambda_c \sum_{n=c}^{N} P_n \sum_{j=c}^{n} \frac{(j-1)!}{(j-c)!} \sum_{k=j}^{n} \mu_k e^{-\mu_k x} \prod_{\substack{m=j \\ m\neq k}}^{n} \frac{\mu_m}{\mu_m - \mu_k}}{\sum_{c=1}^{N} \lambda_c \sum_{n=c}^{N} \frac{n!(N-c)!}{N!(n-c)!} P_n}\,,$$

with μ_m from (6.17).

Now, the above corollary holds for the general queuing system with partial service; for the ESS considered in this chapter, for which $\lambda_c = \lambda\zeta_c'$ and P_{q} as in (6.4) (or (6.5) and (6.6) in the special cases), we get

$$f_{\mathrm{W}^D}(x) = \frac{\sum_{c=1}^{N} \frac{c(N-c)!}{N!} \zeta_c' \sum_{n=c}^{N} P_n \sum_{j=c}^{n} \frac{(j-1)!}{(j-c)!} \sum_{k=j}^{n} \mu_k e^{-\mu_k x} \prod_{\substack{m=j \\ m\neq k}}^{n} \frac{\mu_m}{\mu_m - \mu_k}}{P_{\mathrm{q}}}\,,$$

with

$$\mu_k = k\mu - \lambda \frac{k!}{N!} \sum_{c'=1}^{k} \zeta'_{c'} \frac{(N-c')!}{(k-c')!} \,.$$

This PDF will enable us to compute the convolution of W^D with other relevant random time elements to characterize the response times of the delayed dispatches.

6.1.4 Dispatch Distance

In this section, we obtain the moments and distribution of D_Π the random travel distance for a dispatch assignment under the partial backup dispatch policy Π. In the next section, we will use these results together with a suitable travel time model to determine the properties of the random travel time that a server takes to arrive at the call location.

In the previous chapter, we derived expressions for the distribution of the distance to the k-th neighbor with boundary-effects and under some simplifying assumptions. However, these edge-corrected results are way too complicated to be used in the proceeding analysis and as stated before, our attempts to derive simple and tractable approximations to edge-corrected distance distributions has not yet concluded; therefore, although we used the edge-corrected results of Chapter 5 in computing ζ_c and consequently every other metric dependent on ζ_c such as P_{q}, P_{loss}, and $f_{\mathrm{W}^D}(x)$, we shall temporarily resort to the nearest neighbor distance distribution obtained from the PPP in deriving the properties of the random dispatch distance in the present subsection.

Assuming that entities are scattered on an infinitely large area according to a PPP with an intensity of σ points per unit area, the PDF of the Euclidean distance to the n-th neighbor from an arbitrary origin is given as

$$f_{\mathrm{R}_n}(r) = \frac{2(\pi\sigma)^n}{(n-1)!} r^{2n-1} e^{-\pi\sigma r^2}, \quad r \geq 0,\ n = 1, 2, \cdots, \tag{6.18}$$

with the corresponding CDF

$$F_{\mathrm{R}_n}(r) = \frac{\gamma(n, \pi\sigma r^2)}{(n-1)!} \,, \tag{6.19}$$

and the l-th moment

$$\mathrm{E}[{\mathrm{R}_n}^l] = \frac{\Gamma(n + \frac{l}{2})}{(\pi\sigma)^{\frac{l}{2}}(n-1)!} \,. \tag{6.20}$$

For the Manhattan distance metric, (6.18), (6.19), and (6.20) hold with every π replaced by 2.

We will also need Δ_ξ^l the l-th moment of the distance from a uniformly distributed random

point within distance ξ from an arbitrary origin. From (6.18) we have

$$\Delta_\xi^l = \frac{2}{l+2}\xi^l, \quad \xi \geq 0,\ l = 0, 1, \cdots,$$

for both Euclidean and Manhattan metrics.

We now proceed to obtain the moments and distribution of D_Π the random dispatch distance under different queue disciplines and partial backup dispatch policies.

Loss System

To obtain the l-th raw moment of the expected dispatch distance, we need to condition on the distance rank M of the closest free server. The general relation for $\kappa \in \{0, 1, \cdots, N\}$ and $\xi \geq 0$ is

$$\mathrm{E}[\{\mathrm{D}_{\Pi_\xi^\kappa}\}^l] = \frac{1}{P'_{\text{serv}}}\left[\sum_{m=1}^{\kappa} \Pr\{\mathrm{M}=m\}\mathrm{E}[\mathrm{R}_m{}^l] + \sum_{m=\kappa+1}^{N} \Pr\{\mathrm{M}=m\}\int_0^\xi f_{\mathrm{R}_m}(r)r^l dr\right], \quad (6.21)$$

where

$$P'_{\text{serv}} = \sum_{m=1}^{\kappa} \Pr\{\mathrm{M}=m\} + \sum_{m=\kappa+1}^{N} \Pr\{\mathrm{M}=m\}F_{\mathrm{R}_m}(\xi)$$

gives the probability of an arrival receiving service from the system. In general, we should have $P_{\text{serv}} = 1 - P_{\text{loss}}$; however, this ceases to be the case as we proceed to replace $F_{\mathrm{R}_m}(\xi)$ with the expression obtained from the spacial Poisson process assumption. This is because of the fact that the special Poisson assumption gives us an approximation of $F_{\mathrm{R}_m}(\xi)$, whereas P_{loss} is exact under our basic assumptions. Therefore, to keep the analysis consistent and minimize the approximation errors, instead of $P_{\text{serv}} = 1 - P_{\text{loss}}$, we normalize expressions like (6.21) with P'_{serv} (instead of $P_{\text{serv}} = 1 - P_{\text{loss}}$) which is also obtained in terms of approximate expressions $F_{\mathrm{R}_m}(\xi)$ and $f_{\mathrm{R}_m}(\xi)$.

We use the shorthand notation $\Psi_{m,N}$ to represent the distribution of M as

$$\Psi_{m,N} \equiv \Pr\{\mathrm{M}=m\} = \sum_{n=m-1}^{N-1} \frac{\binom{N-m}{n-m+1}}{\binom{N}{n}} P_n = \sum_{n=m-1}^{N-1} \frac{n!(N-m)!(N-n)}{N!(n-m+1)!} P_n \quad m = 0, 1, \cdots, N.$$

We obtain the integral in (6.21) as

$$\int_0^\xi f_{\mathrm{R}_m}(r)r^l dr = \frac{2(\pi\sigma)^m}{(m-1)!}\int_0^\xi r^{2m-1}e^{-\pi\sigma r^2}dr = \frac{\gamma(m+\frac{l}{2}, \pi\sigma\xi^2)}{(m-1)!(\pi\sigma)^{l/2}},$$

where $\gamma(s,t) = \int_0^t x^{s-1}e^{-x}dx$ is the lower incomplete gamma function. We thus get

$$\mathrm{E}[\{\mathrm{D}_{\Pi_\xi^\kappa}\}^l] = \frac{1}{P'_{\text{serv}}}\left[\sum_{m=1}^{\kappa}\frac{\Gamma(m+\frac{l}{2})}{(\pi\sigma)^{l/2}(m-1)!}\Psi_{m,N} + \sum_{m=\kappa+1}^{N}\frac{\gamma(m+\frac{l}{2},\,\pi\sigma\xi^2)}{(\pi\sigma)^{l/2}(m-1)!}\Psi_{m,N}\right]$$

with

$$P'_{\text{serv}} = \sum_{m=1}^{\kappa}\Psi_{m,N} + \sum_{m=1}^{N}\frac{\gamma(m,\pi\sigma\xi^2)}{(m-1)!}\Psi_{m,N}\,,$$

where $\Gamma(x)$ is the gamma function. For $F_{\mathrm{D}_{\Pi_\xi^\kappa}}(x)$ the CDF of the dispatch distance, we write

$$\begin{aligned}F_{\mathrm{D}_{\Pi_\xi^\kappa}}(x) = \Pr\{\mathrm{D}_{\Pi_\xi^\kappa}\le x\} &= \frac{1}{P'_{\text{serv}}}\left\{\sum_{m=1}^{\kappa}\Pr\{\mathrm{M}=m\}\int_0^x f_{\mathrm{R}_m}(r)dr + \right.\\ &\qquad \left.\sum_{m=\kappa+1}^{N}\Pr\{\mathrm{M}=m\}\int_0^{\min(x,\xi)} f_{\mathrm{R}_m}(r)dr\right\}\\ &= \frac{1}{P'_{\text{serv}}}\left\{\sum_{m=1}^{\kappa}\frac{\gamma(m,\pi\sigma x^2)}{(m-1)!}\Psi_{m,N} + \sum_{m=\kappa+1}^{N}\frac{\gamma(m,\pi\sigma\min(x,\xi)^2)}{(m-1)!}\Psi_{m,N}\right\}.\end{aligned}$$

The PDF is then obtained as

$$f_{\mathrm{D}_{\Pi_\xi^\kappa}}(x) = \frac{1}{P'_{\text{serv}}}\left\{\sum_{m=1}^{\kappa}\frac{2(\pi\sigma)^m x^{2m-1}e^{-\pi\sigma x^2}}{(m-1)!}\Psi_{m,N} + \mathbf{1}(x<\xi)\sum_{m=\kappa+1}^{N}\frac{2(\pi\sigma)^m x^{2m-1}e^{-\pi\sigma x^2}}{(m-1)!}\Psi_{m,N}\right\}.$$

Special Case Π_ξ^0:

The moments and the distribution of the dispatch distance for this policy are given by

$$\mathrm{E}[\{\mathrm{D}_{\Pi_\xi^0}\}^l] = \frac{1}{P'_{\text{serv}}}\sum_{m=1}^{N}\frac{\gamma(m+\frac{l}{2},\,\pi\sigma\xi^2)}{(\pi\sigma)^{l/2}(m-1)!}\Psi_{m,N}\,, \tag{6.22}$$

$$F_{\mathrm{D}_{\Pi_\xi^0}}(x) = \frac{1}{P'_{\text{serv}}}\sum_{m=1}^{N}\frac{\gamma(m,\pi\sigma\min(x,\xi)^2)}{(m-1)!}\Psi_{m,N}\,, \tag{6.23}$$

and

$$f_{\mathrm{D}_{\Pi_\xi^0}}(x) = \frac{\mathbf{1}(x<\xi)}{P'_{\text{serv}}}\sum_{m=1}^{N}\frac{2(\pi\sigma)^m x^{2m-1}e^{-\pi\sigma x^2}}{(m-1)!}\Psi_{m,N}\,, \tag{6.24}$$

with

$$P'_{\text{serv}} = \sum_{m=1}^{N}\frac{\gamma(m,\pi\sigma\xi^2)}{(m-1)!}\Psi_{m,N}\,.$$

Alternatively, we recognize that conditional on n servers busy, the dispatch distance is dis-

tributed as the distance to the nearest server out of a total of $N-n$ free servers; with the Poisson assumption, this means a density of $\sigma = \frac{N-n}{A}$. Therefore, we can write

$$\mathrm{E}[\{\mathrm{D}_{\Pi_\xi^0}\}^l] = \frac{1}{P''_{\mathrm{serv}}} \sum_{n=0}^{N-1} P_n \frac{\gamma(1+\frac{l}{2},\, \pi\frac{N-n}{A}\xi^2)}{(\pi\frac{N-n}{A})^{l/2}}$$

with

$$P''_{\mathrm{serv}} = \sum_{n=0}^{N-1} P_n \gamma(1,\, \frac{\pi\xi^2(N-n)}{A}) \,.$$

Similarly, we obtain for the CDF and PDF as

$$F_{\mathrm{D}_{\Pi_\xi^0}}(x) = \frac{1}{P''_{\mathrm{serv}}} \sum_{n=0}^{N-1} P_n \gamma(1,\, \frac{\pi \min(x,\xi)^2(N-n)}{A})$$

and

$$f_{\mathrm{D}_{\Pi_\xi^0}}(x) = \frac{\mathbf{1}(x<\xi)}{P''_{\mathrm{serv}}} \left\{ \sum_{n=0}^{N-1} 2\pi \frac{N-n}{A} x e^{-\pi\frac{N-n}{A}x^2} \right\} .$$

These alternative expressions are somewhat simpler than (6.22), (6.23) and (6.24) and according to our experiments, lead to almost identical numerical results.

Special case Π_0^κ:

In this case, we have $\lambda_\kappa = \lambda$ and $\lambda_c = 0$, for $c \in \{1, 2, \cdots, N\} \setminus \{\kappa\}$ and obtain

$$\mathrm{E}[\{\mathrm{D}_{\Pi_0^\kappa}\}^l] = \frac{1}{P'_{\mathrm{serv}}} \sum_{m=1}^{\kappa} \frac{\Gamma(m+\frac{l}{2})}{(\pi\sigma)^{l/2}(m-1)!} \Psi_{m,N} \,,$$

$$F_{\mathrm{D}_{\Pi_0^\kappa}}(x) = \frac{1}{P'_{\mathrm{serv}}} \sum_{m=1}^{\kappa} \frac{\gamma(m, \pi\sigma x^2)}{(m-1)!} \Psi_{m,N} \,,$$

and

$$f_{\mathrm{D}_{\Pi_0^\kappa}}(x) = \frac{\mathbf{1}(x<\xi)}{P'_{\mathrm{serv}}} \sum_{m=1}^{\kappa} \frac{2(\pi\sigma)^m x^{2m-1} e^{-\pi\sigma x^2}}{(m-1)!} \Psi_{m,N} \,,$$

with

$$P'_{\mathrm{serv}} = \sum_{m=1}^{\kappa} \Psi_{m,N} \,,$$

for the moments and the distribution of the dispatch distance.

Queuing System

Although the difference between the travel distances associated with the immediate and delayed dispatches is much less significant than between the corresponding travel times, one would still rather make this distinction. Therefore, we will consider the random dispatch distance of the immediate and delayed assignments, respectively denoted by $\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}^{\kappa}_{\xi}}$ and $\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa}_{\xi}}$, and obtain the corresponding moments and distributions. The expressions for all dispatches are then easily recovered through

$$\mathrm{E}[\{\mathrm{D}_{\bar{\Pi}^{\kappa}_{\xi}}\}^l] = (1-P_{\mathrm{q}})\,\mathrm{E}[\{D^{\mathrm{I}}_{\bar{\Pi}^{\kappa}_{\xi}}\}^l] + P_{\mathrm{q}}\,\mathrm{E}[\{D^{\mathrm{D}}_{\bar{\Pi}^{\kappa}_{\xi}}\}^l]\,,$$

$$F_{\mathrm{D}_{\bar{\Pi}^{\kappa}_{\xi}}}(x) = (1-P_{\mathrm{q}})\,F_{\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}^{\kappa}_{\xi}}}(x) + P_{\mathrm{q}}\,F_{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa}_{\xi}}}(x)\,,$$

and

$$f_{\mathrm{D}_{\bar{\Pi}^{\kappa}_{\xi}}}(x) = (1-P_{\mathrm{q}})\,f_{\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}^{\kappa}_{\xi}}}(x) + P_{\mathrm{q}}\,f_{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa}_{\xi}}}(x)\,.$$

We first consider the policy with at least one backup, that is with $\kappa \geq 1$, denoted by $\bar{\Pi}^{\kappa^+}_{\xi}$. To analyze this case, we first need to define $\theta^i_{\xi}(m,r)$ as the probability of at least i servers being in vicinity ξ of the call location given that the m-th neighbor is at distance r from the call location; we then readily see that

$$\begin{aligned}\theta^i_{\xi}(m,r) = \Pr\{\mathrm{R}_i \leq \xi \,|\, \mathrm{R}_m = r\} &= 1 - \sum_{j=0}^{i-1} \frac{(m-1)!}{j!(m-j-1)!}\left(\frac{\xi}{r}\right)^{2j}\left(1-\frac{\xi^2}{r^2}\right)^{m-j-1} \\ &= \sum_{j=i}^{m-1} \frac{(m-1)!}{j!(m-j-1)!}\left(\frac{\xi}{r}\right)^{2j}\left(1-\frac{\xi^2}{r^2}\right)^{m-j-1}\,.\end{aligned}$$

Conditioning on the distance rank of the first free server and then on the location of the κ-th closest server, we write for the moments of the immediate dispatch distances

$$\mathrm{E}[\{\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}^{\kappa^+}_{\xi}}\}^l] \;=\; \frac{1}{1-P'_{\mathrm{q}}}\left\{\sum_{m=1}^{\kappa}\Pr\{\mathrm{M}=m\}\mathrm{E}[{\mathrm{R}_m}^l] + \sum_{m=\kappa+1}^{N}\Pr\{\mathrm{M}=m\}\int_0^{\xi} f_{\mathrm{R}_m} r^l\,dr\right\}\,, \quad (6.25)$$

where $P'_{\mathrm{q}} = \Pr\{W_{\mathrm{q}} > 0\}$, defined only for the system with queuing, is the probability of a served customer entering service with a queuing delay, and is obtained as

$$1 - P'_{\mathrm{q}} = \sum_{m=1}^{\kappa}\Pr\{\mathrm{M}=m\} + \sum_{m=\kappa+1}^{N}\Pr\{\mathrm{M}=m\}F_{\mathrm{R}_m}(\xi)\,.$$

Simplifying gives

$$\mathrm{E}[\{\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}^{\kappa+}_{\xi}}\}^l] \quad = \quad \frac{1}{1-P'_{\mathrm{q}}}\left(\sum_{m=1}^{\kappa}\frac{\Gamma(m+\frac{l}{2})}{(\pi\sigma)^{l/2}(m-1)!}\Psi_{m,N}+\sum_{m=\kappa+1}^{N}\frac{\gamma(m+\frac{l}{2},\pi\sigma\xi^2)}{(\pi\sigma)^{l/2}(m-1)!}\Psi_{m,N}\right),$$

where

$$1-P'_{\mathrm{q}}=\sum_{m=1}^{\kappa}\Psi_{m,N}+\sum_{m=1}^{N}\frac{\gamma(m,\pi\sigma\xi^2)}{(m-1)!}\Psi_{m,N}\,.$$

For the delayed dispatches, we first write

$$\mathrm{E}[\{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa+}_{\xi}}\}^l]=\frac{1}{P'_{\mathrm{q}}}\left\{P_N\left(\Delta_\xi F_{\mathrm{R}_\kappa}(\xi)+\int_\xi^\infty f_{\mathrm{R}_\kappa}(r)\left[\frac{1}{\kappa}r+(1-\frac{1}{\kappa})\Delta_r\right]dr\right)+\right.$$
$$\left.\sum_{m=\kappa+1}^{N}\Pr\{\mathrm{M}=m\}\int_\xi^\infty f_{\mathrm{R}_m}(r)\left(\Delta^l_\xi F_{\mathrm{R}_\kappa|\mathrm{R}_m=r}(\xi)+\int_\xi^r f_{\mathrm{R}_\kappa|\mathrm{R}_m=r}(x)\left[\frac{1}{\kappa}x^l+(1-\frac{1}{\kappa})\Delta^l_x\right]dx\right)dr\right\}\,; \tag{6.26}$$

where $F_{\mathrm{R}_n|\mathrm{R}_m=r}(x)=\Pr\{\mathrm{R}_n\le x|\mathrm{R}_m=r\}$ and $f_{\mathrm{R}_n|\mathrm{R}_m=r}(x)=\frac{d}{dx}F_{\mathrm{R}_n|\mathrm{R}_m=r}(x)$ are respectively the CDF and PDF of R_n conditional on the m-th nearest server residing at distance r. It is easy to see that $F_{\mathrm{R}_n|\mathrm{R}_m=r}(x)=\theta^n_x(m,r)$ and thus $f_{\mathrm{R}_n|\mathrm{R}_m=r}(x)=\frac{d}{dx}\theta^n_x(m,r)$. We then write

$$\int_\xi^\infty f_{\mathrm{R}_m}(r)\Delta^l_\xi F_{\mathrm{R}_\kappa|\mathrm{R}_m=r}(\xi)dr=\sum_{j=\kappa}^{m-1}\frac{2(\pi\sigma)^m}{j!(m-j-1)!}\Delta^l_\xi\left(\Phi_j(\infty)-\Phi_j(\xi)\right),$$

with

$$\begin{aligned}\Phi_j(r)&=\int r^{2m-1}e^{-\pi\sigma r^2}\left(\frac{\xi}{r}\right)^{2j}\left(1-(\frac{\xi}{r})^2\right)^{m-j-1}dr\\&=-\frac{1}{2}\pi^{j-m}\xi^{2j}e^{-\pi\sigma\xi^2}\sigma^{j-m}\Gamma(m-j,\pi\sigma(r^2-\xi^2))\,;\end{aligned}$$

so, we get

$$\int_\xi^\infty f_{\mathrm{R}_m}(r)\Delta^l_\xi F_{\mathrm{R}_\kappa|\mathrm{R}_m=r}(\xi)dr=\Delta^l_\xi e^{-\pi\sigma\xi^2}\sum_{j=\kappa}^{m-1}\frac{(\pi\sigma\xi^2)^j}{j!}\,,$$

which can be written in terms of incomplete gamma functions as

$$\begin{aligned}\int_\xi^\infty f_{\mathrm{R}_m}(r)\Delta^l_\xi F_{\mathrm{R}_\kappa|\mathrm{R}_m=r}(\xi)dr&=\Delta^l_\xi\left(\frac{\Gamma(m,\pi\sigma\xi^2)}{(m-1)!}-\frac{\Gamma(\kappa,\pi\sigma\xi^2)}{(\kappa-1)!}\right)\\&=\Delta^l_\xi\left(\frac{\gamma(\kappa,\pi\sigma\xi^2)}{(\kappa-1)!}-\frac{\gamma(m,\pi\sigma\xi^2)}{(m-1)!}\right).\end{aligned}$$

Since for $m \geq \kappa+1$, $\mathrm{R}_\kappa \geq \xi$ implies $\mathrm{R}_m \geq \xi$, by the law of total expectation we have

$$\int_\xi^\infty f_{\mathrm{R}_m}(r) \int_\xi^r f_{\mathrm{R}_\kappa|\mathrm{R}_m=r}(x) \left[\frac{1}{\kappa}x^l + (1-\frac{1}{\kappa})\Delta_x^l\right] dx\, dr = \int_\xi^\infty f_{\mathrm{R}_\kappa}(x) \left[\frac{1}{\kappa}x^l + (1-\frac{1}{\kappa})\Delta_x^l\right] dx$$
$$= \frac{(2\kappa+l)\Gamma(\kappa+\frac{l}{2}, \pi\sigma\xi^2)}{(l+2)\kappa!(\pi\sigma)^{l/2}}.$$

We also have

$$P_N\left(\Delta_\xi^l F_{\mathrm{R}_\kappa}(\xi) + \int_\xi^\infty f_{\mathrm{R}_\kappa}(r)\left[\frac{1}{\kappa}r^l + (1-\frac{1}{\kappa})\Delta_r^l\right] dr\right) =$$
$$P_N\left(\frac{\Delta_\xi^l \gamma(\kappa, \pi\sigma\xi^2)}{(\kappa-1)!} + \left[-\frac{(2\kappa+l)\Gamma(\kappa+\frac{l}{2}, \pi\sigma r^2)}{(l+2)\kappa!(\pi\sigma)^{l/2}}\right]_{r=\xi}^{r=\infty}\right)$$
$$= P_N\left(\frac{\Delta_\xi^l \gamma(\kappa, \pi\sigma\xi^2)}{(\kappa-1)!} + \frac{(2\kappa+l)\Gamma(\kappa+\frac{l}{2}, \pi\sigma\xi^2)}{(l+2)\kappa!(\pi\sigma)^{l/2}}\right).$$

Combining all the above results into (6.26), we get

$$\mathrm{E}[\{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}_\xi^{\kappa+}}\}^l] = \frac{1}{P'_{\mathrm{q}}}\left[P_N\left(\frac{\Delta_\xi^l \gamma(\kappa, \pi\sigma\xi^2)}{(\kappa-1)!} + \frac{(2\kappa+l)\Gamma(\kappa+\frac{l}{2}, \pi\sigma\xi^2)}{(l+2)\kappa!(\pi\sigma)^{l/2}}\right) + \right.$$
$$\left. \sum_{m=\kappa+1}^N \Psi_{m,N}\left\{\Delta_\xi^l\left(\frac{\gamma(\kappa, \pi\sigma\xi^2)}{(\kappa-1)!} - \frac{\gamma(m, \pi\sigma\xi^2)}{(m-1)!}\right) + \frac{(2\kappa+l)\Gamma(\kappa+\frac{l}{2}, \pi\sigma\xi^2)}{(l+2)\kappa!(\pi\sigma)^{l/2}}\right\}\right].$$

Based on the same conditioning sequence, we write

$$F_{\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}_\xi^{\kappa+}}}(x) = \frac{1}{1-P'_{\mathrm{q}}}\left\{\sum_{m=1}^\kappa \Pr\{\mathrm{M}=m\} F_{\mathrm{R}_m}(x) + \sum_{m=\kappa+1}^N \Pr\{\mathrm{M}=m\} F_{\mathrm{R}_m}(\min(x,\xi))\right\}$$

and

$$F_{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}_\xi^{\kappa+}}}(x) = \frac{1}{P'_{\mathrm{q}}}\left\{\sum_{m=\kappa+1}^N \Pr\{\mathrm{M}=m\}\left[\int_\xi^\infty f_{\mathrm{R}_m}(r) F_{\mathrm{R}_\kappa|\mathrm{R}_m=r}(\xi)\frac{\min(x,\xi)^2}{\xi^2}dr + \right.\right.$$
$$\left.\int_\xi^\infty f_{\mathrm{R}_m}(r)\int_{\max(x,\xi)}^r f_{\mathrm{R}_\kappa|\mathrm{R}_m=r}(y)\left[(1-\frac{1}{\kappa})\frac{x^2}{y^2}\right]dydr\right] +$$
$$\left.P_N\left(\frac{\min(x,\xi)^2}{\xi^2}F_{\mathrm{R}_\kappa}(\xi) + \int_\xi^{\max(x,\xi)} f_{\mathrm{R}_\kappa}(r)dr + \int_{\max(x,\xi)}^\infty f_{\mathrm{R}_\kappa}(r)\frac{\kappa-1}{\kappa}\frac{x^2}{r^2}\,dr\right)\right\},$$

for the CDF of the immediate and delayed dispatch distances, respectively, and finally arrive at

$$F_{\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}_\xi^{\kappa+}}}(x) = \frac{1}{1-P'_{\mathrm{q}}}\left(\sum_{m=1}^\kappa \frac{\gamma(m, \pi\sigma x^2)}{(m-1)!}\Psi_{m,N} + \sum_{m=\kappa+1}^N \frac{\gamma(m, \pi\sigma\min(x,\xi)^2)}{(m-1)!}\Psi_{m,N}\right)$$

and

$$F_{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa+}_{\xi}}}(x) = \frac{1}{P'_{\mathrm{q}}}\left[P_N\left(\frac{\min(x,\xi)^2\gamma(\kappa,\pi\sigma\xi^2)}{\xi^2(\kappa-1)!} + \frac{\gamma(\kappa,\pi\sigma\max(x,\xi)^2)-\gamma(\kappa,\pi\sigma\xi^2)}{(\kappa-1)!} + \frac{(\kappa-1)\pi\sigma x^2\left(\Gamma(\kappa-1,\pi\sigma\max(x,\xi)^2)\right)}{\kappa!}\right) + \sum_{m=\kappa+1}^{N}\Psi_{m,N}\left\{\frac{\min(x,\xi)^2}{\xi^2}\left(\frac{\gamma(\kappa,\pi\sigma\xi^2)}{(\kappa-1)!}-\frac{\gamma(m,\pi\sigma\xi^2)}{(m-1)!)}\right) + \frac{\gamma(\kappa,\pi\sigma\max(x,\xi)^2)-\gamma(\kappa,\pi\sigma\xi^2)}{(\kappa-1)!} + \frac{(\kappa-1)\pi\sigma x^2\Gamma(\kappa-1,\pi\sigma\max(x,\xi)^2)}{\kappa!}\right\}\right],$$

which lead to the corresponding PDFs

$$f_{\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}^{\kappa+}_{\xi}}}(x) = \frac{2}{1-P'_{\mathrm{q}}}\left\{\sum_{m=1}^{\kappa}\Psi_{m,N}\frac{(\pi\sigma)^m x^{2m-1}e^{-\pi\sigma x^2}}{(m-1)!} + \mathbf{1}(x<\xi)\sum_{m=\kappa+1}^{N}\Psi_{m,N}\frac{(\pi\sigma)^m x^{2m-1}e^{-\pi\sigma x^2}}{(m-1)!}\right\}$$

and

$$f_{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa+}_{\xi}}}(x) = \frac{2}{P'_{\mathrm{q}}}\left[P_N\left(\mathbf{1}(x<\xi)\frac{x\,\gamma(\kappa,\pi\sigma\xi^2)}{\xi^2(\kappa-1)!} + \mathbf{1}(x\geq\xi)\frac{(\pi\sigma)^\kappa x^{2\kappa-1}e^{-\pi\sigma x^2}}{\kappa!} + \frac{\pi\sigma x(\kappa-1)\Gamma(\kappa-1,\pi\sigma\max(x,\xi)^2)}{\kappa!}\right) + \sum_{m=\kappa+1}^{N}\Psi_{m,N}\left\{\frac{\mathbf{1}(x<\xi)\,x}{\xi^2}\left(\frac{\gamma(\kappa,\pi\sigma\xi^2)}{(\kappa-1)!}-\frac{\gamma(m,\pi\sigma\xi^2)}{(m-1)!}\right) + \mathbf{1}(x\geq\xi)\frac{(\pi\sigma)^\kappa x^{2\kappa-1}e^{-\pi\sigma x^2}}{\kappa!} + \frac{(\kappa-1)\,\pi\sigma x\,\Gamma(\kappa-1,\pi\sigma\max(x,\xi)^2)}{\kappa!}\right\}\right]$$

for the immediate and delayed dispatch distances, respectively.

In the next subsection, where we consider the response times, we will see that to derive exact expressions under the $\Pi^{\kappa+}_{\xi}$ policy, distributions of the dispatch distance conditional on C the number of covering servers are required. Therefore, we now provide an alternate characterization of the dispatch distance under the $\Pi^{\kappa+}_{\xi}$ policy by conditioning on C instead of M. The conditional PDFs we obtain in the process will be used in the next subsection.

We first recognize that under the policy $\bar{\Pi}^{\kappa+}_{\xi}$, that is with $\kappa = 1, 2, \cdots, N$ and $\xi \geq 0$, we have

$$C = \kappa \;\leftrightarrow\; \mathrm{R}_{\kappa+1} > \xi\,,$$
$$C > \kappa \;\leftrightarrow\; \mathrm{R}_{\kappa+1} \leq \xi\,.$$

Now, conditioning on C and using the above reasoning, we write for the l-th moment of the dispatch distance for calls covered by exactly κ servers

$$\begin{aligned}\mathrm{E}[\{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa+}_{\xi}}|C=\kappa\}^l] =& \frac{1}{1-F_{\mathrm{R}_{\kappa+1}}(\xi)}\int_{\xi}^{\infty} f_{\mathrm{R}_{\kappa+1}}(r)\,\Delta_r^l\,dr \\ =& \frac{2(\pi\sigma)^{-\frac{l}{2}}\,\Gamma(\kappa+1+\frac{l}{2},\pi\sigma\xi^2)}{(l+2)\,\Gamma(\kappa+1,\pi\sigma\xi^2)}\,.\end{aligned}$$

For $C\in\{\kappa+1,\kappa+2,\cdots,N\}$ we simply have

$$\mathrm{E}\left[\left\{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa+}_{\xi}}|C\in\{\kappa+1,\cdots,N\}\right\}^l\right]=\Delta_{\xi}^l\,.$$

For the CDFs conditional on C we write

$$\begin{aligned}F_{\{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa+}_{\xi}}|\mathrm{C}=\kappa\}}(x) =& \frac{1}{1-F_{\mathrm{R}_{\kappa+1}}(\xi)}\left\{\int_{\max\{x,\xi\}}^{\infty} f_{\mathrm{R}_{\kappa+1}}(r)\,\frac{x^2}{r^2}\,dr+\int_{\xi}^{\max\{x,\xi\}} f_{\mathrm{R}_{\kappa+1}}(r)\,dr\right\} \\ =& 1+\frac{\pi\sigma x^2\,\Gamma(\kappa,\pi\sigma\max\{x,\xi\}^2)-\Gamma(\kappa+1,\pi\sigma\max\{x,\xi\}^2)}{\Gamma(\kappa+1,\pi\sigma\xi^2)}\,,\end{aligned}$$

and

$$F_{\{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa+}_{\xi}}|\mathrm{C}\in\{\kappa+1,\cdots,N\}\}}(x)=\frac{\min(x,\xi)^2}{\xi^2}\,.$$

Differentiation yields the required conditional PDFs; after simplifications we get

$$f_{\{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa+}_{\xi}}|\mathrm{C}=\kappa\}}(x)=\frac{2\pi\sigma\,x\,\Gamma(\kappa,\pi\sigma\max\{x,\xi\}^2)}{\Gamma(\kappa+1,\pi\sigma\xi^2)}\,, \tag{6.27}$$

and

$$f_{\{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa+}_{\xi}}|\mathrm{C}\in\{\kappa+1,\cdots,N\}\}}(x)=\mathbf{1}(x<\xi)\frac{2x}{\xi^2}\,. \tag{6.28}$$

Now, for sake of completeness, we give the moments and distribution of the dispatch distance obtained through a conditioning and then de-conditioning on C. Letting Q be the event of a call getting queued, we write

$$\mathrm{E}[\{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa+}_{\xi}}\}^l]=\frac{\sum_{c=\kappa}^{N}\Pr\{\mathrm{C}=c\}\Pr\{Q\,|\,C=c\}\mathrm{E}[\{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa+}_{\xi}}|C=c\}^l]}{\sum_{c=\kappa}^{N}\Pr\{\mathrm{C}=c\}\Pr\{Q\,|\,C=c\}}\,,$$

which after substitutions gives

$$\mathrm{E}[\{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa+}_{\xi}}\}^l]=\left\{\sum_{c=\kappa}^{N}\zeta'_c\sum_{n=c}^{N}P_n\,\frac{(N-c)!n!}{N!(n-c)!}\right\}^{-1}\times$$

$$\left\{\zeta'_\kappa \frac{2(\pi\sigma)^{-\frac{l}{2}}\,\Gamma(\kappa+1+\frac{l}{2},\pi\sigma\xi^2)}{(l+2)\,\Gamma(\kappa+1,\pi\sigma\xi^2)}\sum_{n=\kappa}^{N} P_n\,\frac{(N-\kappa)!n!}{N!(n-\kappa)!}+\Delta_\xi^l\sum_{c=\kappa+1}^{N}\zeta'_c\sum_{n=c}^{N}P_n\,\frac{(N-c)!n!}{N!(n-c)!}\right\}.$$

Similarly, we have

$$\mathrm{F}_{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa+}_{\xi}}}(x)=\left\{\sum_{c=\kappa}^{N}\zeta'_c\sum_{n=c}^{N}P_n\,\frac{(N-c)!n!}{N!(n-c)!}\right\}^{-1}\times$$
$$\left[\left(1+\frac{\pi\sigma x^2\,\Gamma(\kappa,\pi\sigma\max\{x,\xi\}^2)-\Gamma(\kappa+1,\pi\sigma\max\{x,\xi\}^2)}{\Gamma(\kappa+1,\pi\sigma\xi^2)}\right)\zeta'_\kappa\sum_{n=\kappa}^{N}P_n\,\frac{(N-\kappa)!n!}{N!(n-\kappa)!}\right.$$
$$\left.+\frac{\min(x,\xi)^2}{\xi^2}\sum_{c=\kappa+1}^{N}\zeta'_c\sum_{n=c}^{N}P_n\,\frac{(N-c)!n!}{N!(n-c)!}\right],$$

and

$$\mathrm{f}_{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa+}_{\xi}}}(x)=\left\{\sum_{c=\kappa}^{N}\zeta'_c\sum_{n=c}^{N}P_n\,\frac{(N-c)!n!}{N!(n-c)!}\right\}^{-1}\times$$
$$\left[\left(\frac{2\pi\sigma\,x\,\Gamma(\kappa,\pi\sigma\max\{x,\xi\}^2)}{\Gamma(\kappa+1,\pi\sigma\xi^2)}\right)\zeta'_\kappa\sum_{n=\kappa}^{N}P_n\,\frac{(N-\kappa)!n!}{N!(n-\kappa)!}\right.$$
$$\left.+\mathbf{1}(x<\xi)\frac{2x}{\xi^2}\sum_{c=\kappa+1}^{N}\zeta'_c\sum_{n=c}^{N}P_n\,\frac{(N-c)!n!}{N!(n-c)!}\right].$$

for the PDF and the CDF of the dispatch distance, respectively.

Special Case $\bar{\Pi}^0_\xi$:

For the case with $\kappa=0$, denoted by $\bar{\Pi}^0_\xi$, we might have a non-zero probability of loss, that is $P_{\text{loss}}\geq 0$. Under this policy, (6.25) for the moments of the immediate dispatch distance specializes to

$$\mathrm{E}[\{\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}^0_\xi}\}^l]=\frac{1}{(1-P_{\text{loss}})(1-P_{\mathrm{q}})}\sum_{m=1}^{N}\Pr\{\mathrm{M}=m\}\int_0^\xi f_{\mathrm{R}_m}(r)r^l\,dr\,,$$

and when evaluated using the Poisson assumptions, yields

$$\mathrm{E}[\{\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}^0_\xi}\}^l]=\frac{1}{(1-P'_{\text{loss}})(1-P'_{\mathrm{q}})}\sum_{m=1}^{N}\frac{\gamma(m+\frac{l}{2},\pi\sigma\xi^2)}{(\pi\sigma)^{l/2}(m-1)!}\,\Psi_{m,N}\,,$$

where the normalizing factor $(1-P'_{\text{loss}})(1-P'_{\mathrm{q}})$ is understood to represent the fraction of immediate dispatches to all arrivals (including the lost ones) obtained using the distance

distributions derived based on special Poisson assumption, that is

$$(1 - P'_{\text{loss}})(1 - P'_{\text{q}}) = \sum_{m=1}^{N} \frac{\gamma(m, \pi\sigma\xi^2)}{(m-1)!} \Psi_{m,N} \,. \tag{6.29}$$

For the delayed dispatches and for $\kappa = 0$, (6.26) reduces to

$$\mathrm{E}[\{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa^+}_{\xi}}\}^l] = \frac{1}{P_{\text{q}}} \left\{ P_N \Delta^l_\xi + \frac{1}{(1 - P_{\text{loss}})} \sum_{m=1}^{N} \Pr\{\mathrm{M} = m\} \Delta^l_\xi \int_\xi^\infty \theta^1_\xi(m, r)\, f_{\mathrm{R}_m}(r)\, dr \right\},$$

and gives

$$\mathrm{E}[\{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{0}_{\xi}}\}^l] = \frac{1}{(1 - P'_{\text{loss}})P'_{\text{q}}} \sum_{m=1}^{N} \Delta^l_\xi \left(1 - e^{-\pi\sigma\xi^2} - \frac{\gamma(m, \pi\sigma\xi^2)}{(m-1)!} \right) \Psi_{m,N} + \Delta^l_\xi \frac{P_N}{P'_{\text{q}}} \,, \tag{6.30}$$

where it holds that

$$(1 - P_{\text{loss}})P_{\text{q}} = P_N(1 - P_{\text{loss}}) + \sum_{m=1}^{N} \Psi_{m,N} \int_\xi^\infty \theta^1_\xi(m, 1) f_{\mathrm{R}_m}(r)\, dr \,,$$

and thus

$$(1 - P'_{\text{loss}})P'_{\text{q}} = P_N(1 - P'_{\text{loss}}) + \sum_{m=1}^{N} \Psi_{m,N} \left(1 - e^{-\pi\sigma\xi^2} - \frac{\gamma(m, \pi\sigma\xi^2)}{(m-1)!} \right) \,, \tag{6.31}$$

after substituting for $f_{\mathrm{R}_m}(r)$. Combining (6.29) and (6.31) and simplifying (Note that $P_N + \sum_{m=1}^{N} \Psi_{m,N} = 1$) we get

$$P'_{\text{loss}} = e^{-\pi\sigma\xi^2} \,, \tag{6.32}$$

which can be verified by direct computation of P'_{loss} as

$$P'_{\text{loss}} = 1 - F_{\mathrm{R}_1}(\xi) = 1 - \gamma(1, \pi\sigma\xi^2) = e^{-\pi\sigma\xi^2}.$$

Plugging (6.32) back into (6.31) we get P'_{q} as

$$P'_{\text{q}} = 1 - \frac{\sum_{m=1}^{N} \Psi_{m,N} \frac{\gamma(m, \pi\sigma\xi^2)}{(m-1)!}}{1 - e^{-\pi\sigma\xi^2}} \,.$$

With P'_{loss} and P'_{q} determined, the moments of the delayed dispatch distances is then obtained from (6.30). Alternatively, one can simply recognize that

$$\mathrm{E}[\{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^0_\xi}\}^l] = \Delta^l_\xi \,,$$

since any delayed dispatch under this policy will be to the call location from a uniformly to obtain the distribution of the random point within distance ξ. We use this straightforward reasoning in computing the distribution of the dispatch distance as well.

The CDF of the immediate and delayed dispatch distances are then obtained as

$$F_{\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}^0_\xi}}(x) = \frac{1}{(1-P'_{\text{loss}})(1-P'_{\text{q}})} \sum_{m=1}^{N} \frac{\gamma(m, \pi\sigma\min(x,\xi)^2)}{(m-1)!} \Psi_{m,N}$$

and

$$F_{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^0_\xi}}(x) = \frac{\min(x,\xi)^2}{\xi^2} \,;$$

with the corresponding PDFs

$$f_{\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}^0_\xi}}(x) = \frac{\mathbf{1}(x<\xi)}{(1-P'_{\text{loss}})(1-P'_{\text{q}})} \sum_{m=1}^{N} \frac{2(\pi\sigma)^m x^{2m-1} e^{-\pi\sigma x^2}}{(m-1)!} \Psi_{m,N}$$

and

$$f_{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^0_\xi}}(x) = \mathbf{1}(x<\xi)\frac{2x}{\xi} \,,$$

where the normalizing factor $(1-P'_{\text{loss}})(1-P'_{\text{q}})$ is given by (6.29).

Special Case $\bar{\Pi}^\kappa_0$:

For the case with only a limit on the backup number, the queuing probability becomes

$$P_{\text{q}} = \sum_{n=\kappa}^{N} P_n \frac{(N-\kappa)!n!}{N!(n-\kappa)!} \,,$$

and we note that

$$P_{\text{q}} = 1 - \sum_{m=1}^{\kappa} \Psi_{m,N} \,. \tag{6.33}$$

For the immediate dispatches we write

$$\mathrm{E}[\{\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}^\kappa_0}\}^l] = \frac{1}{1-P_{\text{q}}} \sum_{m=1}^{\kappa} \Pr\{\mathrm{M}=m\}\mathrm{E}[\mathrm{R}_m{}^l]$$

which gives

$$\mathrm{E}[\{\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}^{\kappa}_{0}}\}^{l}] = \frac{1}{1-P_{\mathrm{q}}} \sum_{m=1}^{\kappa} \frac{\Gamma(m+\frac{l}{2})}{(\pi\sigma)^{l/2}(m-1)!} \Psi_{m,N} \ .$$

Note that, in this case, the queuing probability depends only on M and not on R_m and hence P_{q} can be used for normalizing, as reflected in (6.33).

Every delayed dispatch under this policy will assign the call to one of the first κ neighbors with equal probability (servers are indistinguishable); therefore

$$\mathrm{E}[\{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa}_{0}}\}^{l}] = \frac{1}{\kappa} \sum_{h=1}^{\kappa} \mathrm{E}[\mathrm{R}_h{}^{l}] = \frac{1}{\kappa} \sum_{h=1}^{\kappa} \frac{\Gamma(h+\frac{l}{2})}{(\pi\sigma)^{l/2}(h-1)!} \ .$$

The corresponding CDFs are

$$F_{\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}^{\kappa}_{0}}}(x) = \frac{1}{1-P_{\mathrm{q}}} \sum_{m=1}^{\kappa} \frac{\gamma(m,\pi\sigma x^2)}{(m-1)!} \Psi_{m,N}$$

and

$$F_{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa}_{0}}}(x) = \frac{1}{\kappa P_{\mathrm{q}}} \sum_{h=1}^{\kappa} \frac{\gamma(h,\pi\sigma x^2)}{(h-1)!} \ ,$$

with the PDFs

$$f_{\mathrm{D}^{\mathrm{I}}_{\bar{\Pi}^{\kappa}_{0}}}(x) = \frac{2}{1-P_{\mathrm{q}}} \sum_{m=1}^{\kappa} \frac{(\pi\sigma)^m x^{2m-1} e^{-\pi\sigma x^2}}{(m-1)!} \Psi_{m,N}$$

and

$$f_{\mathrm{D}^{\mathrm{D}}_{\bar{\Pi}^{\kappa}_{0}}}(x) = \frac{2}{\kappa P_{\mathrm{q}}} \sum_{h=1}^{\kappa} \frac{(\pi\sigma)^h x^{2h-1} e^{-\pi\sigma x^2}}{(h-1)!} \ ,$$

for the immediate and delayed dispatches.

6.1.5 Response Time

With the dispatch distance and queuing delay characterized in previous sections, we are now in position to analyze the response time, which is defined as the time elapsed from the moment the call is received to the moment the server arrived at the emergency scene. In general, we can think of the response time as comprising of two main components, the pre-travel delay and the travel time. The pre-travel delay may further be broken down into two segments, the dispatch time, taken by the dispatcher to assess the call and make a decision, and the time-to-on-route or chute time, from the moment the dispatch decision is made until the server is actually starts moving towards the scene. The number of components and the analytic or numerical methods used in modelling the pre-travel delay, and by extension other non-travel components of the service time, highly depend on the application and the level of

detail justified or desired by the analyst. Therefore, we suffice to write the random response time R as

$$\mathrm{R} = \mathrm{T}_{que} + \mathrm{T}_{pre-trv} + \mathrm{T}\,,$$

where T_{que} is the random queuing delay, $\mathrm{T}_{pre-trv}$ is the total pre-travel delay, and T is the travel time. Now, while the randomness in the pre-travel delay can be reasonably assumed to be independent of the dispatch distance, the travel time obviously depends on the travel distance. However, the travel time may, in reality, depend on many other factors besides the travel distance, and these can also be accounted for via models such as the one proposed in Budge et al. (2010) that we utilized in Section 4. These secondary dependence on factors beside the travel distance are mostly application-specific and might be ignored for a simpler model, as we do here. Thus, we only consider the variation of the travel time with the travel distance.

Recall (4.25) of the Budge et al. (2010) travel time model that gives the *median* of the stochastic time to travel a distance of d. Now, ignoring the stochasticity of travel time about this average value (that is setting $cv = 0$), we arrive at

$$g(d) = \begin{cases} 2\sqrt{d/a} & \text{if } d \le 2d_c \\ v_c/a + d/v_c & \text{if } d > 2d_c \end{cases},$$

which is in fact the original travel time model of Kolesar et al. (1975), where $g(d)$ is now the mean travel time $t(d)$ to cover a given travel distance d; a and v_c are the acceleration and the cruising speed; and we have $d_c = v_c^2/(2a)$. We then get the CDF and PDF of T_Π the random travel distance under a general unspecified dispatch policy Π as

$$F_{\mathrm{T}_\Pi}(t) = F_{\mathrm{D}_\Pi}(h(t)) \tag{6.34}$$

and

$$f_{\mathrm{T}_\Pi}(t) = h'(t)\, f_{\mathrm{D}_\Pi}(h(t))\,, \tag{6.35}$$

where

$$h(t) = g^{-1}(t) = \begin{cases} \frac{1}{4}\, a\, t^2 & t \le \frac{2v_c}{a}\,, \\ t\, v_c - \frac{v_c^2}{a} & t > \frac{2v_c}{a}\,,. \end{cases} \tag{6.36}$$

The l-th raw moment of the travel time is of course obtained via

$$\mathrm{E}[\mathrm{T}_\Pi^l] = \int_{t=0}^{\infty} f_{\mathrm{T}_\Pi}(t)\, t^l\, dt\,,$$

for $l = 1, 2, \cdots$. Now, replacing f_{D_Π} and F_{D_Π} in (6.34), (6.35), and (6.36) with the appropriate

expressions from Section 6.1.4, we obtain the PDF and CDF of $\mathrm{T}^{\mathrm{I}}_{\Pi}$ and $\mathrm{T}^{\mathrm{D}}_{\Pi}$ the random travel time for the immediate and delayed dispatches, respectively.

Having computed the distribution of the travel time and assuming a given distribution for the non-travel components of the response time, the distribution of the response time is obtained as the convolution of its components. For a system with queues, however, we distinguish between the response times of immediate and delayed dispatches. With $f * g$ designating the convolution of functions f and g, the distribution of the response time for the immediate dispatches $\mathrm{R}^{\mathrm{I}}_{\Pi}$ is given as

$$f_{\mathrm{R}^{\mathrm{I}}_{\Pi}}(t) = f_{\mathrm{T}^{\mathrm{I}}_{\Pi}} * f_{\mathrm{T}_{pre-trv}} \,,$$

while for the delayed dispatch we have to include the queuing delay as well; we write

$$f_{\mathrm{R}^{\mathrm{D}}_{\Pi}}(t) = f_{\mathrm{T}^{\mathrm{D}}_{\Pi}} * \left(f_{\mathrm{T}_{que}} * f_{\mathrm{T}_{pre-trv}} \right) \,, \tag{6.37}$$

which is exact when T_{que} and $\mathrm{T}^{\mathrm{D}}_{\Pi}$ are independent; otherwise, it will be an approximation. In fact, within the context considered here, the queuing delay and the travel time may depend on C the random number of compatible servers and thus, depending on the policy, a conditioning and de-conditioning on C may be required to arrive at the exact expression. For the special case of Π^{κ}_{0}, every call is covered by exactly κ servers and hence $\mathrm{C} = \kappa$ with probability 1; therefore, equation (6.37) holds exactly. For the special case of Π^{0}_{ξ}, C might assume values in the set $\{0, 1, \cdots, N\}$; however, as discussed earlier and regardless of C, the dispatch distance will be distributed as the random distance from the call location to a uniformly random point within a neighborhood of radius ξ. Consequently, although the queuing delay will depend on C, the travel time will not, and (6.37) will again be exact. Under the general $\Pi^{\kappa+}_{\xi}$ policy, however, both T_{que} and $\mathrm{T}^{\mathrm{D}}_{\Pi}$ will depend on C and we must condition on C if an exact expression is desired. We have already obtained the distribution of the queuing delay conditional on C as (6.16) and the PDF of the delayed dispatch distance conditional on C as (6.27) and (6.28). We first obtain the travel time conditional on C as

$$f_{\{\mathrm{T}^{D}_{\Pi^{\kappa+}_{\xi}}|\mathrm{C}=c\}}(t) = h'(t)\, f_{\{D^{\mathrm{D}}_{\Pi^{\kappa+}_{\xi}}|\mathrm{C}=c\}}(h(t)) \,, \quad c = \kappa, \kappa+1, \cdots, N \,,$$

and then, conditioning on C as before, we obtain the PDF of the response time as the mixture of convolutions

$$f_{\mathrm{R}^{\mathrm{D}}_{\bar{\Pi}^{\kappa+}_{\xi}}} = f_{\mathrm{T}_{pre-trv}} * \left[\left(\sum_{c=\kappa}^{N} \zeta'_c \sum_{n=c}^{N} P_n \frac{(N-c)!n!}{N!(n-c)!} \right)^{-1} \times \right.$$

$$\sum_{c=\kappa}^{N} \zeta'_c \left(f_{\{\mathrm{T}^{\mathrm{D}}_{\Pi^{\kappa^+}_{\xi}} | C=c\}} * f_{\mathrm{W}^D_c} \right) \sum_{n=c}^{N} P_n \frac{(N-c)!n!}{N!(n-c)!} \Bigg] ,$$

which completes our analysis of the response times.

Finally, we note that the pre-travel delay may itself be a convolution of its components and that these components may depend on whether the dispatch is immediate or delayed. For instance, we might break the pre-travel time of the immediate dispatches into dispatch and chute times, but for the delayed dispatches consider only the chute time since for a queued call, the dispatch process will most probably be completed while the call is waiting in the queue. Likewise, the chute time can be distributed differently for immediate and delayed dispatches.

6.1.6 Response Outcome

Let $u(t)$ be a real-valued function giving the outcome, the service quality, or in general, the effectiveness of responding to a call in t time units. Obviously, in most realistic applications, $u(t)$ will be a non-decreasing function. The average outcome of the ESS under the policy Π is then obtained by evaluating the integral

$$\mathrm{E}[\mathrm{U}_\Pi] = P_{\text{loss}} \left\{ P_{\mathrm{q}} \int_{t=0}^{\infty} f_{\mathrm{R}^{\mathrm{D}}_\Pi}(t)\, u(t)\, dt + (1-P_{\mathrm{q}}) \int_{t=0}^{\infty} f_{\mathrm{R}^{\mathrm{I}}_\Pi}(t)\, u(t)\, dt \right\} + (1-P_{\text{loss}})\, \mathrm{E}[\mathrm{U}_{\text{loss}}] ,$$

where $\mathrm{E}[\mathrm{U}_{\text{loss}}]$ is the expected outcome for the lost calls. Furthermore, if one is interested in the distribution of the outcome, the CDF and PDF are given by

$$F_{\mathrm{U}_\Pi}(x) = P_{\text{loss}} F_{\mathrm{U}_{\text{loss}}}(x) + (1-P_{\text{loss}}) \left\{ (1-P_{\mathrm{q}})(1-F_{\mathrm{R}^{\mathrm{I}}_\Pi}(\tau(x))) + P_{\mathrm{q}}(1-F_{\mathrm{R}^{\mathrm{D}}_\Pi}(\tau(x))) \right\} ,$$

if the distribution of U_{loss} is given, and

$$F_{\mathrm{U}_\Pi}(x) = \mathbf{1}(\mathrm{E}[\mathrm{U}_{\text{loss}}] \le x) P_{\text{loss}} + (1-P_{\text{loss}}) \left\{ (1-P_{\mathrm{q}})(1-F_{\mathrm{R}^{\mathrm{I}}_\Pi}(\tau(x))) + P_{\mathrm{q}}(1-F_{\mathrm{R}^{\mathrm{D}}_\Pi}(\tau(x))) \right\} ,$$

if only the expectation of U_{loss} is available. The corresponding PDFs are

$$f_{\mathrm{U}_\Pi}(x) = P_{\text{loss}} f_{\mathrm{U}_{\text{loss}}}(x) + \tau'(x)(1-P_{\text{loss}}) \left\{ (P_{\mathrm{q}}-1) f_{\mathrm{R}^{\mathrm{I}}_\Pi}(\tau(x)) - P_{\mathrm{q}} f_{\mathrm{R}^{\mathrm{D}}_\Pi}(\tau(x)) \right\} ,$$

and

$$f_{\mathrm{U}_\Pi}(x) = \tau'(x)(1-P_{\text{loss}}) \left\{ (P_{\mathrm{q}}-1) f_{\mathrm{R}^{\mathrm{I}}_\Pi}(\tau(x)) - P_{\mathrm{q}} f_{\mathrm{R}^{\mathrm{D}}_\Pi}(\tau(x)) \right\} ,$$

where the last expression is defined over $\mathbb{R}^+/\{\mathrm{E}[\mathrm{U}_{\mathrm{loss}}]\}$ instead of $\mathbb{R}^+$, and $\tau(x) = u^{-1}(x)$ gives the response time corresponding to the outcome value x.

6.1.7 Service Time

In addition to $[\lambda_c]$, we need the service rate μ to obtain the state probabilities of the queuing or loss system with partial service. Looking at the components of the service time in reverse chronological order, we write

$$\mathrm{T}_{serv} = \mathrm{T}_{reset} + \mathrm{T}_{trans} + \mathrm{T}_{scene} + \mathrm{T}_{trv} + \mathrm{T}_{pre-trv}\,,$$

where T_{reset} is the reset or turnaround time from the moment the call is completed until the server is back in service, T_{trans} is the time it takes to transport the patient to a care center in case a transport is needed, and T_{scene} is the time spent on the scene of the incident. The rest of the time components have been mentioned before as comprising the response time. We obtain the mean service rate as

$$\mu = \frac{1}{\mathrm{E}[\mathrm{T}_{serv}]}\,,$$

which implies that the variations of the service time caused by variations in the travel time can be ignored. This is a classic assumption in the analysis of the spatially distributed systems and is closer to reality when the travel time constitutes a smaller fraction of the total service time (Larson and Odoni (1981)). It is worth clarifying that, unlike the other non-travel components of the service time, the distribution of the travel time will generally depend on system state and thus vary from call to call, which in turn results in state dependent and not identically distributed service times. However, in most practical applications involving ESSs, where the quickness of response is important, it seems reasonable to assume the travel time component to be a small fraction of the service time, and as mentioned in Alanis et al. (2013), we make this assumption to avoid complications.

Assuming that the components of the service time are mutually independent, we write

$$\mathrm{E}[\mathrm{T}_{serv}] = \mathrm{E}[\mathrm{T}_{reset}] + P_{trans}\mathrm{E}[\mathrm{T}_{trans}] + \mathrm{E}[\mathrm{T}_{scene}] + \mathrm{E}[\mathrm{T}_{pre-trv}] + P_{\mathrm{q}}\,\mathrm{E}[\mathrm{T}_{\Pi}^{\mathrm{D}}] + (1 - P_{\mathrm{q}})\,\mathrm{E}[\mathrm{T}_{\Pi}^{\mathrm{I}}]\,,$$

where P_{trans} is the given probability of a patient requiring a transport to a care center.

6.1.8 The Algorithm

We are finally in position to use an iterative algorithm to compute the quantities of interest for the ESS with re-positioning discussed under our dispatch policy.

1. Initialization: Initialize the problem definition parameters N, A, λ_{ip}
2. Compute the demand vector $[\lambda_c]$
3. Compute the state probabilities P_n
4. Compute the loss and queue probabilities, P_{q} and P_{loss};
5. Compute the mean and distribution of the immediate and delayed dispatches
6. Compute the mean and distribution of the travel times for the immediate and delayed dispatches
7. Compute the mean service time and update the service rate accordingly
8. If the change in the service rate is smaller than a given threshold, start a new iteration from step 2.
9. Compute the expected and possibly the distribution of the response outcome using the values of the parameters from the last iteration

6.2 Drone-Van Combo Systems

The analysis presented in the previous section cannot be directly applied to the quickly emerging drone-van combo systems (see the project with Mercedes-Benz vans and drones from Matternet) where drones launch from their host mobile ground bases (vans) in response to calls for service and fly back to their home bases when the service is completed. Fortunately, however, these systems can still be treated using a modified version of the analysis presented in the previous section.

Let there be K mobile ground bases, or vans, each operating and carrying H drones resulting in a total of $N = K\,H$ drones responding to requests for emergency services of some sort with the Poisson arrival rate of λ originating from uniformly random locations over a service region of total area A. In response to a new call, the closest free and compatible drone under a partial backup dispatch policy Π will be assigned to the call for service and return to the same van upon finishing the service. With the dispatching policy specified as Π^{κ}_{ξ} for the loss

system or $\bar{\Pi}^{\kappa}_{\xi}$ in case of the system with queues, $\kappa \in \{0, 1, \cdots, K\}$ is now understood to indicate the maximum number of neighboring ground bases (vans), rather than drones, to consider for dispatch to a call arrival. Likewise, $\xi \geq 0$ specifies the distance from the call location within which every van is deemed eligible for dispatch. A ground base with at least one free drone is considered free. The nearest free and compatible ground base will always respond to the request for service by initiating the launch of one of its free drones to the call location. If no compatible ground bases exist, the call will be lost. If all of the compatible ground bases are busy, that is with no free drones, the call will be lost if queues are not allowed and queued otherwise. The system will attempt to minimize the gaps in covering of the service area by relocating the free ground bases to new waiting spots every time a new ground base becomes available or busy, which happens when the number of busy drones attached to a ground base drops from H to $H-1$ or increases from $H-1$ to H, respectively.

With $k = 0, \cdots, K$ busy ground bases, the number of busy drones will be at least kH and at most $k + K(H-1)$. The probability of k busy ground bases conditional on n busy drones is then

$$\Pr\{\mathrm{K} = k | \mathrm{N} = n\} = \frac{\binom{(K-k)(H-1)}{n-kH}}{\binom{KH}{n}} = \frac{[(K-k)(H-1)]!(KH-n)!n!}{(KH)!(n-kK)![K(H-1)+k-n]!};$$

thus from

$$\Pr\{\mathrm{K} = k\} = \sum_{n=kK}^{k+K(H-1)} \Pr\{\mathrm{N} = n\} \Pr\{\mathrm{K} = k | \mathrm{N} = n\},$$

we get

$$P_k \equiv \Pr\{\mathrm{K} = k\} = \sum_{n=kK}^{k+K(H-1)} P_n \frac{[(K-k)(H-1)]!(MK-n)!n!}{(MK)!(n-kK)![K(H-1)+k-n]!},$$

with the distribution of the the number of busy drones P_n, $n \in \{0, \cdots, N\}$ from (4.1) or (4.3) for the queue and loss systems, respectively. To obtain P_n, however, we need the appropriate demand vector $[\lambda_c]$, $c = 0, \cdots, N$, which, unlike before, cannot be computed from the distance distributions via (6.2) as it is now the locations of the ground bases, and not necessarily the drones', which is assumed uniformly and independently distributed over the service area. Paralleling the definition of C, we define Y as the random number of *ground vehicles*, busy or available, residing within distance ξ from the random call location. Correcting for edge

effects, we write

$$\upsilon_t \equiv \Pr\{\mathrm{Y} = t\} = \begin{cases} \tilde{F}_{\mathrm{R}_t}(\xi) - \tilde{F}_{\mathrm{R}_{t+1}}(\xi) & t = 1, \cdots, K-1, \\ 1 - \tilde{F}_{\mathrm{R}_1}(\xi) & t = 0, \\ \tilde{F}_{\mathrm{R}_N}(\xi) & t = K\,, \end{cases}$$

with $\tilde{F}_{\mathrm{R}_t}(\xi)$ the CDF of the edge-corrected random distance to the t-th neighbor as obtained in 5. The assumption that drones fly back to the home base after each service implies that each dispatch initiates from the location of the home base, tying the location of the drones to their home bases regardless of their busy status. Therefore, we redefine C as the random number of drones located within distance ξ of the call location if each drone were at the same place as their home ground base. This implies the distribution of C as

$$\zeta_c \equiv \Pr\{\mathrm{C} = c\} = \begin{cases} \upsilon_i & c = iH\,, \\ 0 & c \neq iH\,, \end{cases}$$

for $c \in \{0, \cdots, N\}$. As before, we modify υ_t to enforce the minimum number of backups specified in the dispatch policy Π_ξ^κ or $\bar{\Pi}_\xi^\kappa$; that is

$$\upsilon'_t = \begin{cases} 0 & t < \kappa - 1\,, \\ \sum_{t'=0}^{\kappa} \zeta_{t'} & t = \kappa\,, \\ \zeta_t & t > \kappa\,, \end{cases}$$

for $t \in \{0, \cdots, K\}$, which in turn gives ζ'_c as

$$\zeta'_c = \begin{cases} \upsilon'_i & c = iH\,, \\ 0 & c \neq iH\,, \end{cases}$$

and the demand vector $[\lambda_c]$ through $\lambda_c = \lambda\,\zeta_c$, for $c \in \{0, 1, \cdots, N\}$.

Before incorporating these modification into a new algorithm, we remark that the key difference between the analysis of the regular systems discussed in the previous section and the drone-van combo system considered here, is the computation of the moments and the distribution of the dispatch distance, which are now obtained based on the random locations of the ground bases and their busy statuses rather than the drones they operate.

The algorithm for computing the quantities of interest for the drone-van combo ESS system with re-positioning under the dispatch policy Π_ξ^κ or $\bar{\Pi}_\xi^\kappa$ is given below.

1. Initialize the problem definition parameters H, K, N, A, λ, and the dispatch policy Π_{ξ}^{κ} or $\bar{\Pi}_{\xi}^{\kappa}$.
2. Compute $\tilde{F}_{\mathrm{R}_k}(\xi)$ and then ζ_k and ζ_k' for $k = 0, \cdots, K$.
3. From ζ_k' compute ζ_c' and then the demand vector $[\lambda_c]$ for $c = 0, \cdots, N$.
4. With $[\lambda_c]$ and μ, compute the distribution of the number of busy drones P_n, $n = 0, \cdots, N$.
5. With P_n and ζ_c', compute the loss and queue probabilities, P_{q} and P_{loss}.
6. With P_n, compute the distribution of the number of busy ground bases P_k, $k = 0, \cdots, K$.
7. With K, P_k, and $\Psi_{m,K}$ for $m = 1, \cdots, K$ and $k = 0, \cdots, K$ replacing N, $\Psi_{m,N}$, and P_n for $m = 1, \cdots, N$ and $n = 0, \cdots, N$, respectively, compute the mean and distribution of the immediate and delayed dispatch distances using the appropriate expressions from Section 6.1.4.
8. Compute the mean and distribution of the travel times for the immediate and delayed dispatches.
9. Compute the mean service time and update the service rate accordingly.
10. If the change in the service rate is smaller than a given threshold, start a new iteration from step 2.
11. Compute the expected and possibly the distribution of the response outcome using the values of the parameters from the last iteration.

6.3 Examples

We present two application examples of the procedure described in this chapter. In the first application, we deal with a fleet of drones deployed alongside an already existing EMS in an urban environment to deliver Automated External Defibrillators (AEDs) to the scenes of Out-of-Hospital Cardiac Arrest (OHCA). The drone fleet in this example application will operate as a loss system where calls not responded to by the aerial fleet will be handed off to the default ground crew. In the second application, we consider a general queuing ESS operating as the main service provider to a geographic area. For ease of comparison, in both cases, we will first assume a baseline total service area equal to the Island of Montreal, that

is almost exactly 500 km^2. We then examine the effects of the demand density by scaling this baseline area by a factor f_A. Likewise, we may scale the total demand by a factor f_λ, in order to see the effects of congestion. Finally, for both examples, we will take a closer look into the case with zero-backup dispatch policies, that is with $\kappa = 0$.

6.3.1 Drones to deliver AEDs to cardiac arrest incidents in an urban environment

We consider a scenario in which a fleet of drones is deployed alongside an already existing traditional ground EMS and the supporting services (fire, police, etc) to improve the survival chance of patients with OHCAs. Drones can fly directly to the scene at high speeds, unaffected by traffic, and without significant en-route times. In this example, they are equipped with AEDs and reserved solely for the cases of OHCA thus significantly reducing the time-to-defibrilation and ultimately patients' chance of survival. Upon a call arrival, we assume the dispatcher decides whether to dispatch the drone or leave the call to the ground crew. If an intervention with drones is chosen, the closest drone will fly to the scene of the incidence bringing an AED to the call location so the bystanders can start the defibrilation process while the ground crew are on their way, perhaps following instructions on how to use the AED remotely communicated through the drone. We assume that the ground EMS crew will reset the dispatched drone after each service completes and take the drone with them or send it flying to a new waiting location which is chosen to provide the sufficient coverage of the service region according to a relocation policy. The resetting procedure may include, for example, the change of batteries, AEDs, or other medical devices carried by the drone. The ground vehicles will transport the patient to a hospital if needed, therefore, from the perspective of the drone system, no arrival needs a transport to a hospital. Also, the drone network operates as a loss system with the lost calls handled by the default ground EMS.

Naturally, one would expect that an intervention using drones will be worthwhile only if it results in an improvement over the outcome expected from the ground EMS, which, given the exponential decay of the survival chance with the response time, will be the case if the distance between the call location and the dispatched drone is relatively small. As discussed earlier, situations like this suggest possible usefulness of incorporating partial backup dispatch policies in improving the system performance and resource utilization. We also expect the potential improvements to be more pronounced when the same demand is dispersed over a larger area leading to greater distances between the response units and scenes of emergency, and also between the response units themselves. We thus apply the descriptive algorithm described in this chapter to discover if and how a partial dispatch policy for this loss system

will be beneficial. Table 6.1 summarizes the parameters selected for this application. The

Table 6.1 Parameters used in the first example

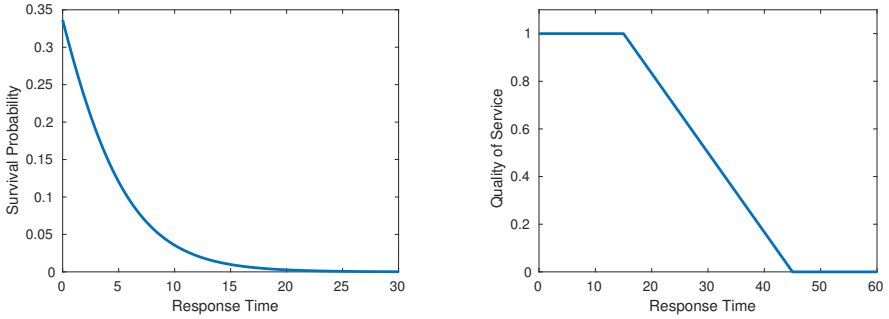
Parameter	Value
Baseline area of service region	500 km^2
Total arrival rate	0.1849 per hour
Average scene time	35 minutes
Average chute time	10 seconds
Average dispatch time	60 seconds
Probability of transport	0
Drones' acceleration	1000 km/hour.min
Drones' top speed	100 km/hour

expected survival probability of a patient experiencing an OHCA and treated by a traditional ground EMS is taken to be 0.08 following the literature (Hansen et al. 2015, Wissenberg et al. 2013, Chan et al. 2014, Brooks et al. 2010, Sasson et al. 2010, Nichol et al. 2008). The arrival rate of 0.1849 per hour was obtained from the estimation of the North American rate of OHCA given by Nichol et al. (2008) as 95 per 100000 people per year, adjusted for the population of the island of Montreal (1704694 as of 2016). The average values for the components of the service time and the acceleration and the maximum travel speed of the drones were estimated from the few published works on the optimization of drone fleets for AED delivery (Claesson et al. 2016, 2017, Sanfridsson et al. 2019, Boutilier et al. 2017) and also the physical properties and capabilities of the most recent professional drones. The model given by De Maio et al. (2003) for the survival probability in cases of OHCA was selected as the outcome function, mainly because of its simplicity since it only depends on the response time. The model is given by

$$u(t) = 1/(1 + e^{0.679+0.262\,t})\,,$$

and is also plotted in Figure 6.1-(a).

The results of the analysis for f_A=1, 2, 4, and 8 are plotted in Figures 6.2—6.5, where the full backup dispatch policy is compared with an optimal partial backup dispatch policy Π_ξ^κ obtained by exploring the space of possible values of κ and ξ. In addition to the (near) optimal values of κ and ξ, we have also plotted the expected survival probability, average server (drone) utilization and the loss probability corresponding to the optimal partial backup and full backup policies. As we noted before, the loss probability in this case represents the fraction of calls that are directly transferred to the ground EMS without a drone dispatch. We observe that, as one would expect, for a large enough fleet size, there is no benefit in imposing



(a) Example 1 (b) Example 2

Figure 6.1 Response outcome functions used in the example applications

limits on the dispatch distance. This is because increasing the fleet size will reduce both the workload and the response area of each drone, which in turn decreases the need for the drones to travel long distances, thus rendering the use of partial backup policies unnecessary. For small fleet sizes, however, explicitly limiting the maximum dispatch distance through a partial backup policy results in improved overal performance while reducing the load imposed on the drones. This is, of course, achieved by avoiding the potential long-distance, inefficient drone interventions by transferring them to the ground EMS as reflected in the plots of loss probabilities and also the supplementary plots given in Appendix C.1 which clearly show how the average travel time increases with ξ.

Also, the partial backup dispatch policies become more relevant for larger service areas. This is natural as increasing the service area is expected to have the same effect as decreasing the fleet size. In particular, the minimum number of drones to achieve the best performance obtainable by a partial backup dispatch policy increases with the size of the service region. For instance, for the baseline case of $f_A = 1$ which represents demand densities typically found in urban environments, no partial backup policies are required with three drones or more; however, for $f_A = 8$, which could reasonably considered to represent a rural region, even with 10 drones, there will be room for performance improvements through utilization of a partial backup dispatch policy.

We already noted that for fleets larger than a certain limit, that is $N > \tilde{N}$, no improvements will be possible by using a partial backup dispatch policy and hence the corresponding optimal survival probability will be equal to the full-backup value within a small computation error. This is reflected in the irregular changes in the optimal κ and ξ values; therefore, the optimal backup numbers and maximum dispatch distances for $N > \tilde{N}$ should be effectively ignored. For $N \leq \tilde{N}$, however, we observe that the optimal backup number is always zero regardless

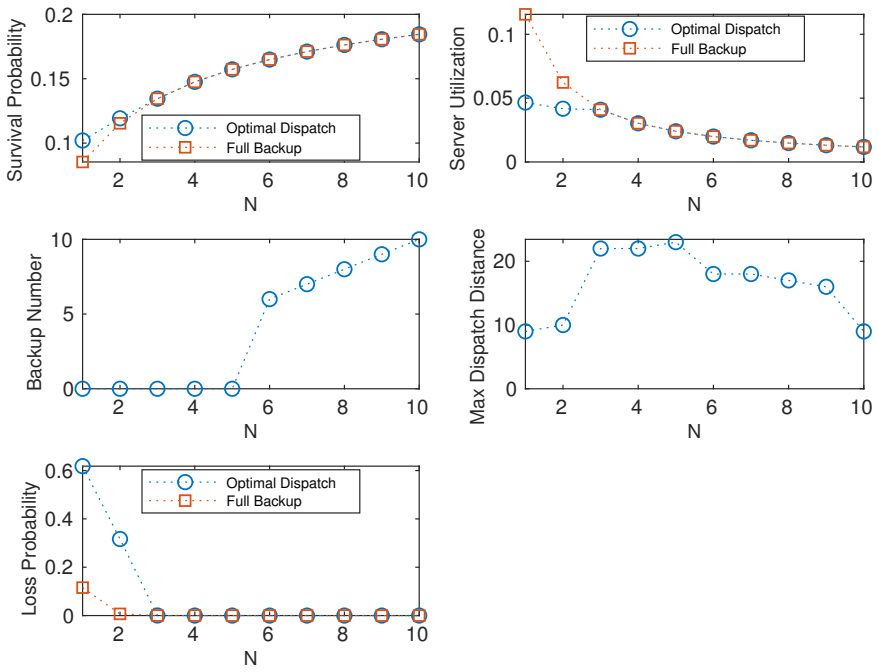


Figure 6.2 Analysis of the optimal dispatch policy for the drone system and $f_A = 1$

of the demand density. This is explained by the fact that, a partial dispatch policy Π_ξ^κ where $\kappa \geq 1$, allows dispatches with travel distances greater than ξ which increase proportionally to the dimension of the service region; therefore, as long as the pure system performance is considered, these policies with $\kappa \geq 1$ will always be inferior to a zero-backup policy that strictly limits the dispatch distances to a maximum of ξ. However, as we will discuss later, policies with $\kappa \geq 1$ will prove important when social aspects such as equity are considered.

The optimal ξ values in Figures 6.2—6.5 were selected based on survival probabilities only. However, in practice, it makes more sense to decide on the optimal value by considering both the survival probability and the corresponding server utilization. This warrants a closer look at the performance of the zero-backup partial dispatch policy for the drone system with different parameters. In Figure 6.6 we observe the variation of the survival probability and the server utilization with ξ in a zero-backup dispatch policy Π_ξ^0 applied to drone system with three drones ($N = 3$) with different service areas ($f_A = 1, 2$), different system loads ($f_\lambda = 1, 5, 10, 20, 40, 80$), and different expected outcomes for lost calls ($U_{\text{loss}} = 0, 0.08$). The server utilization plots for different values of U_{loss} are of course identical and hence are not repeated. The vertical dashed line marks the value of ξ for which the outcome function equals the expected loss outcome, that is $\xi_0 = u^{-1}(U_{\text{loss}})$. For the case with $U_{\text{loss}} = 0$, we have $\xi_0 = \infty$ and hence no marker. We make several key observations. First, although the

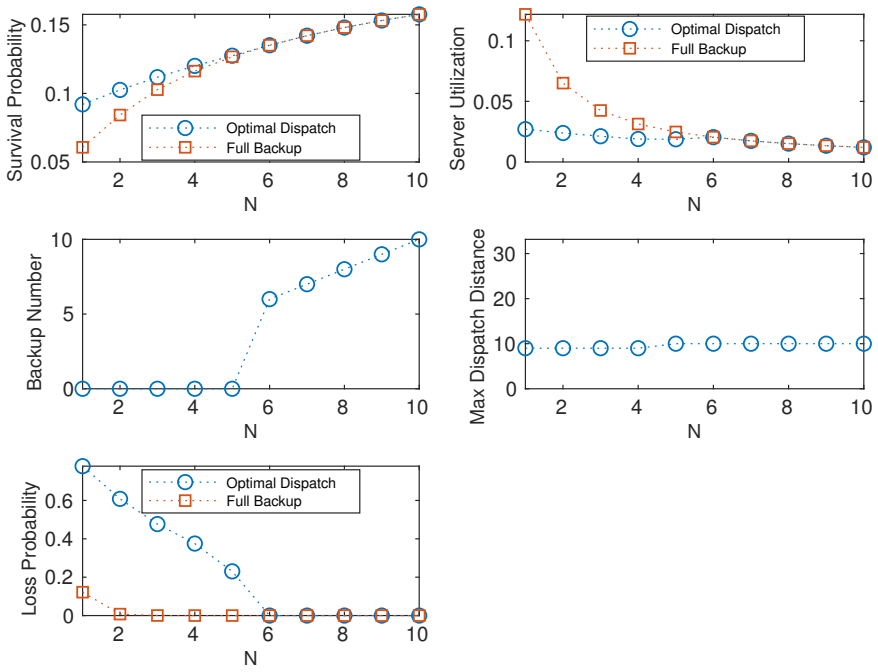


Figure 6.3 Analysis of the optimal dispatch policy for the drone system and $f_A = 2$

optimal ξ value, as expected, is bounded above by ξ_0, the bound is not tight at all as the optimal ξ can be considerably smaller than ξ_0. This means we still need to use the presented model to search for the optimal ξ and the resulting performance although the search interval can be limited to $(0, \xi_0)$ to reduce the computation times. We also note that the gap between ξ and ξ_0 grows as the system becomes more congested. In fact, for a lightly loaded system, say with $f_\lambda = 1$, where servers are rarely busy, a dispatch to a call residing at any distance $x \leq \xi_0$ will always be *worth it* because, first, it results in an outcome better than U_{loss} and second, the temporary absence of the dispatched server is unlikely to have a noticeable adverse effect on the system performance given the low workload. For such a system, any partial dispatch policy within the interval $\xi \in (\xi_1, \xi_{\max}]$ with $\xi_1 \leq \xi_0$ which of course includes the full-backup case (that is $\xi_{\max}$), will perform at least near-optimally. This is reflected in the almost constant portion of the survival probability plot, for example, in the baseline case ($f_\lambda = 1$, $f_A = 1$) for both values of U_{loss} (Figure 6.6-(c) and (e)). For a heavily loaded system, on the other hand, where the negative impact of the temporary absence of a dispatched server on the system performance is significant, a worthwhile dispatch decision will have to yield an outcome significantly better than U_{loss} to justify the drop in the number of available servers. This explains the pushing of the optimal ξ towards 0 as the load (f_λ) increases.

We discussed earlier that, at least in the example application considered here, the best non-

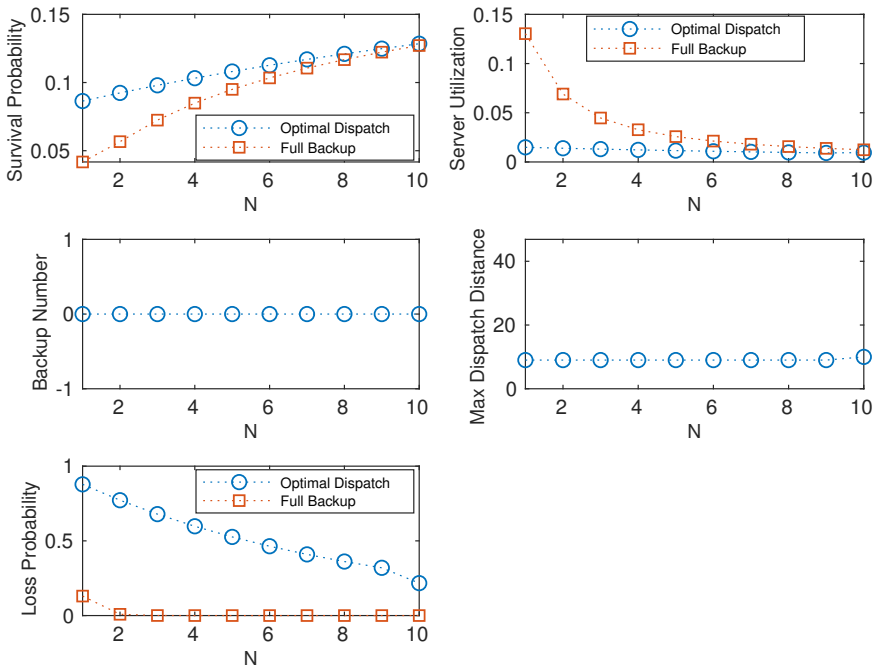


Figure 6.4 Analysis of the optimal dispatch policy for the drone system and $f_A = 4$

zero-backup dispatch policy, that is with $\kappa \geq 1$ will lead to an expected performance inferior to the best zero-backup policy. We attributed this to the fact that a policy with $\kappa \geq 1$ allows long-distance, low-outcome dispatches that become longer in distance and lower in outcome as the service area increases in size. However, even with a zero-backup dispatch policy with the maximum dispatch distance strictly capped at ξ, the size of the service region can significantly shift the distribution of the travel distance towards the larger values and thus significantly decrease the expected outcome without significantly impacting the server workloads. This is clearly seen in Figure 6.6. Despite the server utilization plots for $f_A = 1$ and $f_A = 2$ looking almost exactly identical, the survival probability plots exhibit significantly different behavior: moving from $f_A = 1$ and $f_A = 2$, the best attainable survival probability decreases by 5% to 17% for the case with $U_{\text{loss}} = 0.08$ and by 19% to 29% for the case with $U_{\text{loss}} = 0$. The drop in the performance of the corresponding full-backup policies as the area is doubled is even more significant ranging from 10% to 25% for the case with $U_{\text{loss}} = 0.08$ and from 24% to 34% for the case with $U_{\text{loss}} = 0$. We note that despite the server utilization values changing only slightly with the size of the service area, the survival probabilities corresponding to the optimal partial backup and full-backup policies significantly change as the size of the service area increases. The relative improvement in the survival probability by using an optimal zero-backup dispatch policy for the three-server drone system with the baseline demand ($f_\lambda = 1$), the baseline outcome for lost calls equal to 0 and 0.08 ($U_{loss} = 0,\ 0.08$), and the baseline

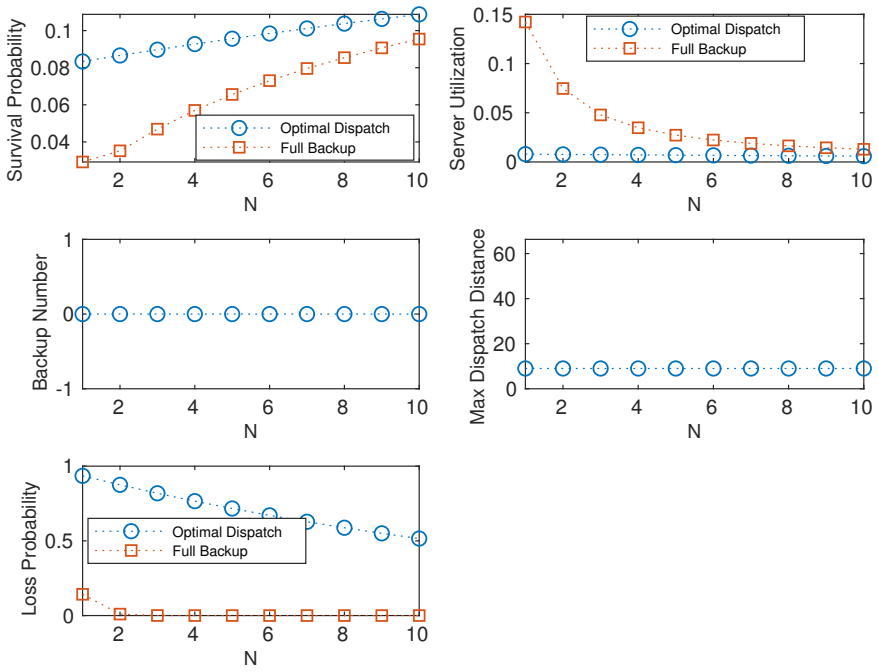


Figure 6.5 Analysis of the optimal dispatch policy for the drone system and $f_A = 8$

and doubled service area ($f_A = 1,\ 2$) is plotted in Figure 6.7. We observe that the impact of using a partial backup policy is proportional to the demand intensity and also to the size of the service area. In other words, we get more benefits from utilizing a partial backup policy with higher arrival rates and larger service areas.

In Figure 6.8, we compare the performance of the drone system with the baseline demand ($f_\lambda = 1$) and a zero-backup dispatch policy with different loss outcomes ($U_{\text{loss}} = 0,\ 0.08$) and service areas ($f_A = 1, 2, 10$). This time, we change the fleet size ($N = 1, 2, 5, 10, 20$) instead of the demand intensity. We observe that almost in all cases it is easy to find a common value of ξ for which the survival probability reaches an optimal or near-optimal value regardless of the fleet size. However, the variation of the survival probability with ξ does not exhibit a common shape and indeed depends on the fleet size.

The scenarios depicted in Figure 6.8 represent light (for $N = 1, 2$) to extremely light loads and thus we might expect the optimal values of ξ to be close to ξ_0. However, this is not generally the case as we observe from the plots of the survival probability versus ξ. For $U_{loss} = 0$ where the nominal value of the optimal ξ is infinity, that is $\xi_0 = \xi_{\max}$, we clearly see in Figure 6.8-(a),(d), and (g) that the survival probability reaches its optimal or near-optimal value for a value of ξ that is generally much smaller than $\xi_{\max}$, that is $\xi^* << \xi_{\max}$. For $U_{loss} = 0.08$, we observe the same pattern with $\xi^* << \xi_0$. We also observe that the

survival probability will drop as ξ is extended beyond ξ^* provided that the ratio of the fleet size to the service area is small enough. For instance, for $f_A = 1$ and fleet sizes of 5, 10, and 20, the full-backup policy performs as good as the optimal zero-backup policy whereas with the service area doubled, that is $f_A = 2$, the full-backup policy is slightly inferior to the best partial policy. Finally, with the service area increased 10-fold ($f_A = 10$), the full-backup policy results in significantly inferior performance compared with the best partial backup policy. In case of the fleet with five drones, the resulting performance is even worse that U_{loss} the expected survival probability provided by the ground system, which suggests no benefit at all to deploying the drone system to aid the ground crew. As we will discuss later, this observation can be used to develop a notion of sufficient fleet size. Finally, we note that despite the server utilization values changing only slightly with the size of the service area, the survival probabilities corresponding to the optimal partial backup and full-backup policies significantly change as the size of the service area increases. In Figure 6.9 we see the relative improvement in the survival probability by using an optimal zero-backup dispatch policy for the drone system with the baseline demand ($f_\lambda = 1$), the baseline outcome for lost calls ($U_{loss} = 0.08$), a fleet of one to seven drones, and the service area sequentially doubled ($f_A = 1, 2, 4, 8$). We observe that the impact of using a partial backup policy is proportional to the average number of servers per unit area of the service region. In other words, we get more benefits from utilizing a partial backup policy with smaller fleet sizes and larger service areas.

6.3.2 Queuing ESS

In the second example, we consider a queuing emergency service system operating as the primary service provider to a service area. An example of such a scenario will be a network of mobile medical units providing medical services, not necessarily urgent, to a community in an underdeveloped region. The fact that the system is the sole service provider to the region and the non-urgent nature of the service provided justify allowing waiting queues. Nevertheless, we assume that the quickness of service is still important as reflected in the chosen outcome function given as (6.38) and plotted in Figure 6.1. This outcome function represents the quality of service as a piece-wise function of the response time. For a response within 15 minutes from the call arrival, the outcome function will be equal to 1 reflecting a perfect service, and for responses taking more than 15 minutes, the service quality drops linearly until it reaches zero at the 45 minutes mark. The summary of the parameters chosen for this example are given in Table 6.2.

Table 6.2 Parameters used in the second example

Parameter	Value
Baseline area of service region	500 km^2
Total arrival rate	1 per hour
Average scene time	30 minutes
Average chute time	10 seconds
Average dispatch time	60 seconds
Probability of transport	0
Drones' acceleration	25 km/hour.min
Drones' top speed	30 km/hour

$$u(t) = \begin{cases} 1 & t \leq 15 \\ 1 - (t-15)/30 & 15 \leq t \leq 45 \\ 0 & t > 45 \end{cases} \tag{6.38}$$

The optimal partial back up dispatch policies and the corresponding performance metrics for the baseline demand ($f_\lambda = 1$), different fleet sizes ($N = 2, 3, \cdots, 10$), the service area sequentially doubled ($f_A = 1, 2, 4, 8$), and the expected loss outcome of 0 and 0.25 ($U_{\text{loss}} = 0, 0.25$) are plotted and compared with the full-backup policy in Figures 6.10—6.13. We make several observations similar to the loss system considered in the previous example: 1) For the partial backup dispatch policy to be effective, the fleet size should be below a certain threshold; 2) Similarly, the larger the service area, the more effective the use of a partial backup dispatch policy in obtaining the best performance possible. 3) For the range of fleet sizes where the full-backup policy is not optimal, a zero-backup dispatch policy is optimal.

In Figure 6.14 we closely examine the performance of the zero-backup policies applied to the queuing system with different loads and service areas. Again, similar trends are observed: 1) The optimal maximum dispatch distance, ξ^* is generally smaller than $\xi_0 = u^{-1}(t)$ and the gap $|\xi^* - \xi_0|$ grows with load intensity (f_λ). 2) Despite the server utilization values only slightly increased with the service area, the expected service quality significantly degrades as the service area is doubled.

Figure 6.16 demonstrates the performance of the zero-backup policy applied to the queuing EMS with different fleet sizes, service areas, and expected outcomes for the lost calls. Similarly to the case with the loss drone system, we observe that: 1) The optimal or near-optimal values of ξ do not change much with the fleet size, and in fact, it is no difficult to find a common value of ξ for which a near optimal performance is achieved across the different fleet sizes;

and this explains the relatively constant value of ξ^* observed in Figure Figures 6.10—6.13. However, the shape of the variation of the service quality with ξ strongly depend on the fleet size with near optimal values reached for values of ξ considerably smaller than the absolute optimum value, ξ^*, particularly in cases of smaller fleet sizes and thus higher congestion; 2) The performance of the system with large values of ξ, and in particular for the full-backup policy ($\xi = \xi_{\max}$) will rapidly drop as the average number of servers per unit service area decreases. For example, in the extreme case of one or two servers and the service area scaled 10-fold, the full-backup policy will be in fact completely inefficient in the scenario where there is another system operating in tandem and providing an expected service quality of $U_{\text{loss}} = 0.25$; 3) As before, while the average server utilization only slightly changes with the service area, the expected service quality corresponding to the optimal partial and full backup policies significantly change as the service area is scaled up.

Finally, the relative improvements in service quality when an optimal zero-backup policy is applied to this example queuing EMS with three servers and different service areas and arrival rates is plotted in Figure 6.15, and with varying fleet sizes, service areas and the baseline arrival rate ($f_\lambda = 1$) in Figure 6.17. Similar to the drone system, we observe very significant potentials for enhancing the system performance via using a partial backup dispatch policy instead of full backups. The plots reflect the fact that the potential for performance increases with increasing demand, increasing service area, and decreasing fleet size.

6.4 Remarks

We conclude by remarks on the validation, potential applications and possible developments of the descriptive model introduced in this chapter.

The validation of the model can be done at different levels. In Appendix C.1, we compare the outputs of the model applied to the example applications given in the previous section with the output obtained from an idealized simulation model. The ideal simulation model used in these comparisons strictly follows the uniformity assumptions we made in our development of the mathematical model. That is, the demand and response units are uniformly distributed over the service area with the shape we assumed for the corresponding metric as in Chapter 5. The goal of these comparisons is to validate the correctness of the mathematical expressions and evaluate the magnitude of the errors introduced by using the Poisson approximation of distance to neighbors instead of the actual edge-corrected distributions. However, to truly gauge the validity of the theoretical framework presented in modelling practical situations, we need to compare the performance of the system predicted by the model to predictions from detailed simulation of large sets of scenarios with arbitrary demand distributions, relocation

schemes, boundary shapes, etc. Such validation studies are not reported here, instead, we propose the descriptive model as a standalone theoretical approach to be used in primary and efficient evaluation of partial backup dispatch policies for ESSs with dynamic relocation. Comprehensive validation tests can then be performed to establish the relationship between this theoretical model and observations of the detailed simulations of realistic instances of ESSs.

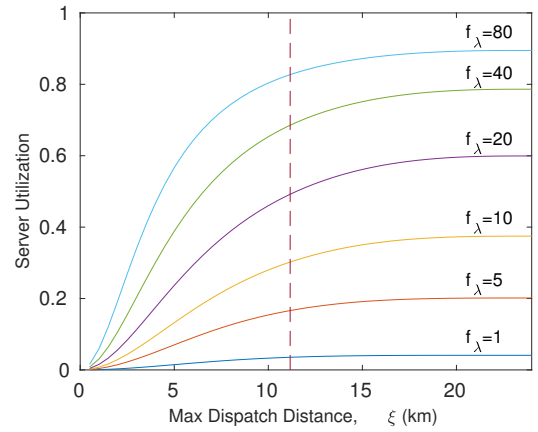
In the extended approximate hypercube model presented in Chapter 4 for analysis of static deployments of ESSs with partial backups, we allowed for heterogeneous demand, categorized, for instance, by priorities. The model presented in this chapter can also be extended to include priorities. In that case, the distribution of the number of busy servers obtained from the queue with partial service, will be approximate; however, the approximation errors will be negligible as discussed in Chapter 4. In this extended model, a partial backup dispatch policy, a total arrival rate, and an outcome function corresponding to each call type will be specified as inputs to the algorithm, in addition to the service area and the fleet size. By exploring the space of possible values of κ and ξ corresponding to each call type, as we did in the example applications, one can arrive at theoretical optimal partial dispatch policies to maximize the overall outcome by blending the individual outcomes with a given mixing weight.

Looking at the plots of the expected server utilization and survival probability or quality of service given for the examples considered in the previous section, may reveal a potential application of the model to generate dynamic policies to control the maximum dispatch distance as the system transitions into different states where the system state is defined as the number of available servers. This is supported by the hypothesis that, as far as the performance of the system per a given partial backup dispatch policy is considered, a snapshot of the system with a fleet size of N servers with $M < N$ free servers, can be treated, at least approximately, as a system with a fleet size of M. Now, if this hypothesis is true, which we conjecture it to be so, then instead of directly searching through a solution space of $2N$ dimensions (N states of $1, \cdots, N$ busy servers, each with a corresponding κ and ξ) for an optimal *dynamic* partial backup dispatch policy, we can take the *static* optimal partial dispatch policies obtained for a range of fleet sizes, and then implement it in real operation (or a simulation of) by simply applying the static optimal policy corresponding to a fleet size of n whenever the system hits the state of n free servers, regardless of the actual fleet size.

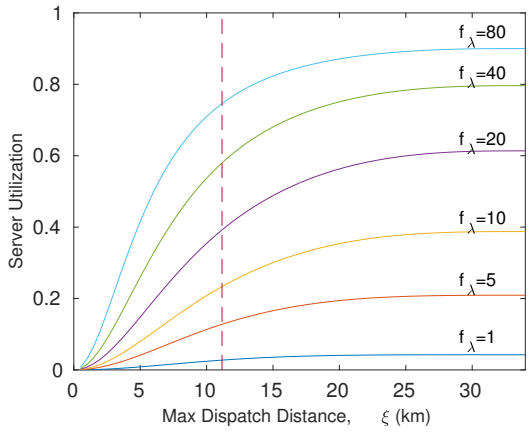
We remark that the limit on the maximum dispatch distance might be imposed by the physical limitations of the response units or obvious operational facts, instead of being a means to achieve potential performance improvements. As a prime example, one can think

of a system comprised of aerial drones that typically have a limited flight range depending on the battery capacity, or as mentioned numerous time before, dispatcher's logical tendency to not assign units to travel unreasonably long distances in response to time critical emergencies. In these cases, an analysis with the model presented in this chapter, can help realistically represent these important physical or operational characteristics of the system. For instance, repeated applications of the algorithm will allow us to determine the minimum number of such range-limited drones to provide a minimum level of expected outcome.

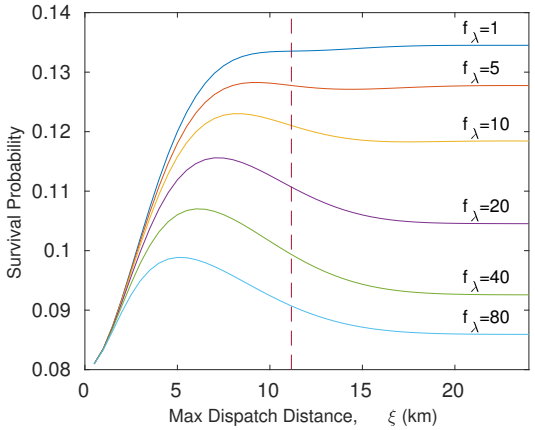
These types of fleet sizing analyses, however, are not limited to networks of range-limited drones. In fact, the limited number of input parameters needed for the algorithm and the short execution times compared to simulation, may suggest fleet-sizing as a promising application of the model. Obviously, the expected outcome of an ESS will increase with the increasing fleet size at an increasingly slower rates. This leaves it to the system manager to decide on the upper bound on the fleet size based on the trade-off between the cost of operating larger fleets versus the expected performance gains. As a reasonable lower bound for the fleet size, we suggest the minimum fleet size for which no potential performance improvements in terms of system workload or the corresponding expected outcome will be possible by introducing a partial backup dispatch policies. This suggested minimum fleet size makes sense as it implies that the number of response units is large enough to ensure, with a certain reliability, that a free response unit will be available at a short enough distance from a call origin to provide an effective intervention. In other words, the probability of the closest free response unit being too far from a call origin becomes too small to adversely affect the overall performance of the system to any noticeable extent. Adopting this notion of sufficient fleet size, it appears that a fleet of at least three response units is required in both baseline application examples (that is without scaling the service area, or $f_A = 1$). However, with the baseline demand density reduced by a factor of 2, that is the case with $f_A = 2$, a minimum of six drones, and three or four ground vehicles can be inferred from the plots in Figures 6.3 and 6.11 for the drone and ground vehicle fleets, respectively.



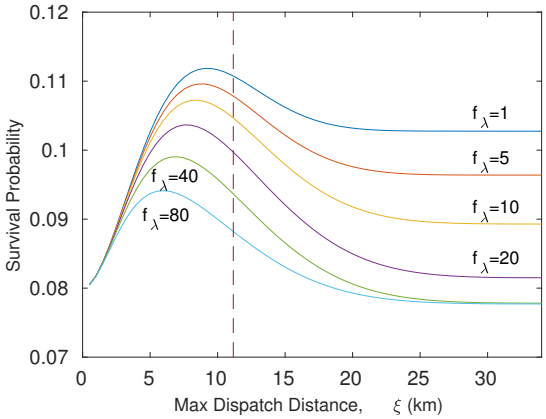
(a) $f_A = 1$



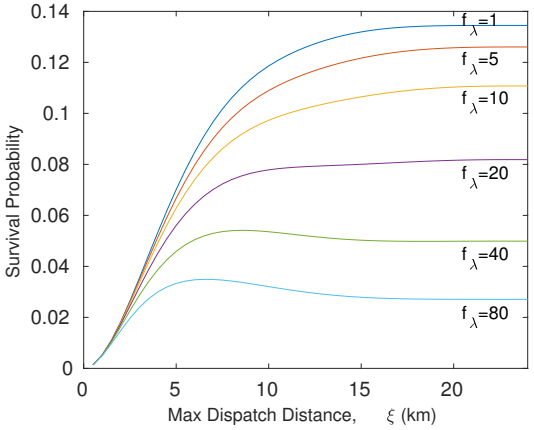
(b) $f_A = 2$



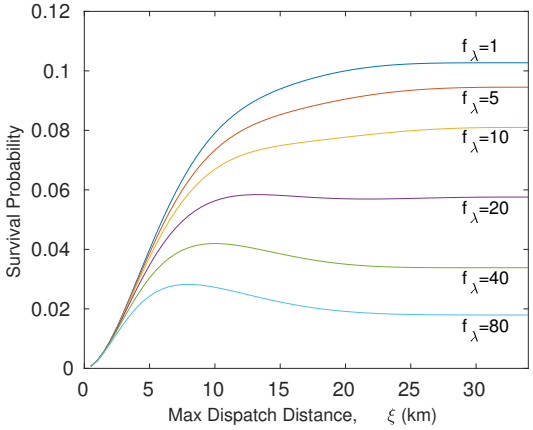
(c) $U_{\text{loss}} = 0.08$, $f_A = 1$



(d) $U_{\text{loss}} = 0.08$, $f_A = 2$

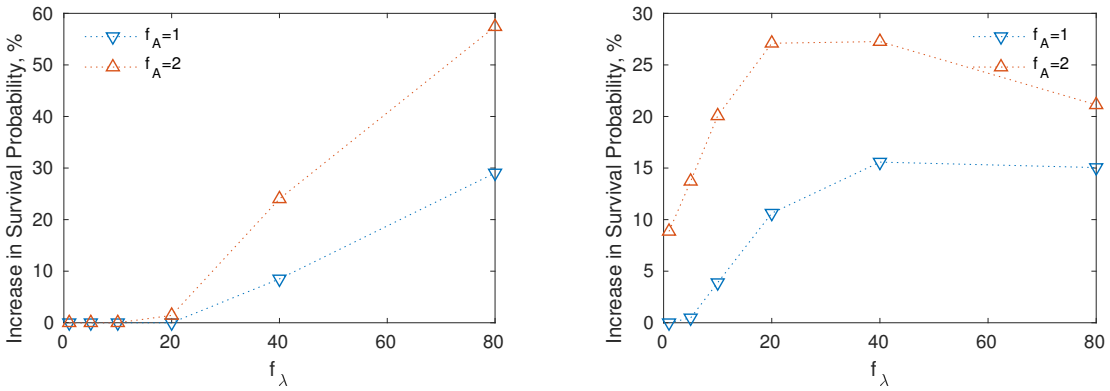


(e) $U_{\text{loss}} = 0$, $f_A = 1$



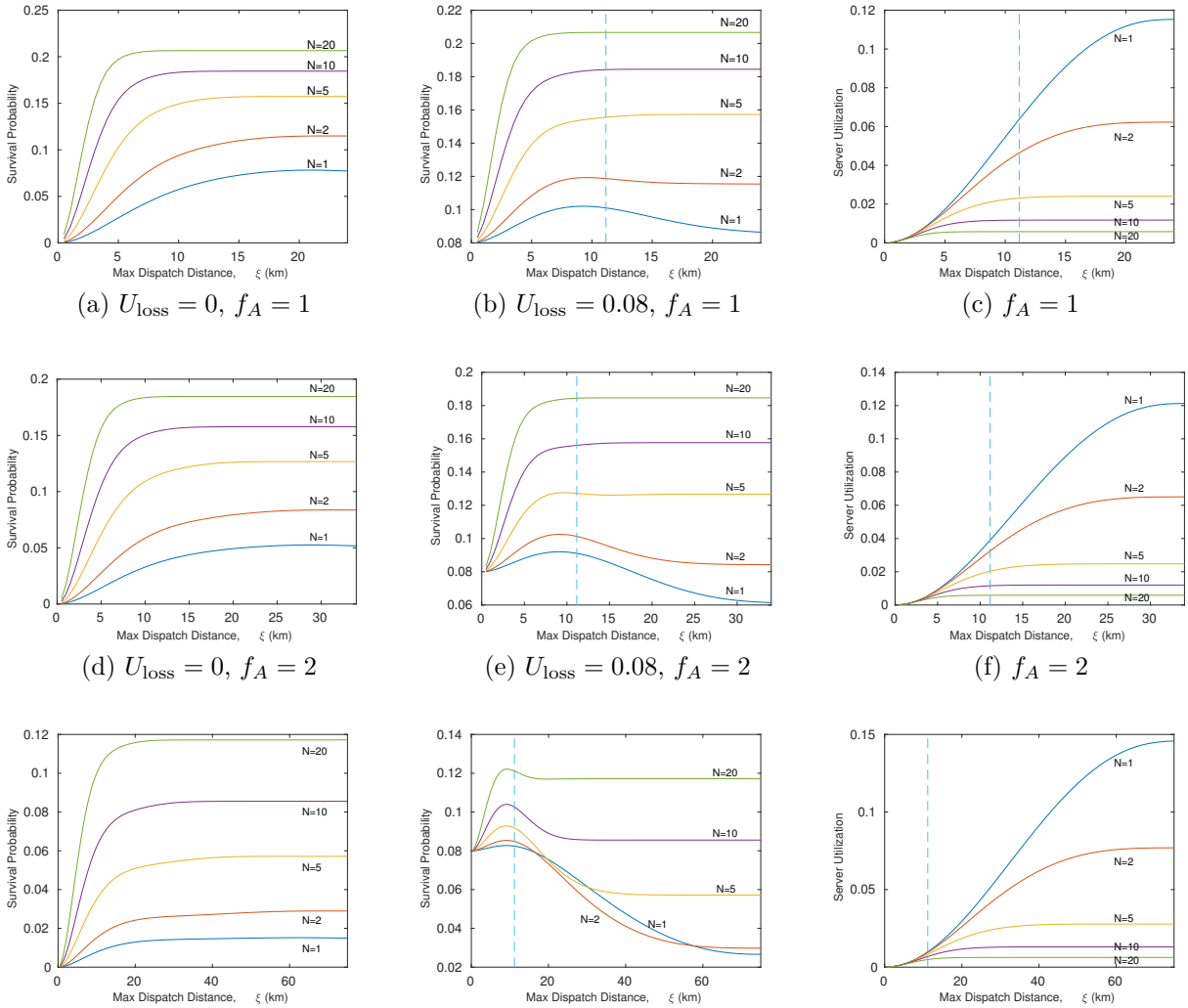
(f) $U_{\text{loss}} = 0$, $f_A = 2$

Figure 6.6 Performance of the zero-backup drone system with different loads



(a) $U_{\text{loss}} = 0$ (b) $U_{\text{loss}} = 0.08$

Figure 6.7 Performance improvement using zero-backup dispatch policy in the drone system with three servers



(a) $U_{\text{loss}} = 0,\ f_A = 1$ (b) $U_{\text{loss}} = 0.08,\ f_A = 1$ (c) $f_A = 1$

(d) $U_{\text{loss}} = 0,\ f_A = 2$ (e) $U_{\text{loss}} = 0.08,\ f_A = 2$ (f) $f_A = 2$

(g) $U_{\text{loss}} = 0,\ f_A = 10$ (h) $U_{\text{loss}} = 0.08,\ f_A = 10$ (i) $f_A = 10$

Figure 6.8 Server utilization and survival probability for the zero-backup drone system with different fleet sizes and service areas

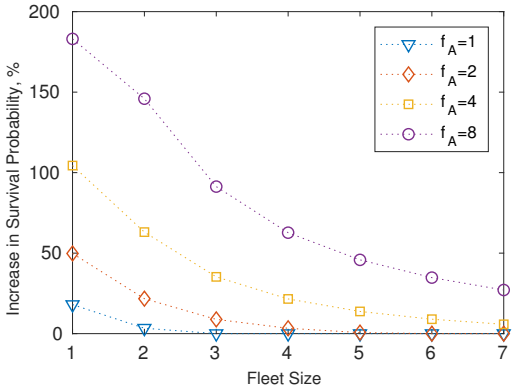


Figure 6.9 Performance improvement using zero-backup dispatch policy in the drone system with baseline demand ($f_\lambda = 1$) and $U_{\text{loss}} = 0.08$

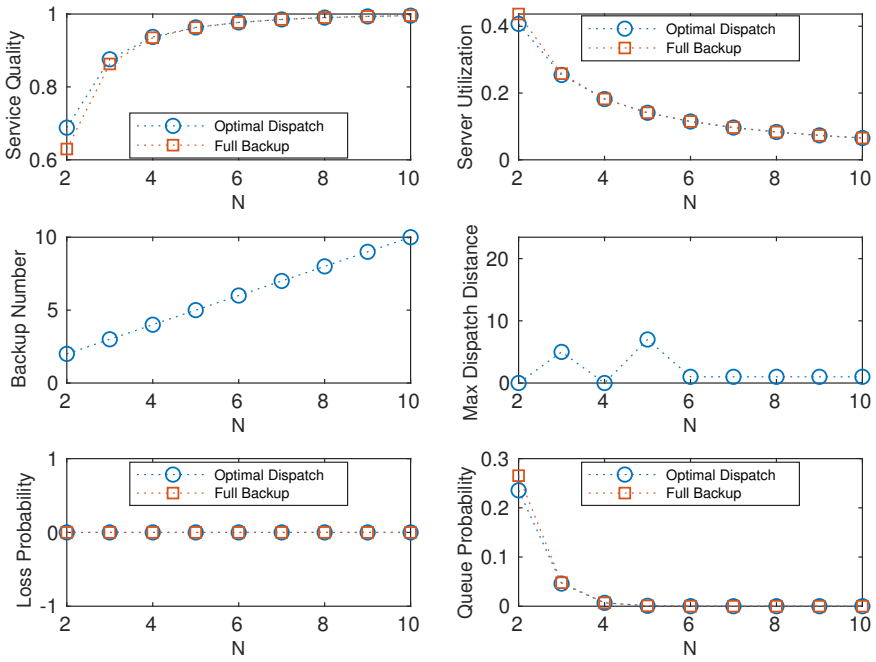


Figure 6.10 Analysis of the optimal dispatch policy for the queuing EMS and $f_A = 1$

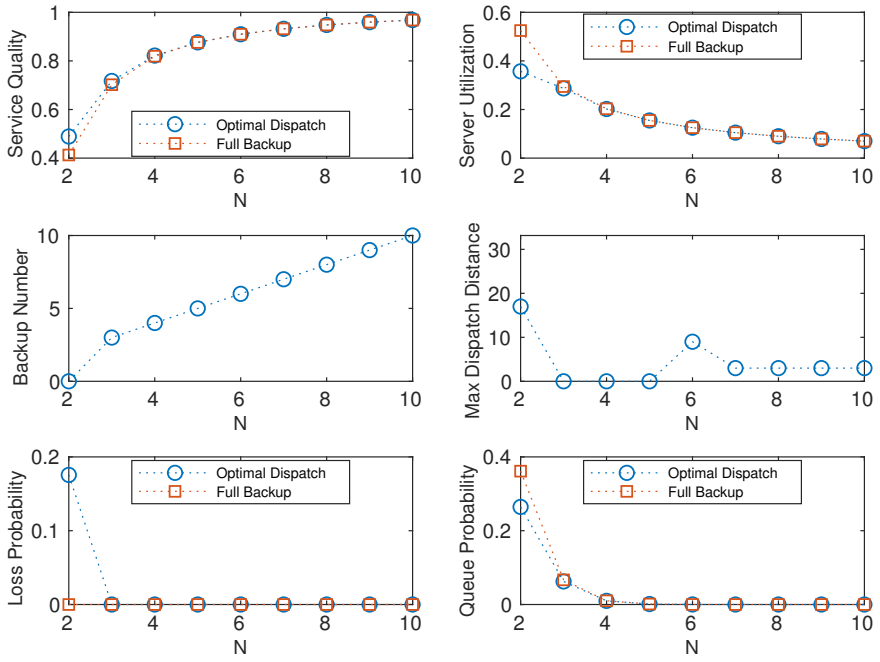


Figure 6.11 Analysis of the optimal dispatch policy for the queuing EMS and $f_A = 2$

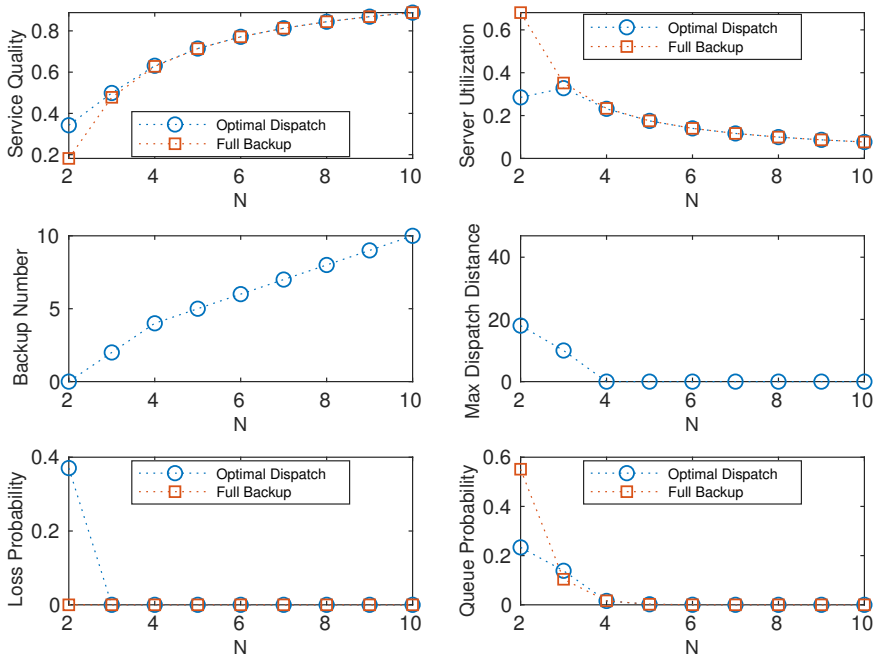


Figure 6.12 Analysis of the optimal dispatch policy for the queuing EMS and $f_A = 4$

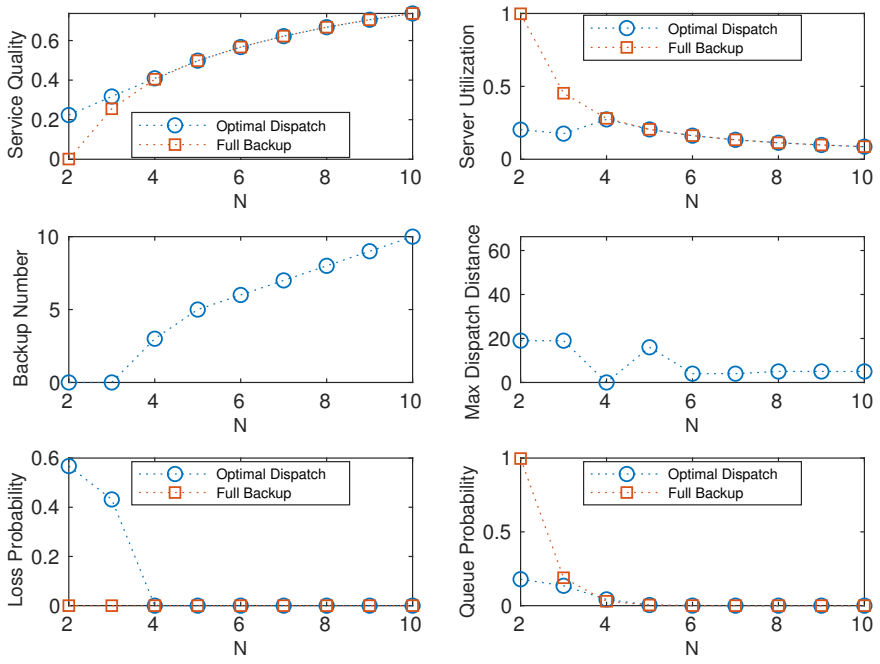
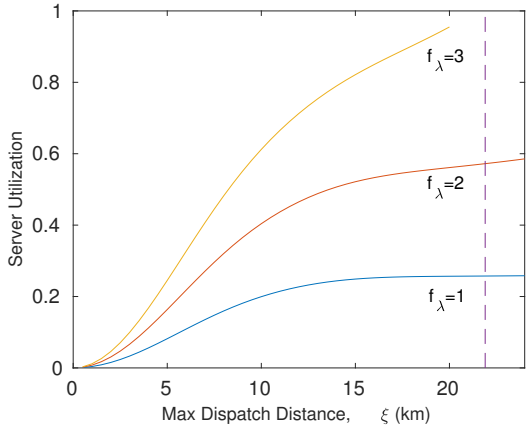
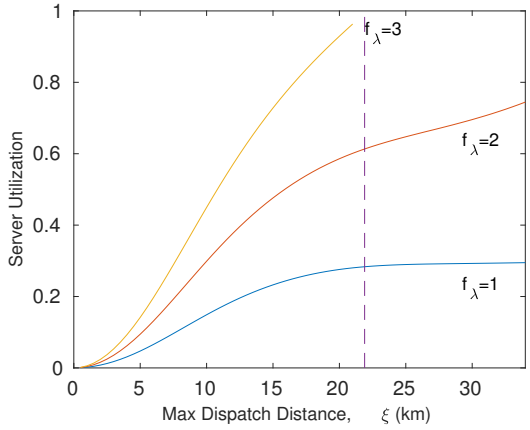


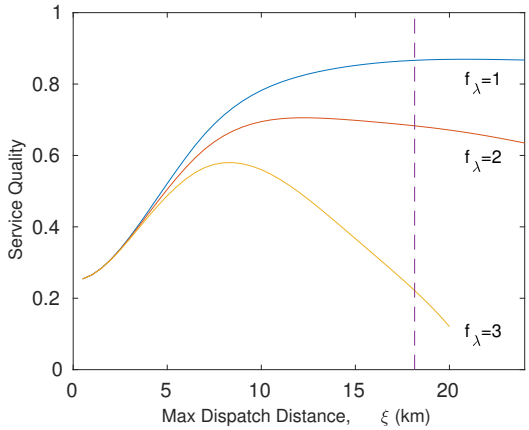
Figure 6.13 Analysis of the optimal dispatch policy for the queuing EMS and $f_A = 8$



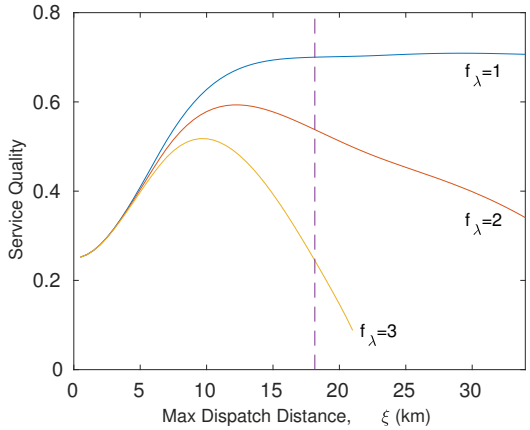
(a) $f_A = 1$



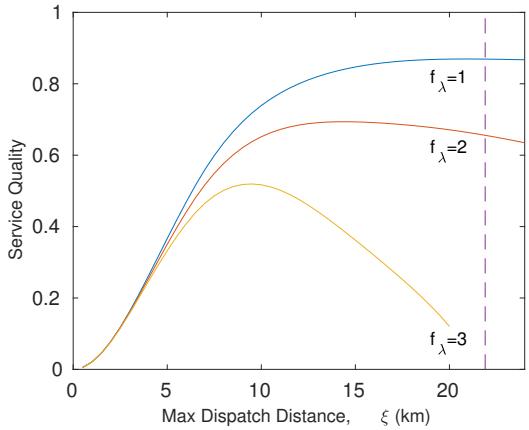
(b) $f_A = 2$



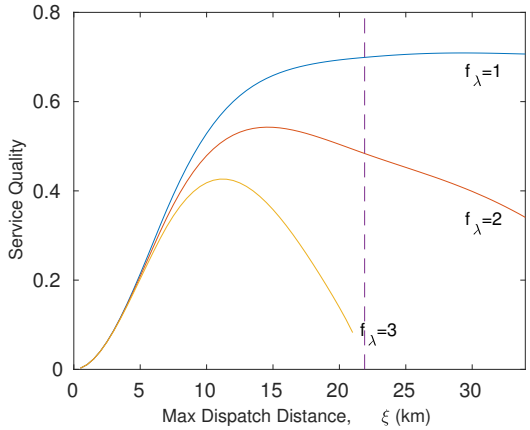
(c) $U_{\text{loss}} = 0.25,\ f_A = 1$



(d) $U_{\text{loss}} = 0.25,\ f_A = 2$

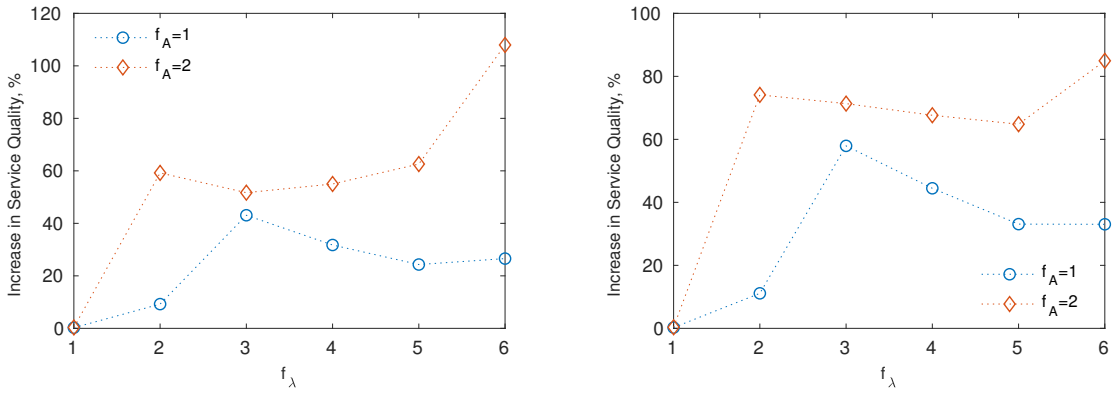


(e) $U_{\text{loss}} = 0,\ f_A = 1$



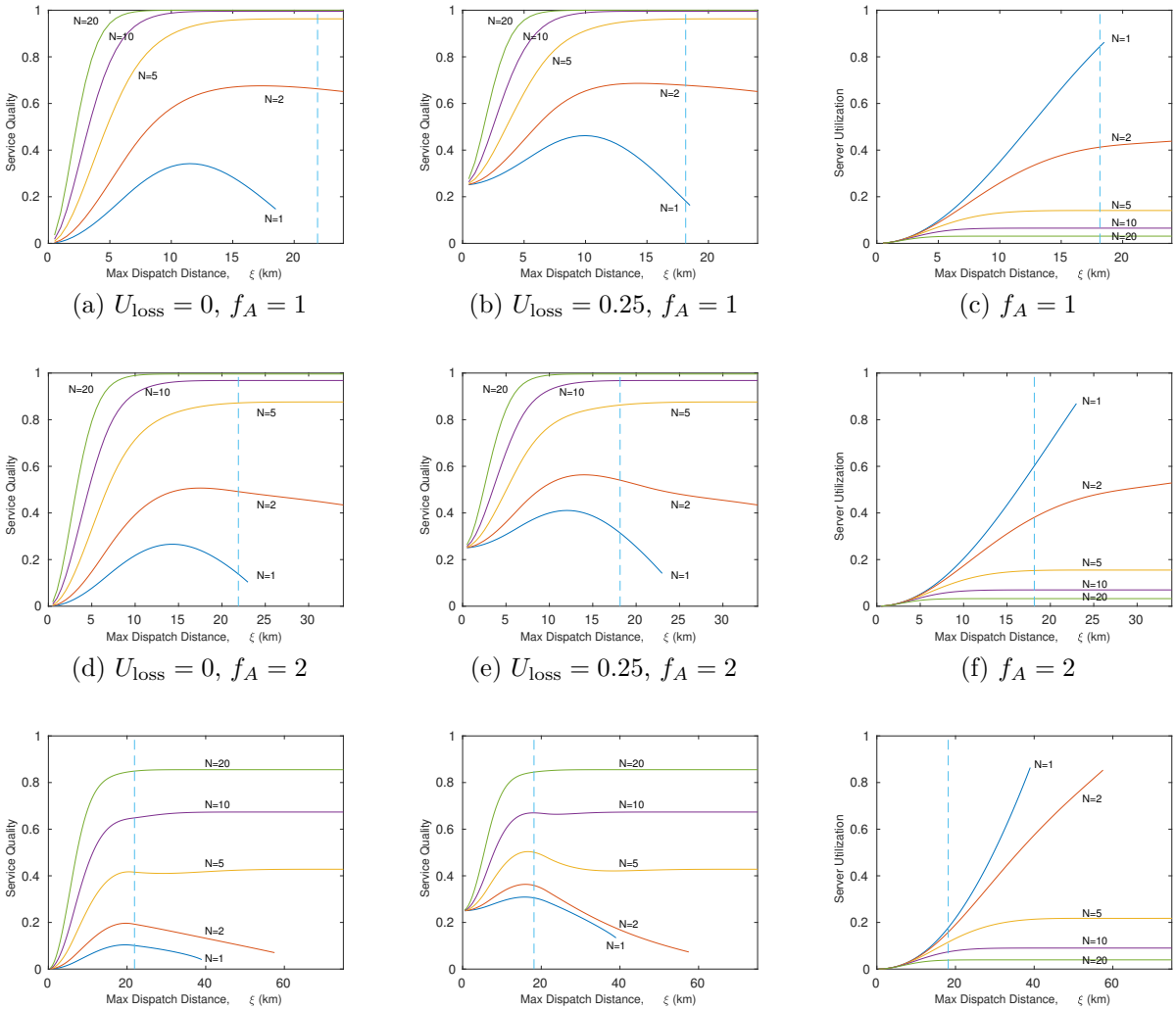
(f) $U_{\text{loss}} = 0,\ f_A = 2$

Figure 6.14 Service quality for the zero-backup queuing EMS



(a) $U_{\text{loss}} = 0$ (b) $U_{\text{loss}} = 0.25$

Figure 6.15 Performance improvement using zero-backup dispatch policy in the queuing EMS with three servers

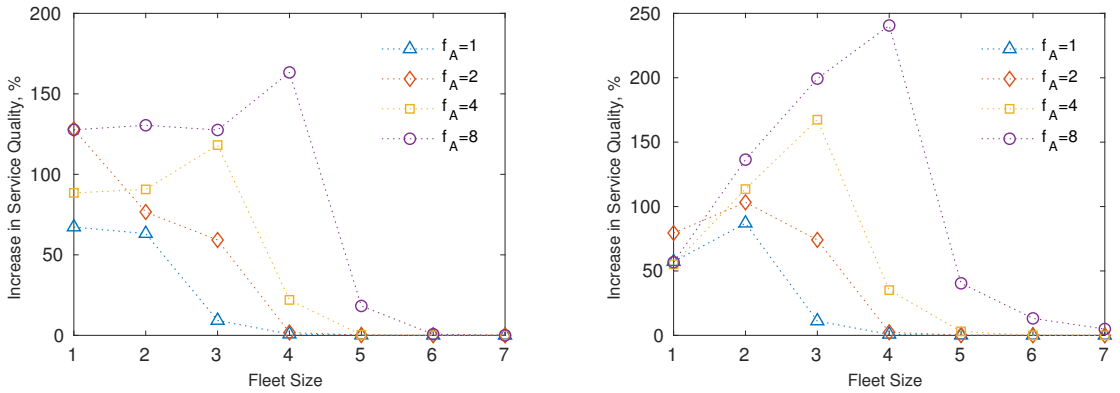


(a) $U_{\text{loss}} = 0,\ f_A = 1$ (b) $U_{\text{loss}} = 0.25,\ f_A = 1$ (c) $f_A = 1$

(d) $U_{\text{loss}} = 0,\ f_A = 2$ (e) $U_{\text{loss}} = 0.25,\ f_A = 2$ (f) $f_A = 2$

(g) $U_{\text{loss}} = 0,\ f_A = 10$ (h) $U_{\text{loss}} = 0.25,\ f_A = 10$ (i) $f_A = 10$

Figure 6.16 Server utilization and service quality for the zero-backup queuing ems with different fleet sizes and service areas



(a) $U_{\text{loss}} = 0$ (b) $U_{\text{loss}} = 0.25$

Figure 6.17 Performance improvement using zero-backup dispatch policy in the queuing EMS with baseline demand

CHAPTER 7 CONDITIONAL EXTENSION TO LITTLE'S LAW

We consider queuing systems where customers may observe different system states during their sojourn and derive various relations between the waiting times spent and queue lengths observed during a certain system state or for customers who observe that state upon arrival. A degree of unification is provided to queuing theory as Little's formula and its distributional form are recovered as special cases.

In a queuing system with partial admissions waiting lines may form at any time when the system is not empty. This naturally motivates us to derive the following conservation results relating the queue lengths and weighting times observed at particular system states or pertaining to the sub-class of customers who arrive at the system at a specific system state

7.1 Results

We consider a general queuing system and let t_i and t_i' for $i = 1, 2, \cdots$ be the i-th customer's arrival and departure epochs, respectively; the waiting times are then given by $w_i = t_i' - t_i$. The number of arrivals in the interval $[0, t]$ is denoted by $\lambda(t) = max\{i : t_i < t]\}$; thus the arrival rate is given by $\lambda = \lim_{t\to\infty} t^{-1}\lambda(t)$. We assume that $\lambda(t) \to \infty$ only if $t \to \infty$, so there is a finite number of arrivals in every finite time interval. The indicator function $\mathbf{1}_i(t)$ implies the presence of the i-th customer in the system at time t; that is $\mathbf{1}_i(t) = 1 \Leftrightarrow t \in [t_i, t_i']$. Thus $N(t) = \sum_{i=0}^{\infty} \mathbf{1}_i(t)$ will be the number of customers in the system at time $t \geq 0$.

Suppose that a state of the system or the environment is given by a continuous-time process $\{Z(t) \in S; t \geq 0\}$ with S the set of all possible states. The state indicator function $\bar{\mathbf{1}}_s(t)$ is defined to imply the system to be in a given subset of states $s \subset S$ at time t; that is $\bar{\mathbf{1}}_s(t) = 1 \Leftrightarrow Z(t) \in s$. Thus $P_s(t) = t^{-1}T_s(t) = t^{-1}\int_0^t \bar{\mathbf{1}}_s(\tau)\, d\tau$ gives the average fraction of the time interval $[0, t]$ spent in state s.

Looking at the arrival epochs t_i, the fraction of the first k customers who observe the system to be in state s just prior to their arrival is given by $\pi_{s,k}^{-} = k^{-1}K_{s,k} = k^{-1}\sum_{n=1}^{k} \bar{\mathbf{1}}_s(t_n-)$. The average number in the system of such customers up to time t is then given by $L_s(t) = t^{-1}\int_0^t N_s(\tau)\, d\tau$ with $N_s(t) = \sum_{n=1}^{\infty} \mathbf{1}_n(t)\bar{\mathbf{1}}_s(t_n-)$ the instantaneous number in the system of these customers. In contrast, restricting our observation to periods where the system is in state s, we denote the average number of customers in the system observed during such periods up to time t by $\bar{L}_s(t) = T_s(t)^{-1}\int_0^t N(\tau)\bar{\mathbf{1}}_s\, d\tau$.

Similarly for the waiting times, we denote by $W_{s,k} = K_{s,k}\sum_{n=1}^{k} w_n\bar{\mathbf{1}}_s(t_n-)$ the average waiting

time of the first k customers who observed state s upon arrival; and let $\bar{W}_{s,k} = k^{-1}\sum_{n=1}^{k} w_{n,s}$ be the average time the first k customers spend in state s where $w_{i,s} = \int_{t_i}^{t'_i} \mathbf{1}_i(t)\bar{\mathbf{1}}_s(t)\,dt$ is the length of time the i-th customer spends in state s.

Finally, letting t and k tend to infinity, we define the limiting averages of the numbers in the system and of the waiting times as $L_s = \lim_{t\to\infty} L_s(t)$, $\bar{L}_s = \lim_{t\to\infty} \bar{L}_s(t)$, $W_s = \lim_{k\to\infty} W_{s,k}$, and $\bar{W}_s = \lim_{k\to\infty} \bar{W}_{s,k}$. Likewise, the customer-average pre-arrival state frequency and the time-average state probabilities will be given by $\pi_s^- = \lim_{k\to\infty} \pi_{s,k}^-$ and $P_s = \lim_{t\to\infty} P_s(t)$, respectively. Throughout the paper we implicitly assume the appeal to an appropriate ergodic theorem or Strong Law of Large Numbers to establish the coincidence of these (sample-path) limiting averages and the corresponding quantities in the stationary framework (for example that $P_s = \Pr[\mathbf{Z}(t) \in s]$, or $W_s = \mathrm{E}[\mathbf{W}_s]$ where $\mathbf{Z}(t)$ and $\mathbf{W}_s$ are stationary random variables. For details see El-Taha and Stidham Jr (2012)).

We are now in position to state our first result relating W_s and L_s.

Theorem 4. *If the customer-average waiting time in a queuing system is finite then L_s exists and is finite and given by $L_s = \pi_s^- \lambda W_s$.*

Proof. The assumption of finite average waiting time implies $\lim_{k\to\infty} \sum_k^\infty k^{-1} w_k < \infty$; therefore, if we write

$$\lim_{n\to\infty} \frac{w_n}{t_n} = \lim_{n\to\infty} \frac{w_n}{n}\frac{n}{t_n} \le \lim_{n\to\infty} \frac{w_n}{n}\frac{\lambda(t_n)}{t_n},$$

for finite λ the right-hand side of the inequality tends to zero and we have $\lim_{n\to\infty} {t_n}^{-1} w_n = 0$. This enables us to apply the original sample-path version of the $H = \lambda G$ law in Heyman and Stidham Jr (1980) with $l_n = w_n$ and $f_n(t) = \bar{\mathbf{1}}_s(t_n-)\mathbf{1}_n(t)$ for $n = 1, 2, \cdots$. We write

$$\begin{aligned} G &= \lim_{k\to\infty} \frac{1}{k}\sum_{n=1}^{k} \int_0^\infty f_n(t)\,dt = \lim_{k\to\infty} \frac{1}{k}\sum_{n=1}^{k} \int_0^\infty \bar{\mathbf{1}}_s(t_n-)\mathbf{1}_n(t)\,dt \\ &= \lim_{k\to\infty} \frac{1}{k}\sum_{n=1}^{k} \bar{\mathbf{1}}_s(t_n-) \int_0^\infty \mathbf{1}_n(t)\,dt = \lim_{k\to\infty} \frac{1}{k}\sum_{n=1}^{k} \bar{\mathbf{1}}_s(t_n-) w_n \\ &= \lim_{k\to\infty} \frac{1}{k}\{W_{s,k}\sum_{n=1}^{k} \bar{\mathbf{1}}_s(t_n-)\} = \lim_{k\to\infty} \frac{1}{k}\sum_{n=1}^{k} \bar{\mathbf{1}}_s(t_n-) \lim_{k\to\infty} W_{s,k} \end{aligned}$$

where the last two limits are recognized as π_s^- and W_s, respectively; thus $G = \pi_s^- W_s$ and is finite since W and hence W_s is finite. On the other hand,

$$H = \lim_{t\to\infty} \frac{1}{t} \int_0^t \sum_{n=1}^{\infty} f_n(\tau)\, d\tau = \lim_{t\to\infty} \frac{1}{t} \int_0^t \sum_{n=1}^{\infty} \bar{\mathbf{1}}_s(t_n-)\mathbf{1}_n(\tau)\, d\tau = \lim_{t\to\infty} \frac{1}{t} \int_0^t N_s(\tau)\, d\tau = L_s$$

Thus $H = \lambda G$ yields $L_s = \pi_s^- \lambda W_s$. □

For systems where arrivals see time averages (the ASTA property) we obtain the following result as an immediate corollary to Theorem 4. For a review of ASTA conditions we refer the reader to Melamed and Yao (1995), Melamed and Whitt (1990).

Theorem 5. *If the customer-average waiting time in a queuing system with ASTA property is finite then L_s exists and is finite and given by $L_s = P_s\lambda W_s$.*

Proof. The result follows from Theorem 4 and $P_s = \pi_s^-$ from ASTA. □

The next results establishes a relation between $\bar{L}_s$ and $\bar{W}_s$.

Theorem 6. *If the average waiting time in a queuing system is finite, then $\bar{L}_s$ exists and is finite, and we have $P_s\bar{L}_s = \lambda\bar{W}_s$.*

Proof. Define $f_n'(t) = \bar{\mathbf{1}}_s(t)\mathbf{1}_n(t)$, for $n = 1, 2, \cdots$. We readily observe that $f_k'(t) = 0$ for $k \notin [t_k, w_n]$. Using the same arguments as in the proof of Theorem 4 we conclude that $\lim_{n\to\infty} n^{-1}w_n = 0$ and then apply the $H = \lambda G$ law with w_n as l_n, for $n = 1, 2, \cdots$; we write

$$\begin{aligned} G' &= \lim_{k\to\infty} \frac{1}{k} \sum_{n=1}^{k} \int_0^{\infty} f_n'(t)\, dt = \lim_{k\to\infty} \frac{1}{k} \sum_{n=1}^{k} \int_0^{\infty} \bar{\mathbf{1}}_s(t)\mathbf{1}_n(t)\, dt \\ &= \lim_{k\to\infty} \frac{1}{k} \sum_{n=1}^{k} \int_{t_k}^{t_k'} \bar{\mathbf{1}}_s(t)\mathbf{1}_n(t)\, dt = \lim_{k\to\infty} \frac{1}{k} \sum_{n=1}^{k} w_{n,s} = \bar{W}_s \end{aligned}$$

On the other hand

$$\begin{aligned} H' &= \lim_{t\to\infty} \frac{1}{t} \int_0^t \sum_{n=1}^{\infty} f_n'(\tau)\, d\tau = \lim_{t\to\infty} \frac{1}{t} \int_0^t \sum_{n=1}^{\infty} \bar{\mathbf{1}}_s(\tau)\mathbf{1}_n(\tau)\, d\tau \\ &= \lim_{t\to\infty} \frac{1}{t} \int_0^t \bar{\mathbf{1}}_s(\tau)\{\sum_{n=1}^{\infty} \mathbf{1}_n(\tau)\}\, d\tau = \lim_{t\to\infty} \frac{1}{t} \int_0^t \bar{\mathbf{1}}_s(\tau)N(\tau)\, d\tau \\ &= \lim_{t\to\infty} \frac{1}{t}\bar{L}_s(t) \int_0^t \bar{\mathbf{1}}_s(\tau)\, d\tau = \lim_{t\to\infty} \bar{L}_s(t) \lim_{t\to\infty} \frac{1}{t} \int_0^t \bar{\mathbf{1}}_s(\tau)\, d\tau \end{aligned}$$

Thus $H' = \bar{L}_s P_s$ and since $G' = \bar{W}_s$ exists and is finite we have $\bar{L}_s P_s = \lambda\bar{W}_s$. □

We are now interested in crossover results relating $\bar{W}_s$ and $\bar{L}_s$ to L_s and W_s. But first we prove a relation in distribution between $\mathbf{W}_s$ and $\bar{\mathbf{L}}_s$ which represent the random variables whose expectations coincide respectively with the limiting averages W_s and $\bar{L}_s$. Assume the queuing system has the following properties

Property 1. *1. Customers arrive at the system and leave it one at a time without blocking, balking, and reneging.*

2. Customers leave the system in the order of arrival (FIFO queue discipline)

3. New arriving customers do not affect the waiting times of the customers already in the system.

4. Variations in the system state (described by $Z(t)$) after a customer's arrival do not affect its waiting time.

Theorem 7. *If a queuing system in steady-state satisfies the assumptions in Property 1 then $\bar{\mathbf{L}}_s \stackrel{d}{=} \lambda(\mathbf{W}_s)$.*

Proof. Suppose a customer who enters the system at an arbitrary time t and observes state s upon arrival, that is $Z(t-) \in s$. Let's number the customers already waiting at time t in the reverse order of arrival and denote the corresponding arrival and departure times respectively by τ_i and τ_i', for $i = 1, 2, \cdots$ as depicted in Figure 7.1. Let t' denote the departure time of the customer arriving at t. Now, $t' - t$ which is assumed independent of both $\lambda(\tau)$ and $\mathbf{Z}(\tau)$ for $[\tau, \infty)$, will be distributed as $\mathbf{W}_s$. Moreover, because of the FIFO discipline, the customer arriving at time t can leave the system only after all the currently waiting customers have departed; thus the following events are equivalent:

$$\{\bar{\mathbf{L}}_s \geq n\} = \{\mathbf{W}_s \geq t' - \tau_n'\}$$

Therefore

$$\Pr\{\bar{\mathbf{L}}_s \geq n\} = \int_0^\infty \Pr\{t' - \tau_n' \leq \omega\}\, dF_{W_s}(\omega)$$

But the event $\{t' - \tau_n' \leq \omega\}$ is equivalent to the event that at least n customers depart during $[t' - \omega, t')$ which coincides with $\gamma(\omega)$ since the system and hence the departure process is stationary and therefore independent of t'. Thus

$$\Pr\{\bar{\mathbf{L}}_s \geq n\} = \int_0^\infty \Pr\{\gamma(\omega) \geq n\}\, dF_{W_s}(\omega) = \Pr\{\gamma(\mathbf{W}_s) \geq n\} = \Pr\{\lambda(\mathbf{W}_s) \geq n\}$$

where the last equality holds since the system is at steady-state. □

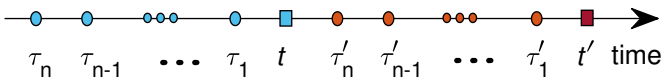


Figure 7.1 The ordered arrivals and departures

Taking the first moments from the above result yields the following relation.

Theorem 8. *If a queuing system in steady-state satisfies the assumptions in Property 1 then* $\bar{L}_s = \lambda W_s$.

Proof. Take first moments of both sides of the result in Theorem 7 □

Combining the previous relations we can derive a host of quick results.

Theorem 9. *For a queuing system in steady-state which satisfies the assumptions in Property 1,* $L_s = \pi_s^- \bar{L}_s$.

Proof. Combine Theorems 4 and 8. □

Theorem 10. *For a queuing system in steady-state for which the ASTA property and the assumptions in Property 1 hold,* $L_s = P_s \bar{L}_s$.

Proof. Combine Theorems 5 and 8. □

Theorem 11. *For a queuing system in steady-state for which the assumptions in Property 1 hold,* $\bar{W}_s = P_s W_s$.

Proof. Combine Theorems 6 and 8. □

Now let define $W_{ss'}$ as the average time spent in state s' by a customer who has observed state s just prior to arrival. We have the following result.

Theorem 12. *For a queuing system in steady-state for which the assumptions in Property 1 hold, for* s *and* $s' = S - s$ *we have* $W_{ss}(P_s - \pi_s^-) = \pi_{\bar{s}}^- W_{\bar{s}s} - P_s W_{s\bar{s}}$.

Proof. Conditioning on the pre-arrival state we have

$$\bar{W}_s = \pi_s^- W_{ss} + \pi_{\bar{s}}^- W_{\bar{s}s}$$

We also have

$$W_s = W_{ss} + W_{s\bar{s}}$$

Multiplying the second equation by P_s and combining the two yields the desired result. □

Theorem 13. *For a queuing system in steady-state for which ASTA and the assumptions in Property 1 hold, for s and $s' = S - s$ we have $P_s W_{s\bar{s}} = (1 - P_s) W_{\bar{s}s}$.*

Proof. Substitute $P_s = \pi_s^-$ and $P_{\bar{s}} = \pi_{\bar{s}}^-$ from ASTA into the result of Theorem 12. □

7.2 An Application

As mentioned earlier, we can use the results presented in this chapter and in particular, Theorem 8 to provide an alternative proof for the equilibrium state probabilities of the queuing system with partial backups. Below we repeat the result and the alternative proof.

Theorem 14. *The N-server queue with partial service, infinite queue capacity (M/M/[N]/∞), service rate μ, and arrival rates $\lambda_c \geq 0$, for $c = 1, \cdots, N$, will reach steady state if and only if*

$$\mu > \frac{1}{N} \sum_{c=1}^{N} \lambda_c \,,$$

with the equilibrium state occupancy probabilities given by

$$P_n = P_0 \frac{N!}{(N-n)!n!} \prod_{j=0}^{n-1} \sum_{c=1}^{N} \lambda_c \sum_{\substack{h=max\{0,\\ c+j-N))\}}}^{min\{c-1,j\}} \frac{c!(N-c)!}{(N+h-c-j)!(j-h)!(c-h)!h!} \times \prod_{j=1}^{n} \frac{j!(N-j)!}{N!j\mu - j! \sum_{c=1}^{j} \lambda_c \frac{N-c}{j-c}} \quad n = 1, \cdots, N,$$

where

$$P_0 = \left[1 + \sum_{n=1}^{N} \frac{N!}{(N-n)!n!} \prod_{j=0}^{n-1} \sum_{c=1}^{N} \lambda_c \sum_{\substack{h=max\{0,\\ c+j-N))\}}}^{min\{c-1,j\}} \frac{c!(N-c)!}{(N+h-c-j)!(j-h)!(c-h)!h!} \times \prod_{j=1}^{n} \frac{j!(N-j)!}{N!j\mu - j! \sum_{c=1}^{j} \lambda_c \frac{N-c}{j-c}} \right]^{-1};$$

Moreover, the expected queue times and queue lengths per customer class, are respectively given by

$$\omega_c = \sum_{n=c}^{N} \omega_c(n) P_n, \quad c = 1, \cdots, N,$$

and

$$L_c = \lambda_c \sum_{n=c}^{N} \frac{n!(N-c)!}{N!(n-c)!} \omega_c(n) P_n. \quad c = 1, \cdots, N.$$

Proof. Let $q_c(n)$ be the probability of a class c arrival entering the queue when there are exactly n busy servers; we then have

$$q_c(n) = \frac{\binom{n}{c}}{\binom{N}{c}} = \frac{n!(N-c)!}{N!(n-c)!}. \tag{7.1}$$

Also, let the $\omega_c(n)$ be the expected waiting delay incurred and $L_c(n)$ the expected queue length observed by class c customers who actually enter the queue when there are exactly n busy servers. Assuming the system is operating at steady state, we obtain the following relation from Theorem 8 for class c customers

$$L_c(n) = \lambda_c q_c(n) \omega_c(n).$$

Now consider a customer from class c who arrives at the system and finds exactly n servers busy, including all of its own covering servers. This customer then joins the queue of already waiting customers which is of the expected length of $L(n) = L_1(n) + L_2(n) + \cdots + L_n(n)$. We also recognize that future arrivals do not affect the waiting time of the new customer since no priorities in the queue are assumed. Therefore, we can remove all these future arrivals and the resulting upward state transitions. This allows us to compute the expected waiting time of the new customer by considering successive downward transitions from state n to c in this suppressed system and conditioning on S_k, the system state when it finally enters service. We readily observe that, on average, $L(k) - L(k-1)$ number of waiting customers should be processed by k servers before a downward transition from state S_k to S_{k-1} is possible. Moreover, the probability that a class c customer arriving at state S_n will enter service at state S_k is simply $\binom{k}{c} / \binom{n}{c}$. Finally, the time from the moment a class c customer enters the queue until any of its c covering servers finishes its current job and becomes ready to receive a new customer is equal to $1/c\mu$. Therefore, with all these observations we can express the expected waiting time of our new customer as

$$\omega_c(n) = \frac{1}{c\mu} + \sum_{k=c}^{n} \frac{1}{k\mu} \frac{\binom{k}{c}}{\binom{n}{c}} \left\{ L(k) - L(k-1) \right\}, \quad c = 1, \cdots, n, \ n = 1, \cdots, N,$$

or

$$\omega_c(n) = \frac{1}{c\mu} + \sum_{k=c}^{n} \frac{1}{k\mu} \frac{\binom{k}{c}}{\binom{n}{c}} \left\{ \sum_{c'=1}^{k} L_{c'}(k) - \sum_{c'=1}^{k-1} L_{c'}(k-1) \right\}, \quad c = 1, \cdots, n, \ n = 1, \cdots, N,$$

and finally

$$\omega_c(n) = \frac{1}{c\mu} + \sum_{k=c}^{n} \frac{1}{k\mu} \frac{\binom{k}{c}}{\binom{n}{c}} \sum_{c'=1}^{k} \lambda_{c'} q_{c'}(k) \omega_{c'}(k) - \sum_{k=c}^{n} \frac{1}{k\mu} \frac{\binom{k}{c}}{\binom{n}{c}} \sum_{c'=1}^{k-1} \lambda_{c'} q_{c'}(k-1) \omega_{c'}(k-1),$$
$$c = 1, \cdots, n, \ n = 1, \cdots, N.$$

Now if we re-arrange the above expression as

$$\omega_c(n) = \frac{1}{c\mu} + \sum_{k=c}^{n-1} \frac{1}{k\mu} \frac{\binom{k}{c}}{\binom{n}{c}} \sum_{c'=1}^{k} \lambda_{c'} q_{c'}(k) \omega_{c'}(k) -$$
$$\sum_{k=c}^{n} \frac{1}{k\mu} \frac{\binom{k}{c}}{\binom{n}{c}} \sum_{c'=1}^{k-1} \lambda_{c'} q_{c'}(k-1) \omega_{c'}(k-1) + \frac{1}{n\mu} \sum_{c'=1}^{n} \lambda_{c'} q_{c'}(n) \omega_{c'}(n), \quad c = 1, \cdots, n, \ n = 1, \cdots, N,$$

and combine it with (7.1), we obtain the following relation

$$\omega_c(n) = \frac{\frac{n}{c} + \frac{n!}{N!} \sum_{c'=c+1}^{n} \lambda_{c'} \frac{(N-c')!}{(n-c')!} \omega_{c'}(n)}{n\mu - \frac{n!}{N!} \sum_{c'=1}^{c} \lambda_{c'} \frac{(N-c')!}{(n-c')!}}. \tag{7.2}$$

For any given n, equation (7.2) can be solved recursively to obtain $\omega_n(n)$, $\omega_{n-1}(n)$,..., and $\omega_1(n)$ which in turn enables us to compute the expected total number of waiting customers for each system state, $L(n)$ as

$$L(n) = \sum_{c=1}^{n} L_c(n) = \sum_{c=1}^{n} \lambda_c q_c(n) \omega_c(n) = \sum_{c=1}^{n} \frac{n!(N-c)!}{N!(n-c)!} \lambda_c \omega_c(n), \qquad n = 0, \cdots, N. \tag{7.3}$$

Now, if we construct a general birth-death model as shown in Figure A.1 and determine its state-dependent transition rates, then we can easily obtain the desired state occupancy probabilities. The effective arrival rate at state S_n designated by $\lambda(n)$ can be simply computed as the sum of the incoming customers who arrive at the system at state S_n and immediately

gain access to a server and hence push the system to state S_{n+1}; or

$$\lambda(n) = \sum_{c=1}^{N} \lambda_c \left(1 - q_c(n)\right) = \sum_{c=1}^{n} \lambda_c (1 - \frac{n!(N-c)!}{N!(n-c)!}) + \sum_{c=n+1}^{N} \lambda_c, \quad n = 0, 1, \cdots, N-1.$$

To find the state-dependent service rates, $\mu(n)$, let us first denote by $T_{n\to n-1}$ the expected time period from the moment the system enters the state S_n (either from S_{n-1} or S_{n+1}) until it transitions to stat S_{n-1}. We then will obviously have $\mu(n) = 1/T_{n\to n-1}$. Moreover, we can break down $T_{n\to n-1}$ as

$$T_{n\to n-1} = \frac{1}{n\mu} + \frac{L(n) - L(n-1)}{n\mu}.$$

The first term on the right hand side is the expected time from the moment of transitioning into state S_n until any of the busy servers finishes its current job; while the second term represents the expected time spent by n servers on processing the waiting customers that will need to be processed until the system can switch back to state S_{n-1}; these include all class n waiting customers and those from other classes $c < n$ that will receive service while the queue of class n customers vanishes. Therefore, we can obtain $\mu(n)$ as

$$\mu(n) = \frac{n\mu}{1 + L(n) - L(n-1)},$$

which can be expanded using (7.3):

$$\mu(n) = \frac{n\mu}{1 + \sum_{c=1}^{n} \frac{n!(N-c)!}{N!(n-c)!}\lambda_c\omega_c(n) + \sum_{c=1}^{n-1} \frac{(n-1)!(N-c)!}{N!(n-1-c)!}\lambda_c\omega_c(n-1)}.$$

Having determined $\lambda(n)$ and $\mu(n)$, the state probabilities can simply be obtained as $P_n = \prod_{k=1}^{n}(\lambda(k)/\mu(k))P_0$, or

$$P_n = P_0 \prod_{k=1}^{k} \frac{\sum_{c=1}^{n} \lambda_c(1 - \frac{k!(N-c)!}{N!(k-c)!}) + \sum_{c=k+1}^{N} \lambda_c}{k\mu}$$
$$\times \left[1 + \sum_{c=1}^{k} \frac{k!(N-c)!}{N!(k-c)!}\lambda_c\omega_c(n) + \sum_{c=1}^{k-1} \frac{(k-1)!(N-c)!}{N!(k-1-c)!}\lambda_c\omega_c(k-1)\right]^{-1}, \quad n = 1, \cdots, N,$$

with P_0 given by

$$P_0 = \frac{1}{1 + \sum_{n=1}^{N} \prod_{i=1}^{n} \frac{\lambda(i)}{\mu(i)}},$$

or

$$P_0 = \left[1 + \sum_{n=1}^{N} \prod_{k=1}^{k} \left(\frac{\sum_{c=1}^{n} \lambda_c (1 - \frac{k!(N-c)!}{N!(k-c)!}) + \sum_{c=k+1}^{N} \lambda_c}{k\mu} \right) \right.$$
$$\left. \cdot \left(1 + \sum_{c=1}^{k} \frac{k!(N-c)!}{N!(k-c)!} \lambda_c \omega_c(n) + \sum_{c=1}^{k-1} \frac{(k-1)!(N-c)!}{N!(k-1-c)!} \lambda_c \omega_c(k-1)\right)^{-1}\right]^{-1}.$$

For the system to be able to reach state state, it is necessary and sufficient to have $\omega_c(n) < \infty$ for all $c = 1, \cdots, N$ and $n = 1, \cdots, N$, which by requiring the denominator of (7.2) to be positive translates to

$$\mu > \frac{(n-1)!}{N!} \sum_{c'=1}^{c} \lambda_{c'} \frac{(N-c')!}{(n-c')!}, \quad c = 1, \cdots, N;\ n = 1, \cdots, N. \tag{7.4}$$

It is easy to verify by inspection that the right hand side of the set of constraints given by (7.4) increases with both c and n; therefore, it suffices to satisfy the most stringent one corresponding to $c = n = N$; that is

$$\mu > \frac{1}{N} \sum_{c=1}^{N} \lambda_c \,.$$

Finally, having computed P_n and $\omega_c(n)$, the expected waiting times and queue lengths for each class, ω_c and L_c can be obtained by conditioning on the system state as

$$\omega_c = \sum_{n=1}^{N} P_n q_c(n) \omega_c(n) = \sum_{n=1}^{N} P_n \frac{n!(N-c)!}{N!(n-c)!} \omega_c(n), \quad c = 1, \cdots, N$$

and

$$L_c = \sum_{n=1}^{N} P_n L_c(n) = \sum_{n=1}^{N} P_n \frac{n!(N-c)!}{N!(n-c)!} \lambda_c \omega_c(n), \quad c = 1, \cdots, N.$$

□

CHAPTER 8 GENERAL DISCUSSIONS

As we stated in the introduction, mathematical models of emergency service systems that allow for partial backups will provide a better approximate to the steady-state behavior of the system since in many cases a limitation is imposed on the response units that can be dispatched to a request for service. For example by a limited range of a drone used to deliver AEDs to patients of OHCAs, by the explicit assignment of patrol zones to police patrol cars, and by the fact that dispatchers may decide not to send vehicles over long distances because of the resulting long response time and poor outcomes. Despite these simple observations on the shortcomings of the usual full backup assumption, the literature is almost non-existent on the descriptive models and algorithms that explicitly allow for partial backups and thus can be used to approximate systems in which the assumption of full backups does not hold. The main theme of the work presented in this thesis, however, is the the development of new mathematical models or extension of the existing ones to include partial backups as an explicit input parameter or decision variable than can then be optimized alongside the other system variable to obtain the best performance possible. The descriptive models we introduced in Chapter 4 and Chapter 6 thus represent our attempt to bridge this perceived gap in the literature.

From the results presented in Chapter 4, it should be clear that relaxing the assumption of full backups in the hypercube queuing model and allowing for priorities, will indeed enable us to approximate and consequently optimize a wider range of emergency operations with reasonable accuracy. In particular, one can apply the extended hypercube model to find the optimal values of location and allocation decision variables at the same time while the allocation variables may depend on priorities as well. As stated earlier, since periodic, dynamic, and compliance-based deployments all can be obtained via solution of a series of static deployments, the improved accuracy of the extended hypercube model in approximating more realistic statistic scenarios, will directly translate to better and more realistic solutions to the static, periodic, compliance-based, and dynamic deployment problems as well.

The queuing theoretical framework we proposed in Chapter 6 provided us with an abstract descriptive model of emergency service systems with dynamic relocation. This model can be used to assess the impact of utilizing partial backup dispatch policies on the system performance or outcome given as a function of response time. While the extended hypercube model in Chapter 4 can be incorporated into an optimization model for concurrent optimization of location and allocation decisions, the algorithm in Chapter 6 can be used to find the best

partial backup dispatch policy to obtain the best possible outcome for a system which is only define by its most basic configurations such as the fleet size, total service area, total arrival rate, a queuing or loss discipline, and the outcome function. The analysis using this simplified model can reveal potential opportunities for improving the performance of a system with dynamic relocation via utilization of a partial backup dispatch policy, especially in under-resourced deployment situations. In a broader perspective, we use the model and the application examples to raise an awareness of the role of managing dispatch rules in achieving optimal performance and its close relation with the system configuration, underlying operating conditions, and the response time profile of the expected outcome.

Further work is however required to apply both of these models in optimization of realistic system deployments and compare the results with solutions obtained from a less refined model with a full backup assumption.

CHAPTER 9 CONCLUSION AND RECOMMENDATIONS

In conclusion, we overview the topics visited in this manuscript along with major limitations of the work presented and promising directions for future work.

9.1 Summary of the Work

The focus in this work has been on descriptive models for Emergency Service Systems where we tried to relax some of the less realistic assumptions commonly made in the literature, such as the assumption of full backups where every server can respond to every call for service. We developed a new extension of the Larson Hypercube Queuing model where this assumption of full backups was replaced with a more realistic partial backups where a given subset of servers can respond to a call coming from a given region.

Building upon the queuing models we developed in the first contribution, we extended our analysis to the systems with dynamic relocation and proposed a new descriptive framework for evaluating the effects of partial backup dispatch policies on the performance of the system in maximizing a given performance objective.

The analyses presented in Chapters 4 and 6, for emergency service systems, depended in part on the results we obtained concerning the distribution of distances to neighbors in Chapter 5 and extensions to Little's law in 7.

9.2 Limitations

Limitations of the study are for the most part reflected in the future research directions we outline in the next Subsection. Besides these technical areas of potential improvement, however, the lack of historical EMS data to use in the validation and evaluation of the performance of the models presented, can be cited as an unfortunate limitation.

9.3 Future Research

The ideas and results presented in this work can be expanded upon and explored in more depth in numerous ways. First of all, the descriptive models presented, can be applied in development of many different optimization (prescriptive) models for different emergency response scenarios. In fact, every probabilistic optimization model proposed in the literature

that uses the classical hypercube model approximation, can easily be improved by replacing it with the extended algorithm presented here.

As mentioned earlier, the framework presented in Chapter 6 for evaluating optimal partial dispatch policies can be studied and modified in a few ways as well. First of all, perhaps using realistic simulations of many application scenarios, we need to verify the optimality or near-optimality of the partial dispatch policies suggested by this model. Second, it can be rather easily extended to include more than one priorities with each priority level having its own partial dispatch policy. Third, the potential use of the set of suggested optimal dispatch policies per system states as a state-dependent adaptive dispatch policy seems promising to explore in detail.

We also stated in Chapter 5 that, despite the effort dedicated, we were not successful in derivation of simple close-form approximations to the distance to the k-th neighbor with boundary effects. Needless to say, this still remains an interesting topic to the author and a challenging open question in the literature with wide applications across many different disciplines.

Finally, potential applications can be sought after for the state-dependent extensions of the Little's Law presented in Chapter 7. In particular, we believe that these results may pave the road to approximate solutions of the queuing systems with skill-based routing.

REFERENCES

Lina Aboueljinane, Evren Sahin, and Zied Jemai. A review on simulation models applied to emergency medical service operations. *Computers & Industrial Engineering*, 66(4):734–750, 2013.

Ivo Adan and Gideon Weiss. A skill based parallel service system under FCFS-ALIS—steady state, overloads, and abandonments. *Stochastic Systems*, 4(1):250–299, 2014.

Ivo Adan, Cor Hurkens, and Gideon Weiss. A reversible Erlang loss system with multitype customers and multitype servers. *Probability in the Engineering and Informational Sciences*, 24(04):535–548, 2010.

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

Roberto Aringhieri, Maria Elena Bruni, Sara Khodaparasti, and J Theresia van Essen. Emergency medical services and beyond: Addressing new challenges through a wide literature review. *Computers & Operations Research*, 78:349–368, 2017.

JB Atkinson, IN Kovalenko, N Yu Kuznetsov, and KV Mikhalevich. Heuristic methods for the analysis of a queuing system describing emergency medical service deployed along a highway. *Cybernetics and Systems Analysis*, 42(3):379–391, 2006.

JB Atkinson, Igor N Kovalenko, N Kuznetsov, and KV Mykhalevych. A hypercube queueing loss model with customer-dependent service rates. *European Journal of Operational Research*, 191(1):223–239, 2008.

Valérie Bélanger, Angel Ruiz, and Patrick Soriano. Déploiement et redéploiement des véhicules ambulanciers dans la gestion d'un service préhospitalier d'urgence. *INFOR: Information Systems and Operational Research*, 50(1):1–30, 2012.

Valérie Bélanger, A Ruiz, and Patrick Soriano. Recent optimization models and trends in location, relocation, and dispatching of emergency medical vehicles. *European Journal of Operational Research*, 272(1):1–23, 2019.

Dimitris Bertsimas and Daisuke Nakazato. The distributional little's law and its applications. *Operations Research*, 43(2):298–310, 1995.

Markus Bibinger. Notes on the sum and maximum of independent exponentially distributed random variables with different scale parameters. *arXiv preprint arXiv:1307.3945*, 2013.

John R Birge and Stephen M Pollock. Using parallel iteration for approximate analysis of a multiple server queueing system. *Operations Research*, 37(5):769–779, 1989.

Justin J Boutilier, Steven C Brooks, Alyf Janmohamed, Adam Byers, Jason E Buick, Cathy Zhan, Angela P Schoellig, Sheldon Cheskes, Laurie J Morrison, and Timothy CY Chan. Optimizing a drone network to deliver automated external defibrillators. *Circulation*, 135 (25):2454–2465, 2017.

Steven C Brooks, Robert H Schmicker, Thomas D Rea, Tom P Aufderheide, Daniel P Davis, Laurie J Morrison, Ritu Sahni, Gena K Sears, Denise E Griffiths, George Sopko, et al. Out-of-hospital cardiac arrest frequency and survival: evidence for temporal variability. *Resuscitation*, 81(2):175–181, 2010.

Luce Brotcorne, Gilbert Laporte, and Frederic Semet. Ambulance location and relocation models. *European journal of operational research*, 147(3):451–463, 2003.

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

Susan Budge, Armann Ingolfsson, and Dawit Zerom. Empirical analysis of ambulance travel times: the case of Calgary emergency medical services. *Management Science*, 56(4): 716–723, 2010.

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

Paul S Chan, Bryan McNally, Fengming Tang, and Arthur Kellermann. Recent trends in survival from out-of-hospital cardiac arrest in the united states. *Circulation*, 130(21): 1876–1882, 2014.

A Claesson, D Fredman, L Svensson, M Ringh, J Hollenberg, P Nordberg, M Rosenqvist, T Djarv, S Österberg, J Lennartsson, et al. Unmanned aerial vehicles (drones) in out-of-hospital-cardiac-arrest. *Scandinavian journal of trauma, resuscitation and emergency medicine*, 24(1):124, 2016.

A Claesson, L Svensson, P Nordberg, M Ringh, M Rosenqvist, T Djarv, J Samuelsson, O Hernborg, P Dahlbom, A Jansson, et al. Drones may be used to save lives in out of hospital cardiac arrest due to drowning. *Resuscitation*, 114:152–156, 2017.

Michael F Dacey. Two-dimensional random point patterns: A review and an interpretation. *Papers in Regional Science*, 13(1):41–55, 1964.

Valerie J De Maio, Ian G Stiell, George A Wells, Daniel W Spaite, Ontario Prehospital Advanced Life Support Study Group, et al. Optimal defibrillation response intervals for maximum out-of-hospital cardiac arrest survival rates. *Annals of emergency medicine*, 42 (2):242–250, 2003.

Regiane Máximo de Souza, Reinaldo Morabito, Fernando Y Chiyoshi, and Ana Paula Iannoni. Incorporating priorities for waiting customers in the hypercube queuing model with application to an emergency medical service system in Brazil. *European Journal of Operational Research*, 242(1):274–285, 2015.

Muhammad El-Taha and Shaler Stidham Jr. *Sample-path analysis of queueing systems*, volume 11. Springer Science & Business Media, 2012.

Roberto D Galvao and Reinaldo Morabito. Emergency service systems: The use of the hypercube queueing model in the solution of probabilistic location problems. *International Transactions in Operational Research*, 15(5):525–549, 2008.

Jeffrey Goldberg and Ferenc Szidarovszky. Methods for solving nonlinear equations used in evaluating emergency vehicle busy probabilities. *Operations research*, 39(6):903–916, 1991.

Rasoul Haji and Gordon F Newell. A relation between stationary queue and waiting time distributions. *Journal of Applied Probability*, 8(03):617–620, 1971.

Carolina Malta Hansen, Kristian Kragholm, David A Pearson, Clark Tyson, Lisa Monk, Brent Myers, Darrell Nelson, Matthew E Dupre, Emil L Fosbøl, James G Jollis, et al. Association of bystander and first-responder intervention with survival after out-of-hospital cardiac arrest in north carolina, 2010-2013. *Jama*, 314(3):255–264, 2015.

Daniel P Heyman and Shaler Stidham Jr. The relation between customer and time averages in queues. *Operations Research*, 28(4):983–994, 1980.

I, 2019.

Ana Paula Iannoni and Reinaldo Morabito. 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, 2007.

Ana Paula Iannoni, Fernando Chiyoshi, and Reinaldo Morabito. A spatially distributed queuing model considering dispatching policies with server reservation. *Transportation Research Part E: Logistics and Transportation Review*, 75:49–66, 2015.

Armann Ingolfsson. Ems planning and management. In *Operations research and health care policy*, pages 105–128. Springer, 2013.

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

Helena Jasiulewicz and Wojciech Kordecki. Convolutions of erlang and of pascal distributions with applications to reliability. *Demonstratio Mathematica*, 36(1):231–238, 2003.

Norman L Johnson, Samuel Kotz, and Narayanaswamy Balakrishnan. Univariate continuous distributions, 1994.

Akbar Karimi, Michel Gendreau, and Vedat Verter. Performance approximation of emergency service systems with priorities and partial backups. *Transportation Science*, 2018.

J Keilson and LD Servi. A distributional form of Little's law. *Operations Research Letters*, 7(5):223–227, 1988.

Peter Kolesar, Warren Walker, and Jack Hausner. Determining the relation between fire engine travel times and travel distances in new york city. *Operations Research*, 23(4):614–627, 1975.

Wojciech Kordecki. Reliability bounds for multistage structures with independent components. *Statistics & probability letters*, 34(1):43–51, 1997.

R.C. Larson and A.R. Odoni. *Urban operations research*. Prentice Hall PTR, 1981. ISBN 9780139394478. URL `https://books.google.ca/books?id=5moeAQAAIAAJ`.

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

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

Richard C Larson. Hypercube queueing model. In *Encyclopedia of Operations Research and Management Science*, pages 733–739. Springer, 2013.

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

John DC Little. OR FORUM-Little's Law as Viewed on Its 50th Anniversary. *Operations research*, 59(3):536–549, 2011.

MV Lomonosov. Bernoulli scheme with closure. *Problemy Peredachi Informatsii*, 10(1): 91–101, 1974.

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

Benjamin Melamed and Ward Whitt. On arrivals that see time averages. *Operations Research*, 38(1):156–172, 1990.

Benjamin Melamed and D Yao. The ASTA property. *Advances in Queueing: Theory, Methods and Open Problems*, pages 195–224, 1995.

Dmitri Moltchanov. Distance distributions in random networks. *Ad Hoc Networks*, 10(6): 1146–1166, 2012.

Masaaki Morisita. Estimation of population density by spacing method. *Memoirs of Faculty of Science, Kyushu University, Series E (Biology)*, 1:187–197, 1954.

Graham Nichol, Elizabeth Thomas, Clifton W Callaway, Jerris Hedges, Judy L Powell, Tom P Aufderheide, Tom Rea, Robert Lowe, Todd Brown, John Dreyer, et al. Regional variation in out-of-hospital cardiac arrest incidence and outcome. *Jama*, 300(12):1423–1431, 2008.

Lásara Fabrícia Rodrigues, Reinaldo Morabito, Fernando Y Chiyoshi, Ana Paula Iannoni, and Cem Saydam. Towards hypercube queuing models for dispatch policies with priority in queue and partial backup. *Computers & Operations Research*, 84:92–105, 2017.

J Sanfridsson, J Sparrevik, J Hollenberg, P Nordberg, T Djärv, M Ringh, L Svensson, S Forsberg, A Nord, Magnus Andersson-Hagiwara, et al. Drone delivery of an automated external defibrillator–a mixed method simulation study of bystander experience. *Scandinavian journal of trauma, resuscitation and emergency medicine*, 27(1):40, 2019.

Comilla Sasson, Mary AM Rogers, Jason Dahl, and Arthur L Kellermann. Predictors of survival from out-of-hospital cardiac arrest: a systematic review and meta-analysis. *Circulation: Cardiovascular Quality and Outcomes*, 3(1):63–81, 2010.

Ananda Sen and N Balakrishnan. Convolution of geometrics and a reliability problem. *Statistics & probability letters*, 43(4):421–426, 1999.

John Gordon Skellam. Random dispersal in theoretical populations. *Biometrika*, 38(1/2): 196–218, 1951.

Statistics Canada. Dissemination areas cartographic boundary file for the 2011 census, December 25 2016a. `http://www12.statcan.gc.ca/census-recensement/2011/geo/bound-limit/bound-limit-2011-eng.cfm` accessed on July 4, 2017.

Statistics Canada. Census profile of dissemination areas for the 2011 census, Dec 22 2016b. `http://www12.statcan.gc.ca/census-recensement/2011/dp-pd/prof/index.cfm?Lang=E` accessed on July 4, 2017.

HR Thompson. Distribution of distance to nth neighbour in a population of randomly distributed individuals. *Ecology*, 37(2):391–394, 1956.

Fei Tong, Jianping Pan, and Ruonan Zhang. Distance distributions in finite ad hoc networks: Approaches, applications, and directions. In *Ad Hoc Networks*, pages 167–179. Springer, 2017.

Nicolaas Godfried Van Kampen. *Stochastic processes in physics and chemistry*, volume 1. Elsevier, 1992.

Jeremy Visschers, Ivo Adan, and Gideon Weiss. A product form solution to a system with multi-type jobs and multi-type servers. *Queueing Systems*, 70(3):269–298, 2012.

Mads Wissenberg, Freddy K Lippert, Fredrik Folke, Peter Weeke, Carolina Malta Hansen, Erika Frischknecht Christensen, Henning Jans, Poul Anders Hansen, Torsten Lang-Jensen, Jonas Bjerring Olesen, et al. Association of national initiatives to improve cardiac arrest management with rates of bystander intervention and patient survival after out-of-hospital cardiac arrest. *Jama*, 310(13):1377–1384, 2013.

APPENDIX A PROOF OF THEOREMS IN CHAPTER 4

A.1 Theorem 1

Proof. We first briefly summarize the results obtained by Visschers et al. (2012) for a class of queues with skill based service and exploit them to derive our expressions of interest.

In a skill based queue with a set of servers $\mathcal{M} = \{m_1, \cdots, m_N\}$, and a set of customer classes $\mathcal{C} = \{\kappa_1, \kappa_2, \cdots\}$, each server $m_k \in \mathcal{M}$ has the skills to only serve a subset $\mathcal{C}(m_k) \subset \mathcal{C}$ of customer types. Servers are memory-less with service rates μ_m, $m \in \mathcal{M}$ and customers of class $\kappa \in \mathcal{C}$ arrive in a Poisson stream with rate λ_κ. The set of customer classes only compatible with servers $\{M_1, \cdots, M_n\} \in \mathcal{M}^n$ is denoted by $\mathcal{U}(\{M_1, \cdots, M_n\})$ where $\mathcal{M}^n$ is the set of all subsets of size n of $\mathcal{M}$. We readily observe that $\mathcal{U}(\{M_1, \cdots, M_n\}) = \mathcal{C} \setminus \bigcup_{m \in \mathcal{M} \setminus \{M_1, \cdots, M_n\}} \mathcal{C}(m)$. For a subset $\mathcal{C}' \subset \mathcal{C}$ of customer types, we denote $\lambda_{\mathcal{C}'} = \sum_{\kappa \in \mathcal{C}'} \lambda_{c'}$; similarly $\mu_{\mathcal{M}' \subset \mathcal{M}} = \sum_{m \in \mathcal{M}'} \mu_m$.

A state of the system, given by the sequence $(M_1, l_1, M_2, l_2, \cdots, M_n, l_n)$, represents a snapshot of the waiting customers and the relative positions of the busy servers in the queue: there are $l_1 + \cdots + l_i$ ($l_i = 0, 1, \cdots$) customers waiting between the sequence of busy servers $(M_1, \cdots, M_n)$, plus n customers currently in service. Arriving customers start service immediately if they find a compatible server among the set of idle servers $\{M_{n+1}, \cdots, M_N\}$, and push the system to state $(M_1, l_1, M_2, l_2, \cdots, M_n, l_i, M_{n+1}, 0)$; otherwise, they join the right end of the queue, pushing the system into state $(M_1, l_1, M_2, l_2, \cdots, M_n, l_n + 1)$. Customers who find several available and compatible servers upon arrival, choose the receiving server randomly according to given assignment probability distributions. Upon finishing their current job, servers will scan the queue from the left and start servicing the first customer they can handle, if one exists. This results in a new system state with the hosting server relocated to the position of the received customer. If no compatible customers are found, the newly released servers will join the set of other idle servers, transitioning the system to state $(M_1, l_1, \cdots, M_{k-1}, l_{k-1} + l_k, M_{k+1}, \cdots, M_n, l_n)$ where M_k is the newly idle server ($k \leq n$). We therefore have that the l_k customers waiting between servers M_k and M_{k+1} must belong to $\mathcal{U}(\{M_1, \cdots, M_k\})$. For a detailed description of this system and its transitions see the original paper.

Within the setup summarized above and with $\Pr(\emptyset)$ the probability of an empty system,

Visschers et al. (2012) consider the product form distribution

$$\Pr(M_1, l_1, M_2, l_2, \cdots, M_n, l_n) = \Pr(\emptyset) \frac{\Pi_\lambda(\{M_1, \cdots, M_n\})}{\Pi_\mu(M_1, \cdots, M_n)} \prod_{j=1}^{n} \alpha_j^{l_j}, \tag{A.1}$$

where

$$\alpha_j = \frac{\lambda_{\mathcal{U}(\{M_1, \cdots, M_j\})}}{\mu_{\{M_1, \cdots, M_j\}}}, \quad \text{for } j = 1, 2, \cdots, n,$$

and

$$\Pi_\lambda(\{M_1, \cdots, M_n\}) = \prod_{j=1}^{n} \lambda_{M_j}(\{M_1, \cdots, M_{j-1}\}), \tag{A.2}$$

for every subset $\{M_1, \cdots, M_n\} \in \mathcal{M}$ of servers and

$$\Pi_\mu(M_1, \cdots, M_n) = \prod_{j=1}^{n} \mu_{\{M_1, \cdots, M_j\}},$$

for every sequence $(M_1, \cdots, M_n) \in \mathcal{P}^n$ with $\mathcal{P}^n$ the set of all permutations of all subsets of size n of $\mathcal{M}$. While $\Pi_\mu(M_1, \cdots, M_n)$ may depend on the order of servers in the sequence $(M_1, \cdots, M_n)$, for the product form solution to exist, Visschers et al. (2012) show that $\Pi_\lambda(\{M_1, \cdots, M_n\}$ must be independent of the order of servers. It is then shown that there exist unique values for the *activation rates* $\lambda_{M_j}(\{M_1, \cdots, M_{j-1}\})$ which satisfy this *assignment condition* and that these unique values can be calculated recursively. The assignment probability distributions leading to these activation rates are often not unique and can be obtained by solving a maximal flow problem for each set $\{M_1, \cdots, M_{j-1}\}$.

Finally, the product form (A.1) implies the probability of having the sequence of busy servers $(M_1, \cdots, M_n)$ as

$$\Pr(M_1, \cdots, M_n) = \Pr(\emptyset) \frac{\Pi_\lambda(\{M_1, \cdots, M_n\})}{\Pi_\mu(M_1, \cdots, M_n)} \prod_{j=1}^{n} \frac{1}{1 - \alpha_j}. \tag{A.3}$$

The assignment condition is trivially satisfied for the queue with partial service where servers are indistinguishable and hence the assignment of available compatible servers to incoming customers is purely random. This allows us to derive the distribution of the number of busy servers of the queue with partial service (not to be confused with class κ in the skill based queue) by evaluating (A.3). Distinguishing the expressions obtained for the queue with partial service with tildes we write

$$\tilde{\lambda}_{\mathcal{U}\{M_1,\cdots,M_j\}} = \sum_{c=1}^{j} \lambda_c \frac{\binom{j}{c}}{\binom{N}{c}}\,, \tag{A.4}$$

and

$$\tilde{\mu}_{\{M_1,\cdots,M_j\}} = j\mu\,, \tag{A.5}$$

which immediately follow from servers being indistinguishable. We then have

$$\tilde{\alpha}_j = \frac{\tilde{\lambda}_{\mathcal{U}(\{M_1,\cdots,M_j\})}}{\tilde{\mu}_{\{M_1,\cdots,M_j\}}} = (j\mu)^{-1} \sum_{c=1}^{j} \lambda_c \frac{\binom{j}{c}}{\binom{N}{c}}\,,$$

and

$$\tilde{\Pi}_\mu(M_1,\cdots,M_n) = \prod_{j=1}^{n} j\mu = n!\mu^n\,.$$

To derive an expression for $\tilde{\Pi}_\lambda(\{M_1,\cdots,M_n\})$, we consider the situation with $j-1$ busy servers and designate one of the idle servers to act as M_j. We then recognize the probability of a class c customer being simultaneously compatible with the designated server as well as exactly h of the busy servers as

$$\phi_c(h,j) = \frac{\binom{j-1}{h}\binom{N-j}{c-h-1}}{\binom{N}{c}}, \qquad \begin{array}{l} j \in \mathcal{M} \\ h = \max\{0, c+j-N-1\}\,,\cdots,\min\{c-1, j-1\} \end{array}\,,$$

and the probability of such a customer being assigned to the designated server as $1/(c-h)$. Conditioning on h and interpreting $\tilde{\lambda}_{M_j}(\{M_1,\cdots,M_{j-1}\})$ as the total activation rate of a designated idle server given the set of busy servers $\{M_1,\cdots,M_{j-1}\}$ we get

$$\tilde{\lambda}_{M_j}(\{M_1,\cdots,M_{j-1}\}) = \sum_{c=1}^{N} \lambda_c \sum_{\substack{h=\max\{0,\\ c-(N-(j-1))\}}}^{\substack{\min\{c-1,\\ j-1\}}} \frac{\phi_c(h,j)}{c-h}\,,$$

and then from (A.2)

$$\tilde{\Pi}_\lambda(\{M_1,\cdots,M_n\}) = \prod_{j=1}^{n} \sum_{c=1}^{N} \lambda_c \sum_{h=\max\{0,c-(N-(j-1))\}}^{\min\{c-1,j-1\}} \frac{\binom{j-1}{h}\binom{N-j}{c-h-1}}{\binom{N}{c}} \frac{1}{c-h}\,,$$

which of course satisfies the assignment condition. Substituting the above expressions into (A.3), summing over all members of $\mathcal{P}^n$ for $n = 0,\cdots,N$, and simplifying we obtain

$$P_n = P_0 \frac{N!}{(N-n)!n!} \prod_{j=0}^{n-1} \sum_{c=1}^{N} \lambda_c \sum_{\substack{h=\max\{0,\\ c+j-N))\}}}^{\min\{c-1,j\}} \frac{c!(N-c)!}{(N+h-c-j)!(j-h)!(c-h)!h!} \times \prod_{j=1}^{n} \frac{j!(N-j)!}{N!j\mu - j! \sum_{c=1}^{j} \lambda_c \frac{N-c}{j-c}}, \quad n = 0, \cdots, N, \quad \text{(A.6)}$$

with the normalizing factor $P_0 = (1 + \sum_{j=1}^{N} P_j)^{-1}$. □

Remark 2. *Adan and Weiss (2014) showed that this product form is also valid for the case where the random assignment policy is replaced with the Assign to the Longest Idle Server (ALIS). Consequently, one can derive* (A.6) *assuming an FCFS-ALIS queue discipline; however, we find the presented approach more direct and intuitive.*

A.2 Theorem 2

Proof. A class c customer arriving to the loss system with partial service in state S_n will be lost with probability $q_c(n) = \binom{n}{c} / \binom{N}{c}$ and will enter service immediately with probability $1 - q_c(n)$. The system then can be modeled as a birth-death process shown in Figure A.1 with $\mu(n) = n\mu$ for $n = 1, \cdots, N$, and

$$\lambda(n) = \sum_{c=1}^{N} \lambda_c q_c(n) = \sum_{c=1}^{N} \lambda_c \frac{n!(N-c)!}{N!(n-c)!}, \quad n = 0, \cdots, N-1.$$

It is then easy to verify that the steady state distribution will be

$$P_n = P_0 \prod_{k=1}^{n} \frac{\sum_{c=1}^{k} \lambda_c (1 - \frac{k!(N-c)!}{N!(k-c)!}) + \sum_{c=k+1}^{N} \lambda_c}{k\mu}, \quad n = 1, \cdots, N,$$

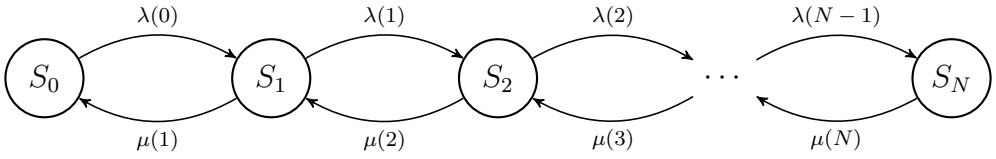
with the probability of the empty system $P_0 = (1 + \sum_{n=1}^{N} P_n)^{-1}$ as the normalizing factor. □

Remark 3. *This result also follows from the product form given in Adan et al. (2010) for the loss version of the skill based systems considered in Visschers et al. (2012); the derivation presented here, however, is simpler.*

A.3 Correction of State Probability Approximations

The accuracy of the distribution of the number of busy servers obtained from the M/M/[N] model with the demand vector $[\lambda_c]$ and service rate μ given by (4.5) and (4.6) as parameters can be further improved through the empirical scheme outlined her, which is obtained by

Figure A.1 The birth-death model to compute state probabilities



analyzing a large set of test cases and comparing the outputs of the M/M/[N] and simulation models.

We use Algorithm 1 to map any given partial service demand vector $[\lambda_c]$ to a reshaped vector $[\hat{\lambda}_c]$ by moving the demand concentration towards the lower admission classes while keeping the total arrival rate $\lambda = \sum_{c=1}^{N} \lambda_c = \sum_{c=1}^{N} \hat{\lambda}_c$ intact. The parameter $\zeta \in [0, 1]$ controls the intensity of the operation; with $\zeta = 0$, we will have $[\hat{\lambda}_c] = [\lambda_c]$ and with $\zeta = 1$, all demand will be moved into the admission class $c = 1$, that is $\hat{\lambda}_1 = \lambda$ and $\hat{\lambda}_c = 0$ for $c = 2, 3, \cdots, N$.

input : $\zeta \in [0, 1]$ and λ_c for $c = 1, 2, \cdots, N$
output: $\hat{\lambda}_c$ for $c = 1, 2, \cdots, N$
$\hat{\lambda}_c \leftarrow \lambda_c$, $c = 1, 2, \cdots, N$;
for $i \leftarrow N$ **to** 2 **do**
  $\Delta \leftarrow \zeta \hat{\lambda}_i$;
  for $j \leftarrow 1$ **to** $i - 1$ **do**
    $\hat{\lambda}_j \leftarrow \hat{\lambda}_j + \frac{\Delta}{i-1}$;
  end
  $\hat{\lambda}_i \leftarrow \hat{\lambda}_i - \Delta$;
end

Algorithm 1: Reshaping λ_c with a factor ζ

As the measure of variation among server workloads $\rho_1, \rho_2, ..., \rho_N$, we define

$$\eta = \frac{\sum_{j \in \mathcal{J}} |\rho_j - \rho|}{\rho N^{1.3}},$$

where $\rho = \frac{1}{N} \sum_{j \in \mathcal{J}} \rho_j$ is the mean server workload. This specific expression for η was observed to make the formulas proposed here effectively independent of the fleet size N. For a given η, suitable values of ζ can be best predicted using a piece-wise expression given by

$$\zeta = \begin{cases} b_0 \eta^3 & \text{if } \eta \leq \eta_s \\ \sqrt{\frac{\eta - a_0}{a_2}} & \text{if } \eta \geq \eta_s \end{cases},$$

where b_0, a_0, a_2, η_s, and ζ_s are parameters depending on the nominal loading $\varrho = \lambda/N\mu$ through as follows:

$$a_0 = \begin{cases} 0.0454\varrho^3 + 0.0510\varrho^2 - 0.3140\varrho + 0.2173 & \text{for systems with queues} \\ 0.1567\varrho^3 - 0.1043\varrho^2 - 0.2442\varrho + 0.2071 & \text{for loss systems} \end{cases}$$

$$\zeta_s = \begin{cases} 0.7262\varrho^2 - 1.761\varrho + 1.341 & \text{for systems with queues} \\ 1.006\varrho^2 - 2.191\varrho + 1.3592 & \text{for loss systems} \end{cases}$$

$$a_2 = \begin{cases} 0.18 & \text{for systems with queues} \\ -0.4419\varrho^3 + 0.3262\varrho^2 - 0.0889\varrho + 0.1878 & \text{for loss systems} \end{cases}$$

$$\eta_s = a_2\zeta_s^2 + a_0$$

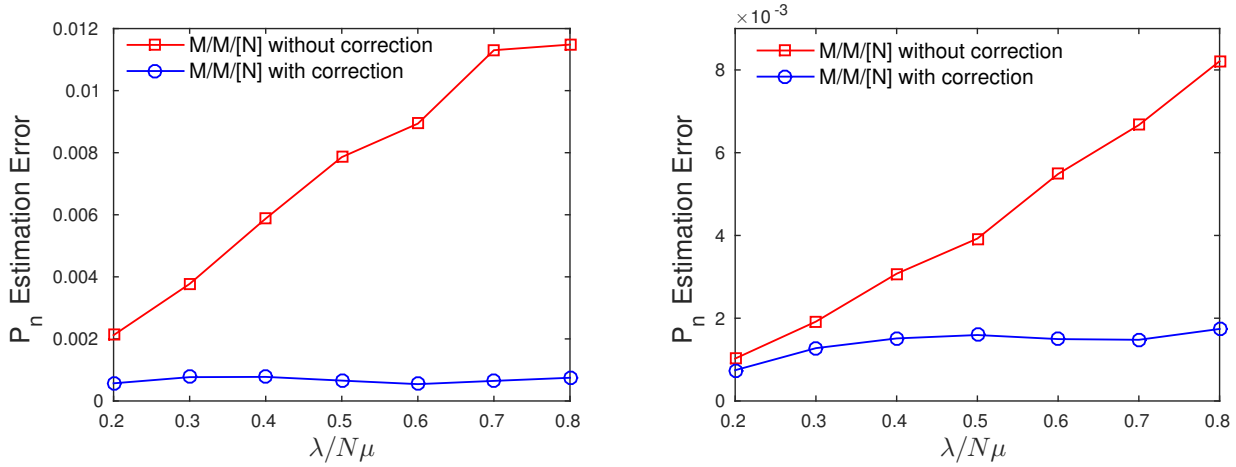
$$b_0 = \zeta_s\eta_s^{-3}\,.$$

For systems with queues, this reshaping procedure does not affect the average server utilization ρ as the total demand λ is kept unchanged; however, for loss systems, any alteration of $[\lambda_c]$ through the reshaping algorithm may also change ρ. Therefore, we also need to perform a normalization step so that the resulting average server workload matches the current iteration value ρ. This is achieved by replacing the nominal service rate μ with an adjusted version $\hat{\mu}$ through the following relation

$$\hat{\mu} = \begin{cases} \mu & \text{for systems with queues} \\ (\rho N)^{-1} \sum_{n=0}^{N} P_n \sum_{c=1}^{N} \lambda_c(1 - q_c(n)) & \text{for loss systems} \end{cases},$$

where $q_c(n)$ is given by (4.8). This normalization step is equivalent to multiplying the input vector $[\lambda_c]$ by a factor of $\mu/\hat{\mu}$.

Finally, the corrected state probabilities are obtained from an M/M/[N] model with the reshaped input vector $[\hat{\lambda}_c]$ and the normalized service rate $\hat{\mu}$ as parameters. Figure A.2 shows the average improvement of P_n estimations for queuing and loss systems at varying workloads.



(a) Loss System

(b) Queuing System

Figure A.2 Comparison of the state probability estimation errors with and without the correction scheme averaged over randomly generated cases with 20 servers ($N = 20$) and varying workloads. The estimation error is computed as $\sum_{n=1}^{N} |P_n^{mod} - P_n^{sim}|$.

APPENDIX B COMPLEMENTARY COMPUTATIONAL EXPERIMENTS FOR CHAPTER 4

In this document, we present our algorithm for further improving the state probability approximations together with the extra computational experiments conducted to assess the performance of the method.

B.1 Sensitivity to Service Time Distribution

So far we have assumed the service times to be exponentially distributed; however, this assumption may be violated in many real-world applications where the service times exhibit much less variability. Jarvis (1985) and Budge et al. (2009) showed *loss* service systems to be fairly insensitive to the service time distribution beyond its mean. We consider both queuing and loss systems and hence are interested in evaluating the impact of the service time distribution on the accuracy of the approximation in each case. We have run simulations with different service time distribution scenarios outlined in Table B.1 and compared the results with the outputs of the approximation model which assumes exponential service times. Scenario D1 represents the ideal case where the total service time is exponentially distributed with the mean values given as the sum of the expectations of its components, that is

$$\bar{t}_{ijp}^{ser} = \bar{t}_{ijp}^{disp} + \bar{t}_{ijp}^{chut} + \bar{t}_{ijp}^{trv} + \bar{\alpha}_{trans}(\bar{t}_{ijp}^{scen/trans} + \bar{t}_{ijp}^{trans} + \bar{t}_{ijp}^{turnaround}) + (1 - \bar{\alpha}_{trans})(\bar{t}_{ijp}^{scen/no.trans}) \,,$$

with the barred symbols indicating the mean values and $\bar{\alpha}_{trans}$ the probability of a patient requiring a transport from the scene of the incident to a care center. Note that the mean travel and transport times $\bar{t}_{ijp}^{trv}$ and $\bar{t}_{ijp}^{trans}$ are obtained following from (4.27). Scenario D2 represents deterministic service times equal to $\bar{\alpha}_{trans}$. In scenario D3, service time components are also deterministic, but the patient transport outcome α_{trans} is simulated in real time as a binary random variable with $\Pr\{\alpha_{trans} = 1\} = \bar{\alpha}_{trans}$. Patient transport is also random in the rest of the scenarios. In scenario D4, each service time component is random and exponentially distributed. Scenario D5 serves as our *realistic base-case* with stochastic travel times and transport times given by (4.24), (4.25) and (4.26), with the remaining service time components computed based on the empirical results given in Budge et al. (2010) and Alanis et al. (2013), which suggest log-normally distributed stochastic models for service time components with CVs around 0.394 (For each component, we can determine the scale and

location parameters of the distribution based on its known mean value). In scenarios D6 and D7, we test the effects of increased randomness in our realistic base-case by scaling the CV of each component by a factor of two and four, respectively. Note that while multiplying the CV by a given factor α, we also need to re-adjust the median travel times $m(d)$ to keep the mean values unchanged; that is

$$\hat{m}(d) = m(d)e^{\frac{(1-\alpha^2)c(d)^2}{2}},$$

$$\hat{c}(d) = \alpha \cdot c(d),$$

with $\hat{m}(d)$ and $\hat{c}(d)$ the parameters to use in the stochastic travel time model. Finally, in scenarios D8 to D11, we let the total service time be log-normally distributed with CVs equal to 0.5, 1, 1.5 and 2. This allows us to examine the impact of variability in random service times on the accuracy of the approximation.

The average errors in the estimation of various performance measures with different service time distributions described above are given in Tables B.2, B.3 and B.4. As expected, the loss system is almost insensitive to the service time distribution used in the simulation. It is more interesting, however, to observe the same level of insensitivity for the estimation of server workloads and dispatch rates for the system with queues as well.

Unlike server workloads and dispatch rates, the waiting times computed by the simulation model with non-exponential service times can substantially differ from those predicted by the approximation model, especially in more restricting coverage threshold scenarios where queues and waiting times are typically longer. More interestingly, the actual shape of the service time distribution seems to have no significant impact on the waiting time estimation accuracy compared with its CV. The error margins are comparable to the exponential case (D1) when the CV is equal to one (D9, D12 and D13) and grow rapidly as it deviates from one. This is in agreement with the original basic hypercube queuing model where the standard deviation of the service time is assumed to be approximately equal to its mean. Beyond that, however, the literature is scarce in studies on the impact of practical deviations from the exponential service times on the accuracy of the exact or approximate hypercube procedures and the existing works (Jarvis (1985) and Budge et al. (2009)) focus on loss systems. The simulation model developed by de Souza et al. (2015) to test their extended exact hypercube model, relaxes many of the restricting assumptions of the exact hypercube method of Larson (1974); however, the computational results reported are rather limited for our purpose as they only include the comparison of the simulation and model outputs for a case where the average waiting times and the fraction of the delayed customers are low. Nevertheless, they

report their hypercube model to be effective in their test cases.

We conclude that the approximation model can indeed be used to accurately predict the server workloads and dispatch rates of loss and queuing systems where service times are not necessarily exponentially distributed. However, the waiting times predicted by the model remain valid only if the coefficient of variation of the service time distribution is sufficiently close to one. Fortunately, in many practical applications, waiting times are of secondary importance compared to server workloads and dispatch rates; this is particularly true for systems designed to operate at low congestion levels where customers rarely have to wait.

Table B.1 Service time distribution scenarios used in the experiments.

Scenario	Specification
D1	Total service time exponentially distributed
D2	Deterministic service times
D3	Deterministic service times with random patient transport outcomes
D4	Each service time component exponentially distributed
D5	Realistic base case
D6	Realistic base case with CV multiplied by two
D7	Realistic base case with CV multiplied by four
D8	Total service time log-normally distributed with CV=0.5
D9	Total service time log-normally distributed with CV=1
D10	Total service time log-normally distributed with CV=1.5
D11	Total service time log-normally distributed with CV=2
D12	Total service time beta-distributed with shape parameters $a = 0.4$ and $b = 0.9333$ (CV=1)
D13	Total service time beta-distributed with shape parameters $a = 0.3$ and $b = 0.5571$ (CV=1)

B.2 Impact of Location and Priority Dependent Service Times

In this section, we investigate the effects of letting service times depend on call priority and server and customer locations on the approximation accuracy. To this end, we compare the outputs of the simulation model with service times depending on both location and priority with the values estimated by the approximation method with different scenarios in which service times are identical or depend on either location or priority or both. For each test problem, we average t_{ijp} values over all demand locations and servers to obtain service times that depend only on the call priority; that is

$$t_p = \frac{\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} b_{ijp}t_{ijp}}{\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} b_{ijp}}, \quad p \in \mathcal{P}\,;$$

Table B.2 Server workload estimation errors for different simulated service time distributions (in %).

	Loss						Queue					
Scenario	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6
D1	1.68	1.51	1.50	1.31	1.09	0.70	1.69	1.58	1.26	1.34	1.13	0.46
D2	1.59	1.46	1.48	1.29	1.08	0.72	1.55	1.51	1.19	1.21	1.07	0.48
D3	1.61	1.48	1.48	1.30	1.09	0.69	1.60	1.52	1.21	1.25	1.07	0.48
D4	1.67	1.49	1.49	1.30	1.12	0.70	1.68	1.57	1.26	1.31	1.13	0.44
D5	1.64	1.48	1.47	1.31	1.11	0.69	1.63	1.52	1.22	1.28	1.08	0.45
D6	1.68	1.49	1.49	1.31	1.11	0.72	1.66	1.54	1.24	1.28	1.10	0.44
D7	1.68	1.51	1.52	1.32	1.11	0.72	1.71	1.59	1.29	1.36	1.16	0.45
D8	1.64	1.48	1.49	1.30	1.10	0.72	1.63	1.52	1.21	1.29	1.09	0.46
D9	1.67	1.49	1.50	1.31	1.15	0.74	1.68	1.58	1.26	1.32	1.15	0.46
D10	1.70	1.51	1.53	1.33	1.13	0.73	1.72	1.64	1.29	1.38	1.19	0.47
D11	1.70	1.51	1.50	1.28	1.15	0.73	1.72	1.66	1.30	1.40	1.20	0.48
D12	1.67	1.49	1.50	1.29	1.10	0.73	1.66	1.55	1.23	1.29	1.10	0.45
D13	1.63	1.51	1.50	1.29	1.12	0.70	1.62	1.55	1.23	1.28	1.10	0.46

similarly, we take averages over priorities

$$t_{ij} = \frac{\sum_{p\in\mathcal{P}} b_{ijp} t_{ijp}}{\sum_{p\in\mathcal{P}} b_{ijp}}, \quad i \in \mathcal{I}, \quad j \in \mathcal{J}\,,$$

and the universal averages

$$t = \frac{\sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{P}} b_{ijp} t_{ijp}}{\sum_{p\in\mathcal{P}} b_{ijp}}\,,$$

as location-dependent and fixed service times, respectively.

Reported in Tables B.5 and B.6 are the corresponding estimation errors for server workloads and total dispatch rates in different coverage threshold scenarios. We can see that removing the service time dependency either on location or priority, significantly increases the estimation errors. In addition, the approximation using a fixed service time value independent of call location and priority, increases the error margins even further. This observation encourages the use of location and priority-specific service times to enhance the approximation of both loss and queuing systems, especially considering the little effort and negligible computational overhead associated with implementing travel time models available in the literature.

Table B.3 Total dispatch rate estimation errors for different simulated service time distributions (in %).

	Loss						Queue					
Scenario	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6
D1	5.43	4.31	3.32	3.23	2.47	0.96	5.42	5.19	4.01	3.97	3.19	1.20
D2	5.56	4.31	3.20	3.18	2.31	0.94	5.54	4.92	3.68	3.66	2.84	1.15
D3	5.54	4.30	3.24	3.21	2.37	0.95	5.53	4.93	3.80	3.70	2.92	1.16
D4	5.49	4.33	3.30	3.24	2.46	0.96	5.51	5.13	3.94	3.91	3.13	1.18
D5	5.53	4.30	3.29	3.22	2.42	0.95	5.53	5.02	3.82	3.75	2.98	1.16
D6	5.53	4.32	3.30	3.24	2.45	0.97	5.52	5.06	3.91	3.85	3.08	1.17
D7	5.46	4.32	3.35	3.25	2.48	0.97	5.48	5.25	4.11	4.05	3.22	1.22
D8	5.50	4.31	3.28	3.21	2.43	0.96	5.49	5.02	3.85	3.79	3.02	1.15
D9	5.43	4.32	3.31	3.22	2.46	0.97	5.44	5.21	4.06	4.00	3.18	1.21
D10	5.42	4.32	3.32	3.23	2.50	0.96	5.42	5.33	4.18	4.12	3.28	1.24
D11	5.41	4.33	3.34	3.26	2.50	0.96	5.43	5.37	4.28	4.21	3.33	1.27
D12	5.44	4.30	3.29	3.23	2.45	0.96	5.45	5.13	3.93	3.91	3.11	1.18
D13	5.46	4.30	3.30	3.22	2.43	0.96	5.46	5.08	3.90	3.86	3.05	1.16

B.3 Impact of System Workload

In this subsection, we look into the effect of varying system workloads on the performance of the algorithm. The errors in prediction of system workloads, dispatch allocations and waiting times for different coverage threshold scenarios with varying system workloads are given in Tables B.7, B.8, B.9 and B.10 with the corresponding average server workloads given in Table B.11. The *load factor* L is the scaling value used to modify the demand intensities of the original problem. In the case where queues are allowed, one or more servers may get saturated in some test problems; therefore, situations in which the number of these saturated cases exceeds 60 (out of 100) are marked.

We can conclude from the results that regardless of the system queue discipline or the coverage scenario, the error margins in prediction of the performance measures, do not significantly change with the system workload. The error margins in the estimation of server workloads and dispatch rates remain generally under 2% and 6%, respectively. The waiting time estimations are also within the range of 5% to 10%, except for some near saturation situations. We also note that for the full-backup loss system, the errors in the estimation of server workloads and dispatch rates are observed to take their maximum around the average workload of 0.5 and decrease as the average workload approaches 0 or 1.0. However, in most cases, a slight monotonous increase in the error margins with the load factor is observed. Overall, we expect the approximation model to be applicable in a wide range of system workloads as

Table B.4 Waiting time estimation errors for different simulated service time distributions (in minutes).

Scenario	C1	C2	C3	C4	C5	C6
D1	0.00	0.15	0.69	0.58	0.49	1.18
D2	0.00	0.12	1.81	2.32	5.87	19.01
D3	0.00	0.10	1.51	2.01	5.24	17.06
D4	0.00	0.10	0.71	0.86	2.42	7.49
D5	0.00	0.10	1.34	1.83	4.81	15.61
D6	0.00	0.10	0.96	1.28	3.50	11.09
D7	0.00	0.13	0.97	1.18	2.20	7.96
D8	0.00	0.09	1.24	1.67	4.44	14.31
D9	0.00	0.11	0.55	0.39	0.41	1.09
D10	0.00	0.22	2.59	3.24	6.90	22.02
D11	0.01	0.38	5.44	7.46	16.32	49.61
D12	0.00	0.26	1.32	0.90	0.81	1.53
D13	0.01	0.30	1.52	1.00	0.90	1.65

Table B.5 Server workload estimation errors with alternative scenarios of service time dependence on priority and location (in %).

	Loss						Queue					
Case	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6
Location and Priority	1.68	1.52	1.51	1.32	1.09	0.710	1.70	1.59	1.26	1.34	1.14	0.47
Location Only	2.22	2.38	2.67	3.21	1.21	2.01	2.24	2.13	1.88	2.27	1.54	1.80
Priority Only	6.55	2.76	2.32	2.30	2.06	1.25	6.66	2.38	1.60	1.83	1.65	1.15
None	8.65	4.19	3.58	3.91	1.99	2.25	8.85	3.90	2.69	2.92	2.45	2.12

long as none of the servers becomes saturated.

B.4 Computational Expense Reduction

Finally, we consider the computational expense of the algorithm and propose some strategies to reduce it. On a Linux system with 32 GB of RAM and a Core-i7-3770 CPU clocked at 3.4 GHz, the MATLAB implementation of the algorithm typically runs in less than a tenth of a second when no queues are allowed. However, when queues are allowed, it can take a few seconds to converge, depending on the termination condition and the problem at hand. Naturally, problems with more restricting coverage thresholds and hence significantly uneven distribution of delayed dispatches will require longer run times; these problems also tend to be the most sensitive to the values of $\kappa_{i'p'}^{ip}$.

Table B.6 Total dispatch rate estimation errors with different scenarios of service time dependence on priority and location (in %).

	Loss						Queue					
Case	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6
Location and Priority	5.44	4.31	3.32	3.23	2.47	0.970	5.42	5.19	4.01	3.98	3.20	1.20
Location Only	5.67	4.75	3.85	4.06	2.46	1.34	5.63	5.55	4.56	4.47	3.51	1.57
Priority Only	8.09	4.93	3.69	3.69	2.83	1.15	8.08	5.69	4.33	4.33	3.45	1.35
None	9.54	5.70	4.29	4.41	2.76	1.52	9.56	6.52	5.02	4.77	4.03	1.72

The two major time-consuming steps of the algorithm when queues are allowed are the updating of the residual service rates μ^D_{ijp} (including the prerequisite $\kappa^{ip}_{i'p'}$ and ρ^D_{ijp}) and the updating of the state probabilities P_n through the solution of the M/M/[N] model. Therefore, one approach to reduce the computational cost of the algorithm would be to reduce the frequency of performing these steps. As an example, we have conducted tests with random cases with 20 servers, different load factors and varying number of iterations between successive updates denoted by K_u. We have considered the top 10 percent of the test cases with the most sensitivity to the update frequency. Figure B.1 illustrates how the average estimation errors and computation times for these selected worst cases vary with the update interval. A trade-off between the computation time and the estimation accuracy can be seen, and interestingly, server workload estimations seem to be the most influenced, instead of the delayed dispatch rates that we would naturally expect.

Our experiments with different problems show that an integer in the range $0.1N_{iter}$ to $0.2N_{iter}$, where N_{iter} is the typical number of iterations required for convergence with a given termination threshold, should be a good starting point for the optimal value of the update interval K_u. Moreover, we suggest updating the state probabilities P_n whenever the residual service rates are updated.

Finally, as described in Section 4.3, the approach chosen to compute $\kappa^{ip}_{i'p'}$ will have a potentially significant impact on the computational expense of the algorithm depending on the problem characteristics and the implementation platform. In our experiments, using (4.18) led to slightly better estimations in most cases; however, using (4.17) was on average 50% faster after all possible code optimizations. In practical applications, the choice between the two methods should probably be based on their computational efficiency when implemented on a given platform rather than the small differences in estimation errors.

Table B.7 Server workload estimation errors for different load factors (in %).

	Loss						Queue					
L	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6
0.2	0.55	0.55	0.50	0.41	0.32	0.26	0.55	0.55	0.46	0.44	0.34	0.25
0.4	1.19	1.12	0.92	0.78	0.62	0.33	1.19	1.13	0.89	0.83	0.67	0.23
0.6	1.73	1.43	1.15	1.09	0.86	0.49	1.73	1.52	1.17	1.15	0.91	0.33
0.8	1.85	1.54	1.31	1.20	0.99	0.62	1.85	1.65	1.23	1.27	1.07	0.39
1.0	1.68	1.51	1.50	1.31	1.09	0.70	1.69	1.58	1.26	1.34	1.13	0.47
1.2	1.26	1.36	1.60	1.38	1.22	0.83	1.24	1.53	1.18	1.27	1.20	0.49
1.4	0.88	1.25	1.65	1.38	1.29	0.88	0.82	1.48	1.11	1.23	1.16	0.50
1.6	0.60	1.14	1.65	1.37	1.32	0.93	0.41	1.44	1.09	1.18	1.12	0.51*
1.8	0.42	1.02	1.63	1.34	1.32	0.97	0.23	1.29	0.83*	1.11*	0.91	0.56*

*Fewer than 40 unsaturated cases

Table B.8 Immediate dispatch rate estimation errors for different load factors (in %).

	Loss						Queue					
L	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6
0.2	1.26	1.26	1.05	0.95	0.74	0.38	1.26	1.26	1.09	0.99	0.78	0.40
0.4	2.69	2.55	2.01	1.82	1.34	0.53	2.69	2.59	2.16	1.98	1.49	0.62
0.6	3.92	3.60	2.76	2.62	1.85	0.70	3.90	3.78	3.14	3.00	2.26	0.89
0.8	4.78	4.12	3.15	3.08	2.21	0.84	4.77	4.54	3.90	3.71	2.87	1.13
1.0	5.44	4.31	3.32	3.23	2.47	0.97	5.41	5.17	4.30	4.10	3.29	1.38
1.2	5.96	4.44	3.36	3.39	2.53	1.07	5.82	5.73	4.53	4.41	3.57	1.62
1.4	5.64	4.31	3.41	3.45	2.65	1.14	5.13	5.91	4.59	4.65	3.78	1.79
1.6	4.82	4.02	3.36	3.44	2.70	1.20	4.05	5.44	4.17	4.20	3.74	1.95*
1.8	3.99	3.62	3.23	3.30	2.68	1.23	2.60	4.51	3.56*	3.93*	3.39	2.91*

*Fewer than 40 unsaturated cases

Table B.9 Delayed dispatch rate estimation errors for different load factors (in %).

L	C1	C2	C3	C4	C5	C6
0.2	0.00	0.01	0.12	0.09	0.13	0.21
0.4	0.00	0.08	0.44	0.26	0.33	0.39
0.6	0.00	0.30	0.98	0.55	0.65	0.65
0.8	0.03	0.68	1.69	0.96	1.04	0.91
1.0	0.10	1.16	2.33	1.33	1.43	1.17
1.2	0.30	1.73	3.01	1.72	1.89	1.45
1.4	0.77	2.24	3.59	2.06	2.35	1.67
1.6	1.41	2.86	3.71	2.12	2.75	1.74*
1.8	1.54	3.65	3.66*	2.67*	2.92	2.65*

*Fewer than 40 unsaturated cases

Table B.10 Waiting time estimation errors for different load factors, with the actual simulation values in parentheses (in minutes).

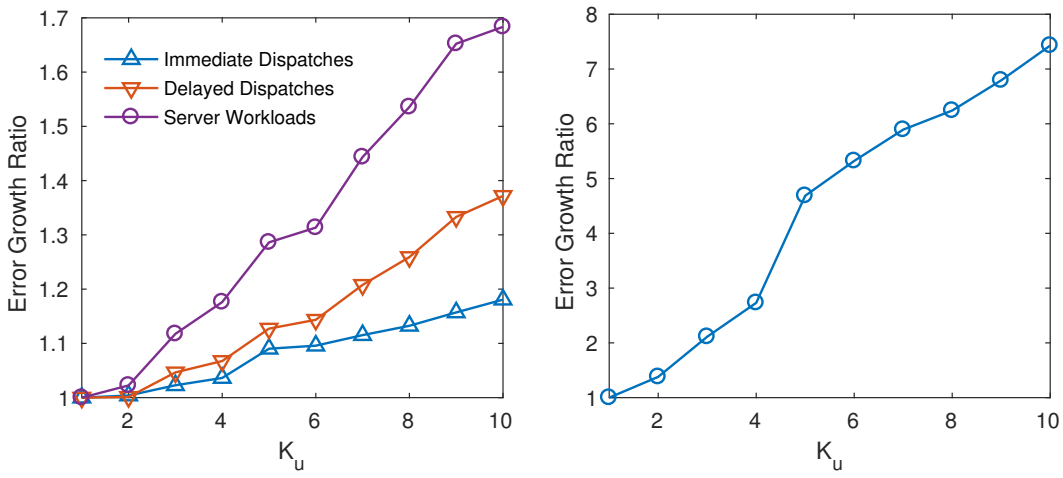
L	C1	C2	C3	C4	C5	C6
0.2	0.00 (0.00)	0.00 (0.00)	0.04 (0.27)	0.05 (0.80)	0.08 (1.88)	0.17 (4.12)
0.4	0.00 (0.00)	0.01 (0.03)	0.12 (0.82)	0.09 (1.75)	0.14 (4.05)	0.26 (9.43)
0.6	0.00 (0.00)	0.04 (0.12)	0.29 (1.85)	0.18 (2.87)	0.24 (6.49)	0.41 (16.49)
0.8	0.00 (0.00)	0.09 (0.34)	0.48 (3.80)	0.33 (4.37)	0.37 (9.49)	0.71 (27.06)
1.0	0.00 (0.02)	0.16 (0.80)	0.70 (6.72)	0.58 (6.26)	0.50 (13.22)	1.19 (40.35)
1.2	0.02 (0.19)	0.23 (1.83)	1.10 (10.85)	0.98 (9.48)	0.75 (18.31)	2.21 (50.99)
1.4	0.10 (1.10)	0.54 (4.25)	1.89 (17.59)	1.84 (15.34)	1.31 (23.67)	4.13 (67.91)
1.6	0.31 (5.11)	2.40 (9.86)	4.57 (28.38)	4.36 (26.27)	3.29 (33.28)	4.83 (65.05)*
1.8	1.14 (28.77)	17.41 (27.42)	15.97 (45.91)*	8.68 (43.79)*	9.26 (47.64)	3.43 (59.51)*

*Fewer than 40 unsaturated cases

Table B.11 Average server workloads for different load factors.

	Loss						Queue					
L	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6
0.2	0.10	0.10	0.10	0.09	0.09	0.07	0.10	0.10	0.10	0.09	0.09	0.07
0.4	0.19	0.19	0.19	0.18	0.18	0.15	0.19	0.19	0.19	0.18	0.17	0.15
0.6	0.29	0.29	0.29	0.27	0.26	0.22	0.29	0.29	0.29	0.28	0.26	0.22
0.8	0.39	0.39	0.39	0.37	0.36	0.30	0.39	0.39	0.39	0.37	0.35	0.30
1.0	0.49	0.49	0.49	0.46	0.44	0.37	0.49	0.49	0.49	0.46	0.44	0.37
1.2	0.59	0.59	0.58	0.56	0.53	0.44	0.59	0.59	0.58	0.56	0.53	0.45
1.4	0.69	0.69	0.68	0.64	0.63	0.52	0.70	0.69	0.68	0.65	0.62	0.52
1.6	0.80	0.79	0.78	0.74	0.72	0.59	0.80	0.79	0.78	0.75	0.71	0.59*
1.8	0.90	0.88	0.88	0.83	0.81	0.67	0.92	0.89	0.88*	0.85*	0.80	0.66*

*Fewer than 40 unsaturated cases



(a) Server workloads and dispatch rates

(b) Waiting times

Figure B.1 Variation of estimation errors and computation times with the update frequency.

APPENDIX C COMPARISON OF SIMULATION AND APPROXIMATION MODELS FOR ESS WITH RELOCATION

In this appendix, we compare the outputs of simulation and mathematical model presented in Chapter 6. Note that the simulation model used here follows the basic assumptions we made in deriving our mathematical model. In particular, it assumes uniform distribution of call locations and of the response units over the service region and hence can be considered an *idealized* simulation model. The primary goal of comparison with this idealized simulation model is to validate the correctness of the expressions comprising the mathematical model and also measure the extent of errors introduced by the use of Poisson distance distributions. We remember that distance distributions from the Poisson Point Process were used in the most parts of the mathematical analysis. However, a Poisson Point Process which assumes an infinite deployment area with independent number of points falling in any of its distinct subsets can not be faithfully recreated by a simulation model which places an exact number of N response units within a bounded service region. We thus use plot comparisons such as the ones given in this appendix to assess the magnitude of the errors caused by this discrepancy between the assumptions of the Poisson Point Process used in the mathematical model and the more realistic simulation model.

C.1 Loss Drone System

The first set of figures correspond to the drone loss system introduced in Section 6.3.1. In the interest of the space, we only include comparisons for a fleet size of $N = 2$, area scale factor of $f_A = \{1, 2, 4\}$, demand scale factor of $f_\lambda = \{1, 2\}$, and an expected outcome of $U_{\text{loss}} = 0.08$ for lost calls handed off to the alternative supporting system.

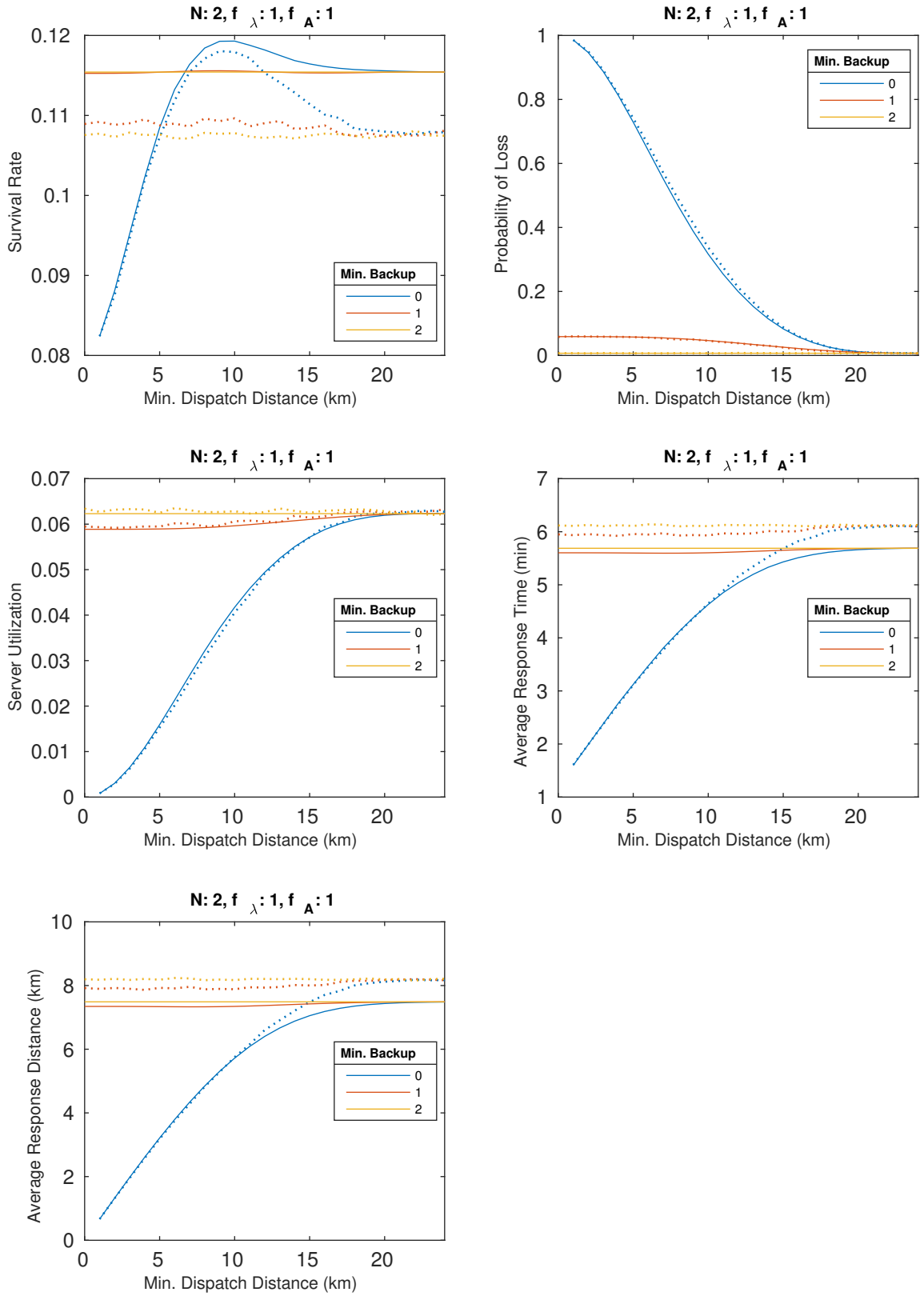


Figure C.1 Simulation versus model: loss system, $N = 2$, $f_\lambda = 1$, $f_A = 1$.

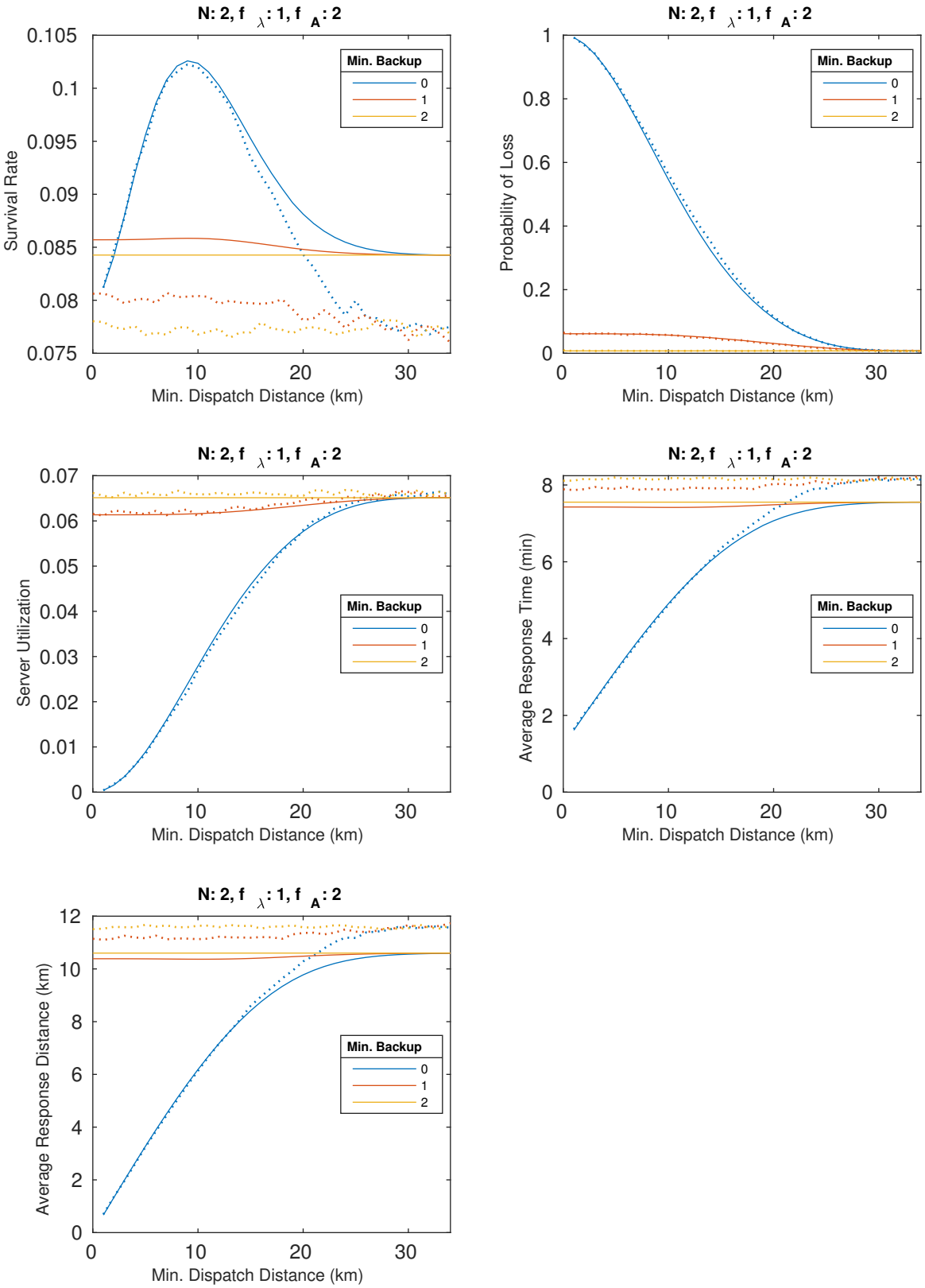


Figure C.2 Simulation versus model: loss system, $N = 2$, $f_\lambda = 1$, $f_A = 2$.

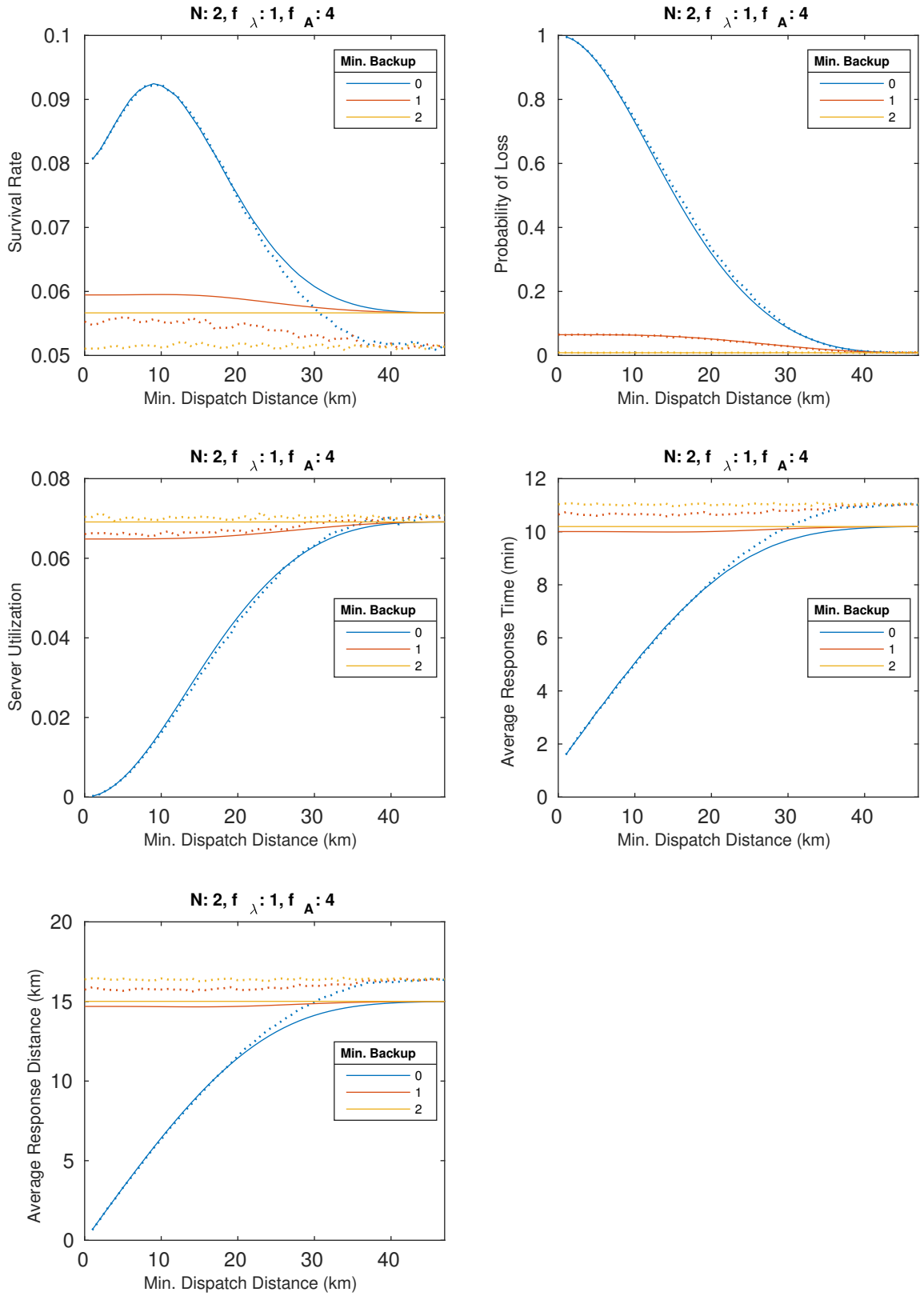


Figure C.3 Simulation versus model: loss system, $N = 2$, $f_\lambda = 1$, $f_A = 4$.

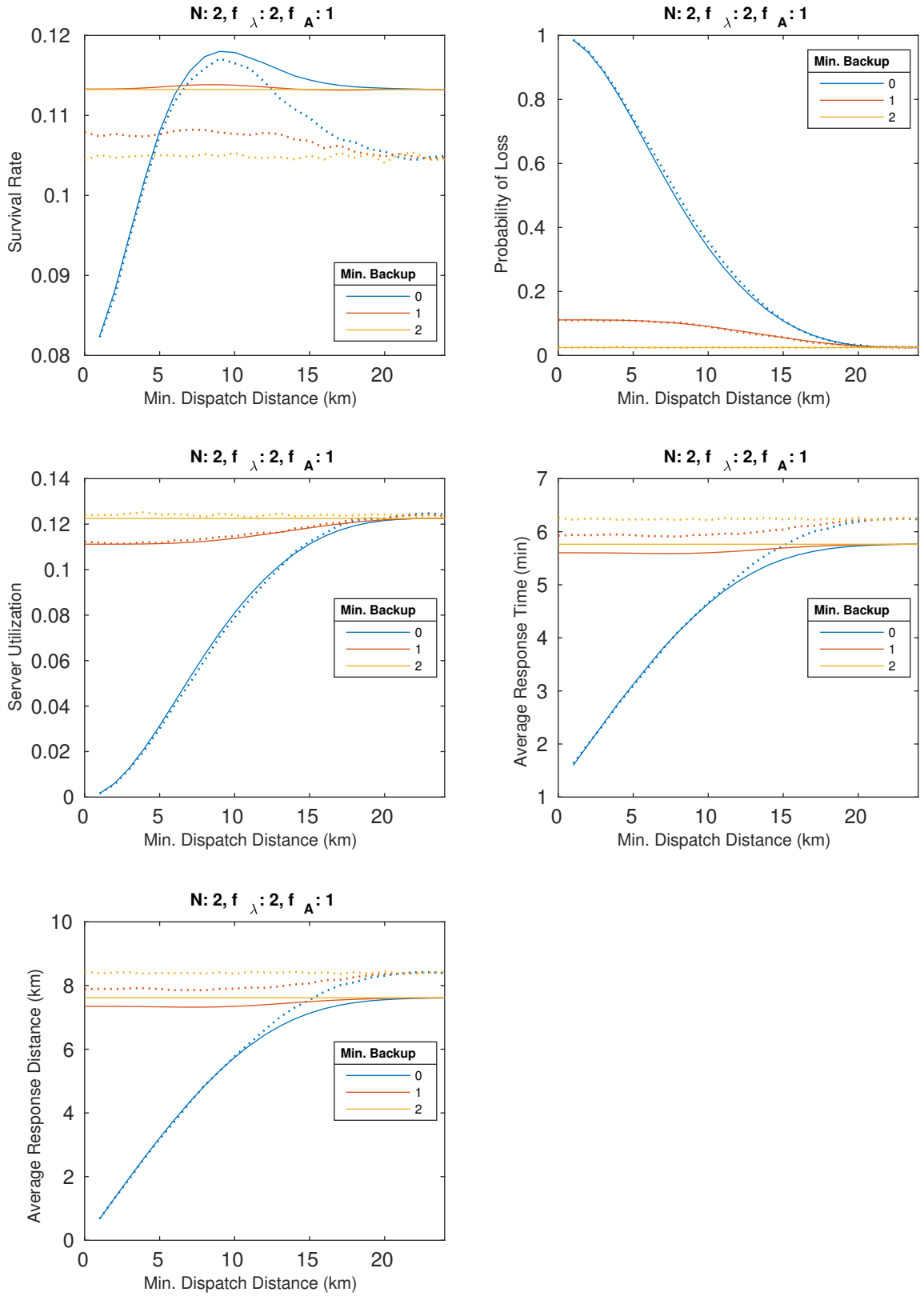


Figure C.4 Simulation versus model: loss system, $N = 2$, $f_\lambda = 2$, $f_A = 1$.

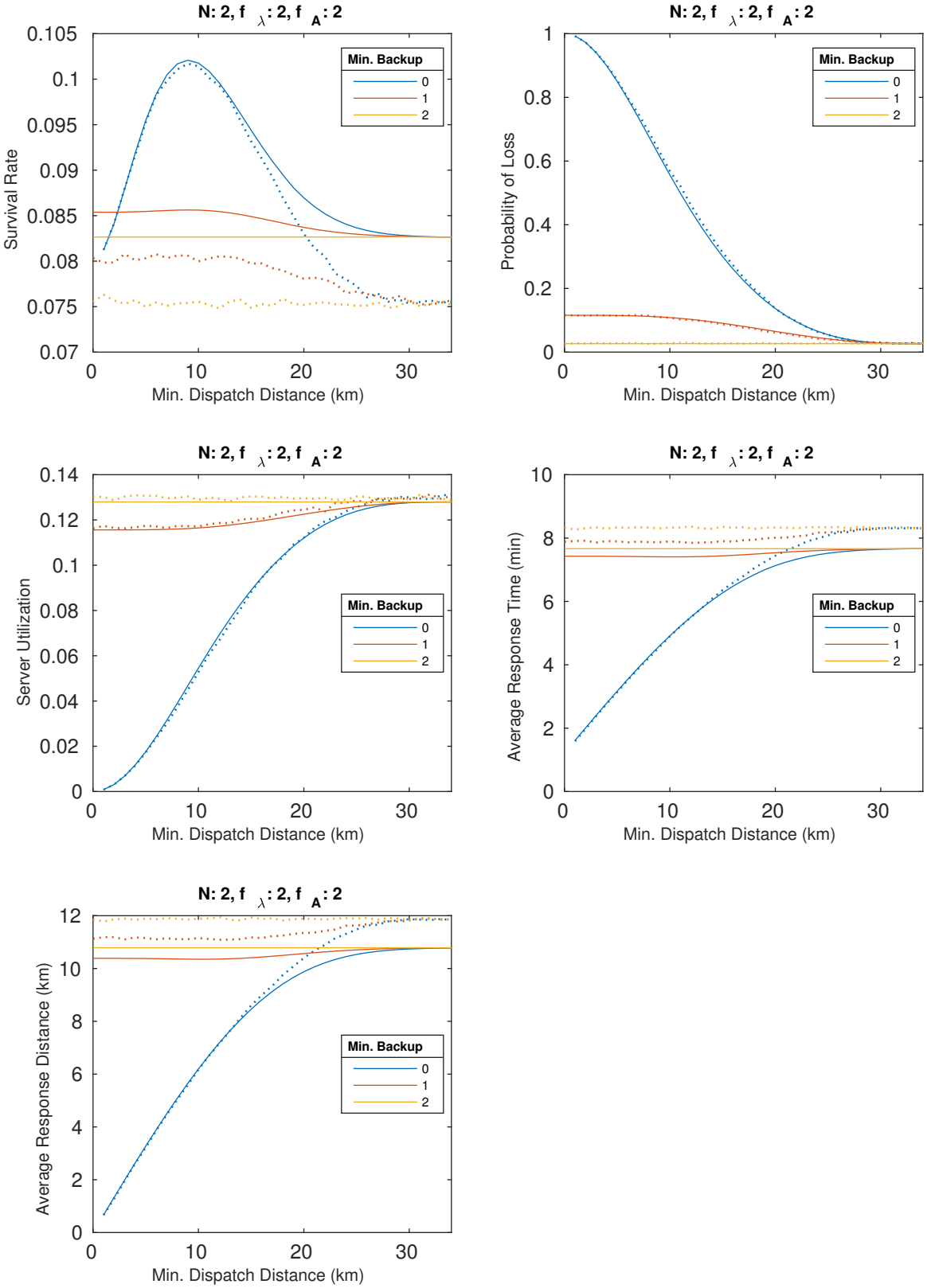


Figure C.5 Simulation versus model: loss system, $N = 2$, $f_\lambda = 2$, $f_A = 2$.

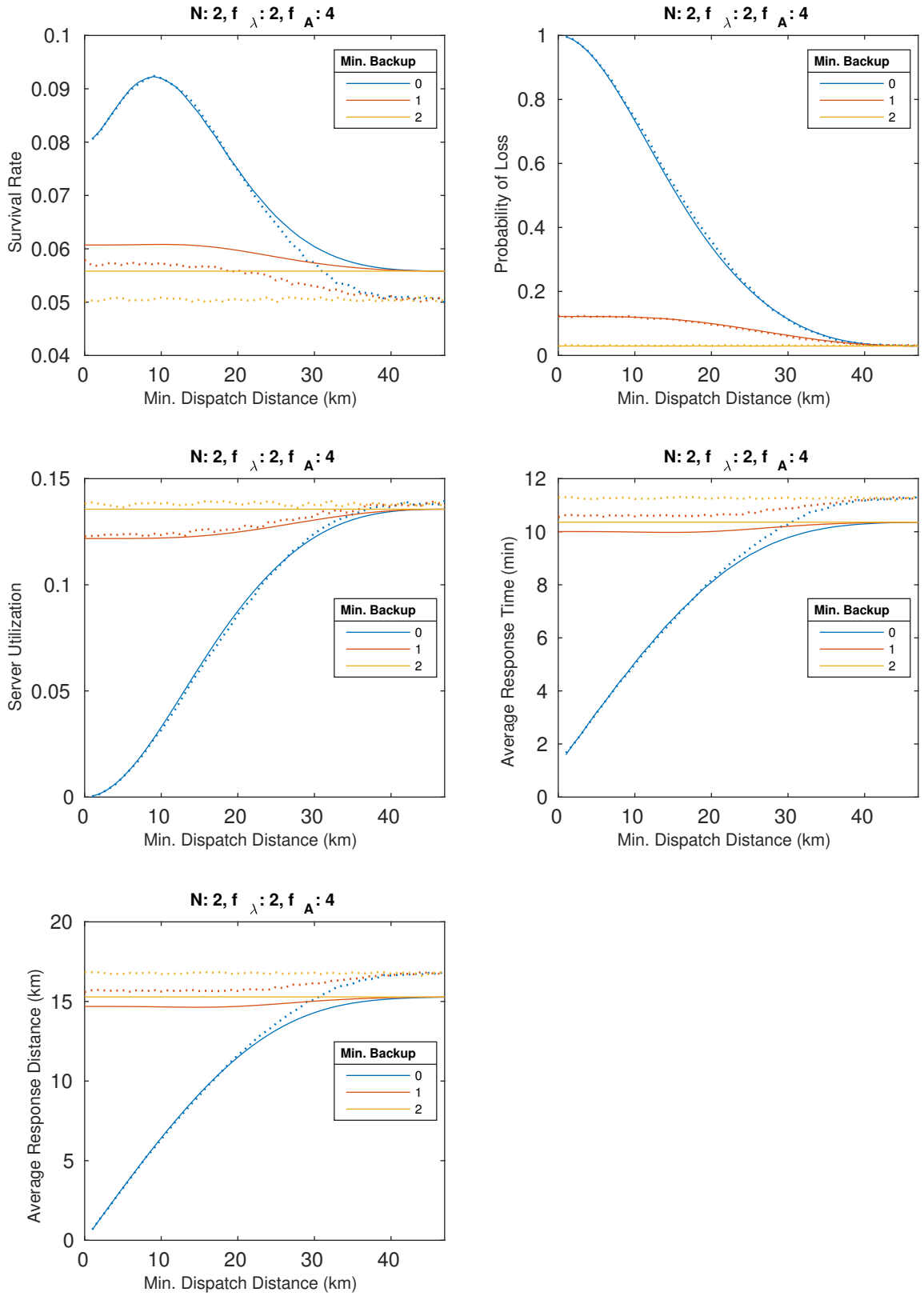


Figure C.6 Simulation versus model: loss system, $N = 2$, $f_\lambda = 2$, $f_A = 4$.

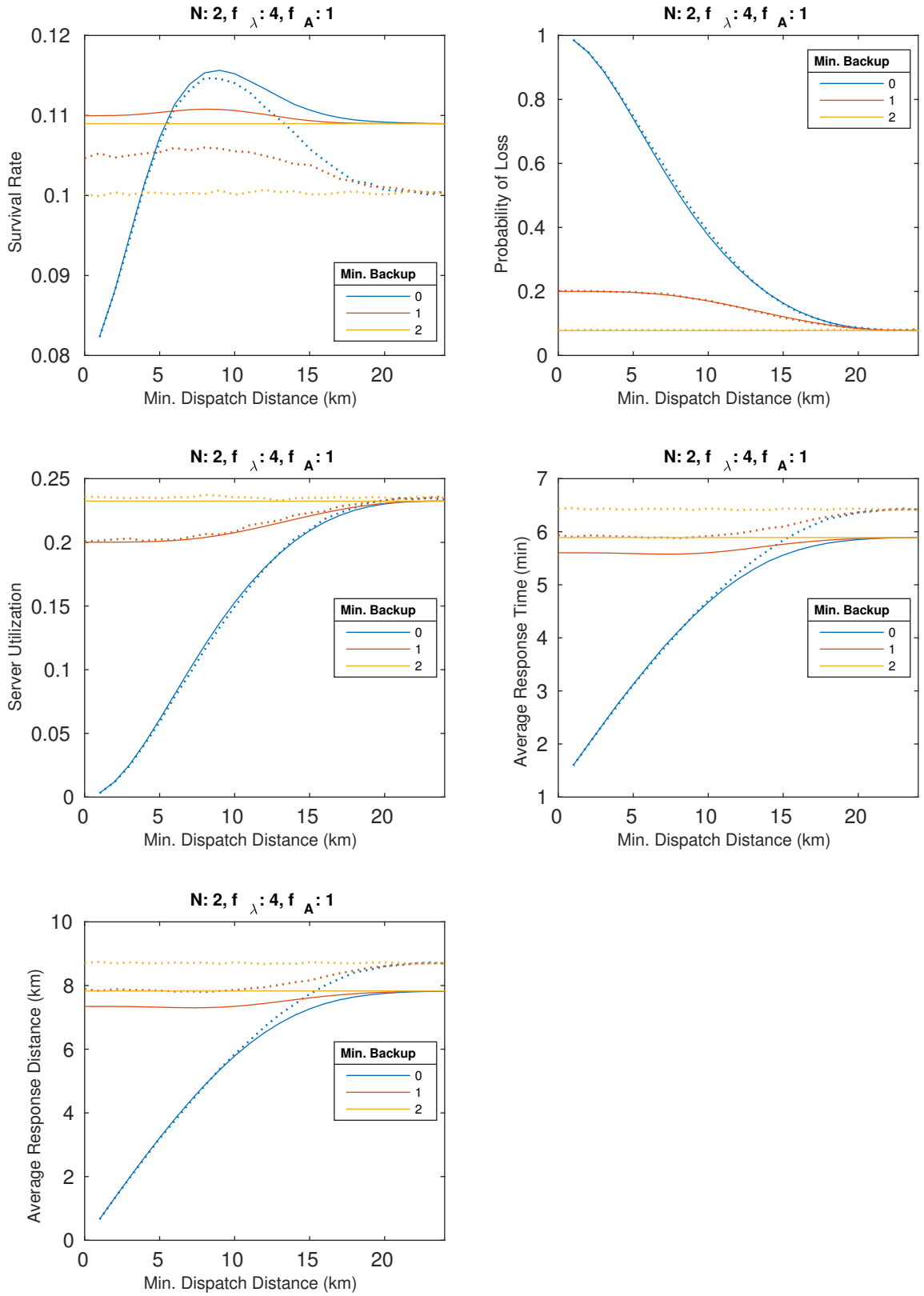


Figure C.7 Simulation versus model: loss system, $N = 2$, $f_\lambda = 4$, $f_A = 1$.

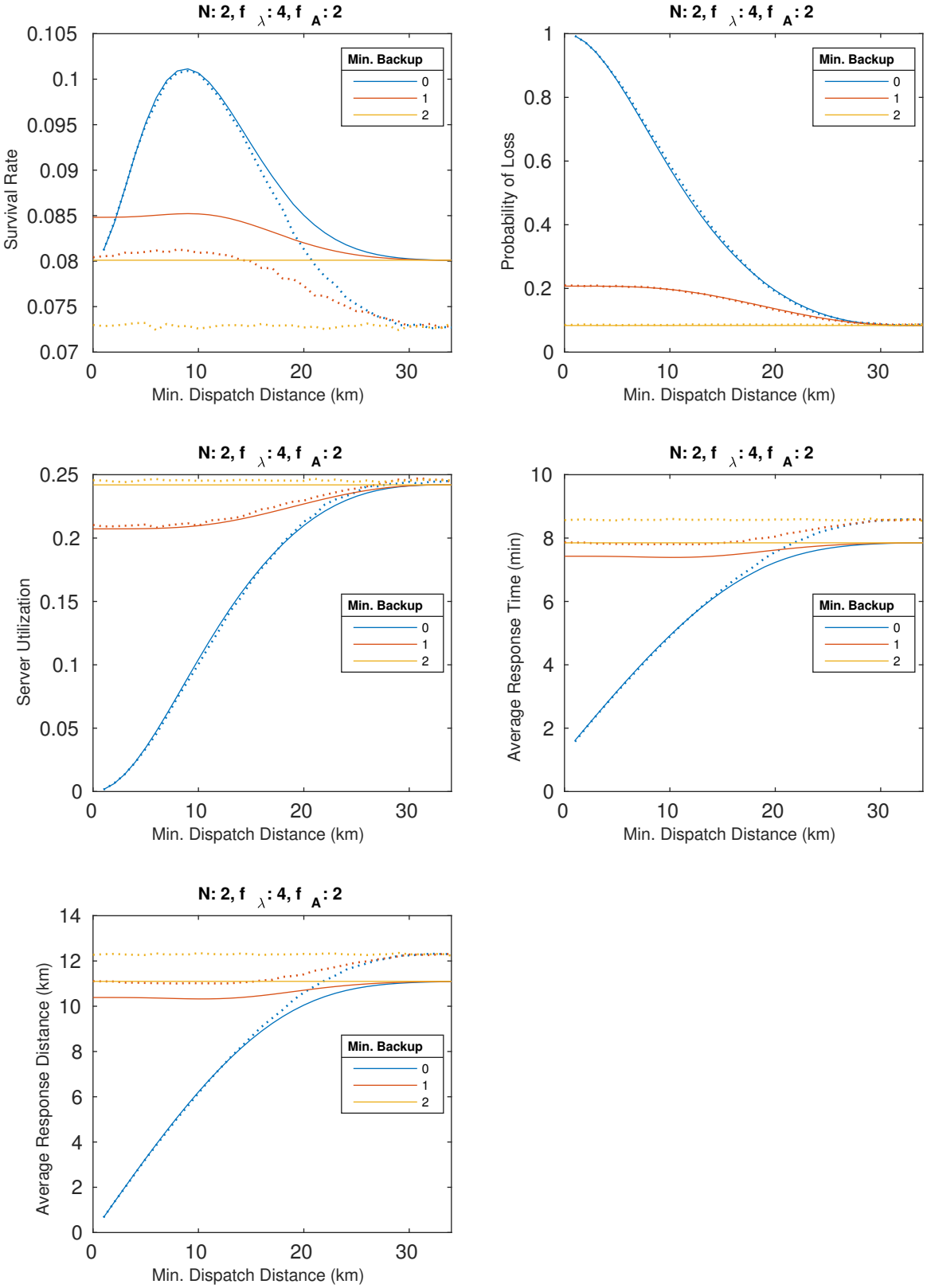


Figure C.8 Simulation versus model: loss system, $N = 2$, $f_\lambda = 4$, $f_A = 2$.

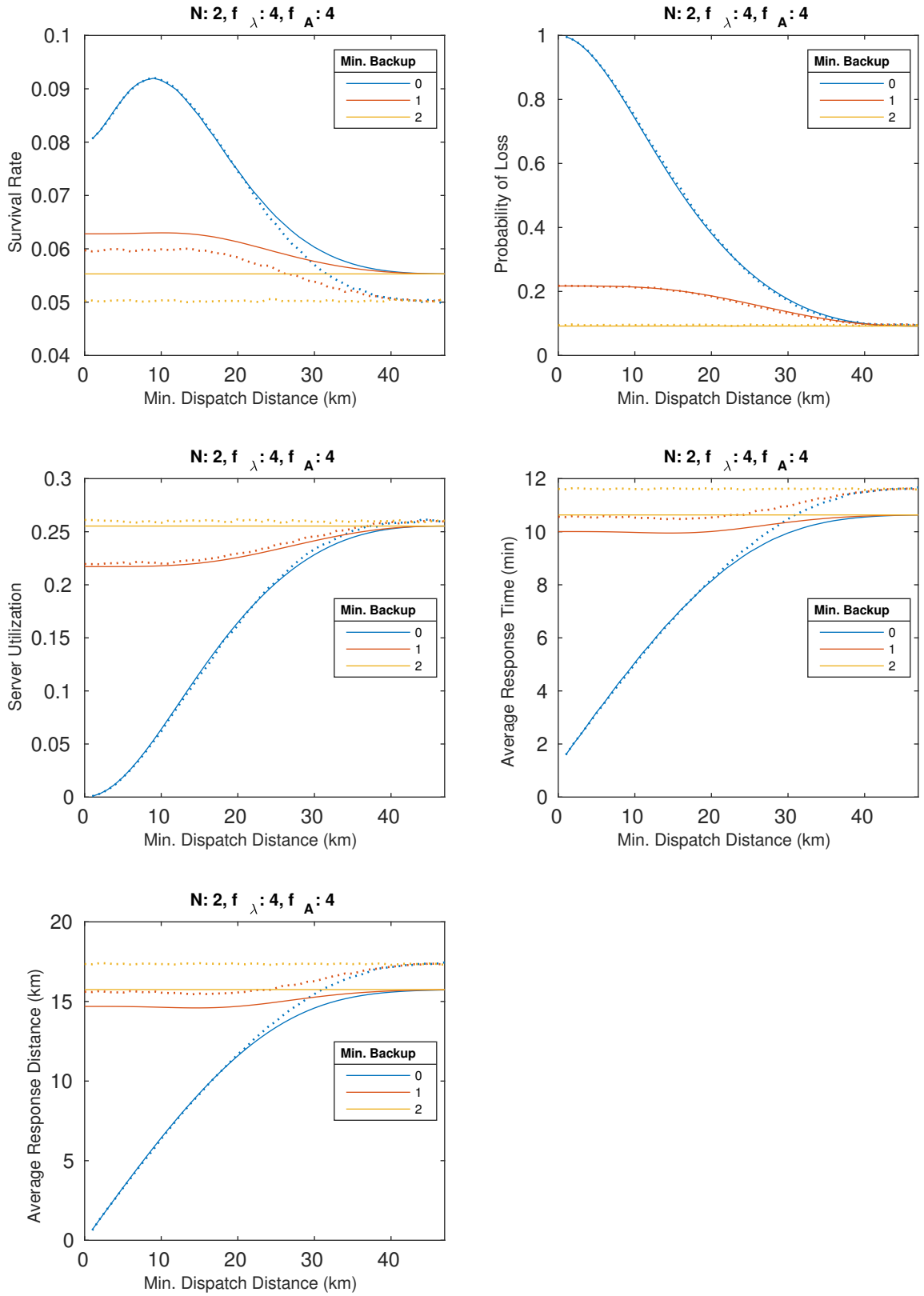


Figure C.9 Simulation versus model: loss system, $N = 2$, $f_\lambda = 4$, $f_A = 4$.

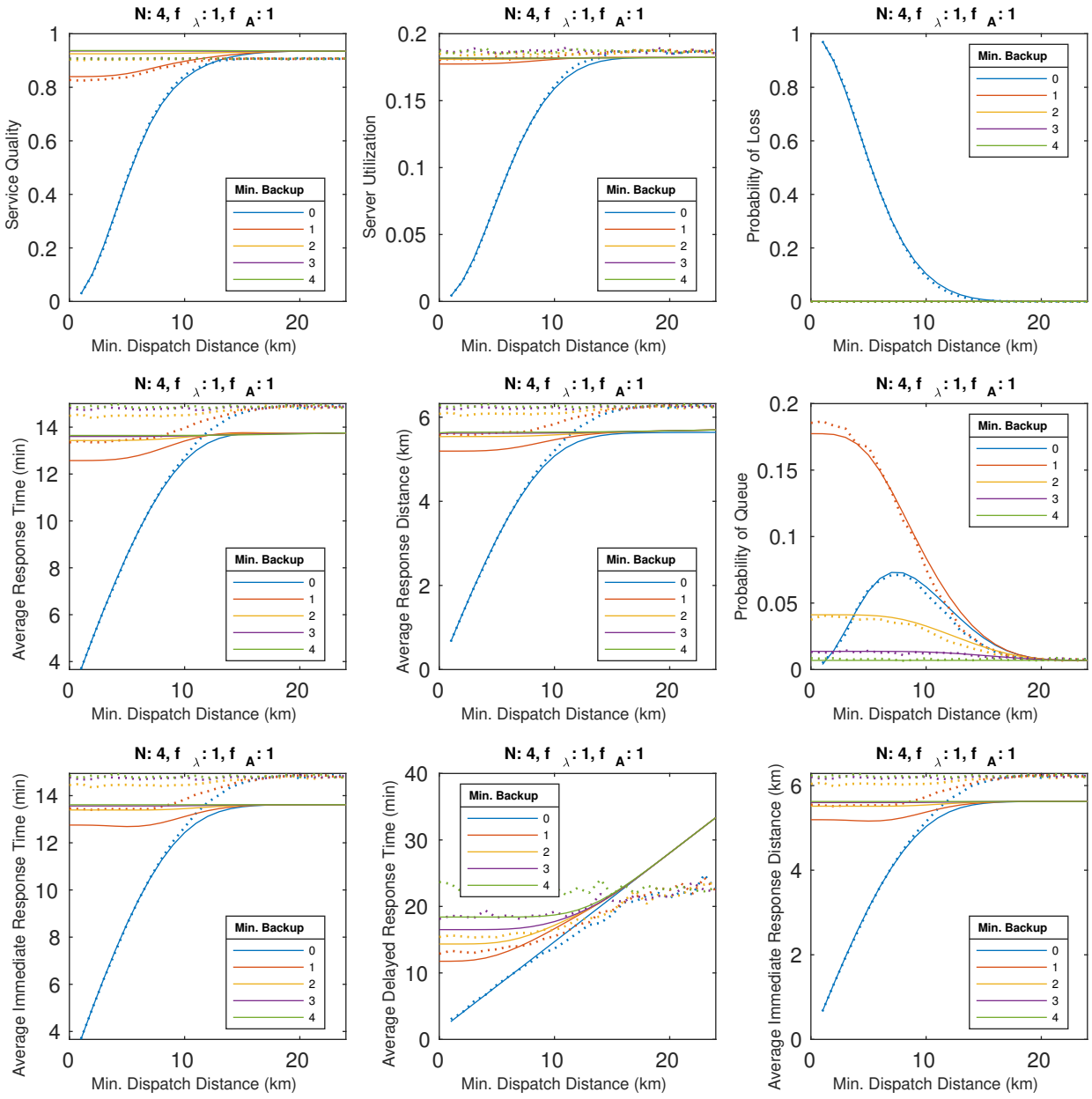


Figure C.10 Simulation versus model: queuing system, $N = 4$, $f_\lambda = 1$, $f_A = 1$.

C.2 Queuing EMS

The second set of figures correspond to the queuing system introduced as an example in Section 6.3.2. In the interest of the space, we only include comparisons for a fleet size of $N = 4$, area scale factor of $f_A = \{1, 2, 4\}$, demand scale factor of $f_\lambda = \{1, 2\}$, and an expected outcome of $U_{\text{loss}} = 0$ for lost calls.

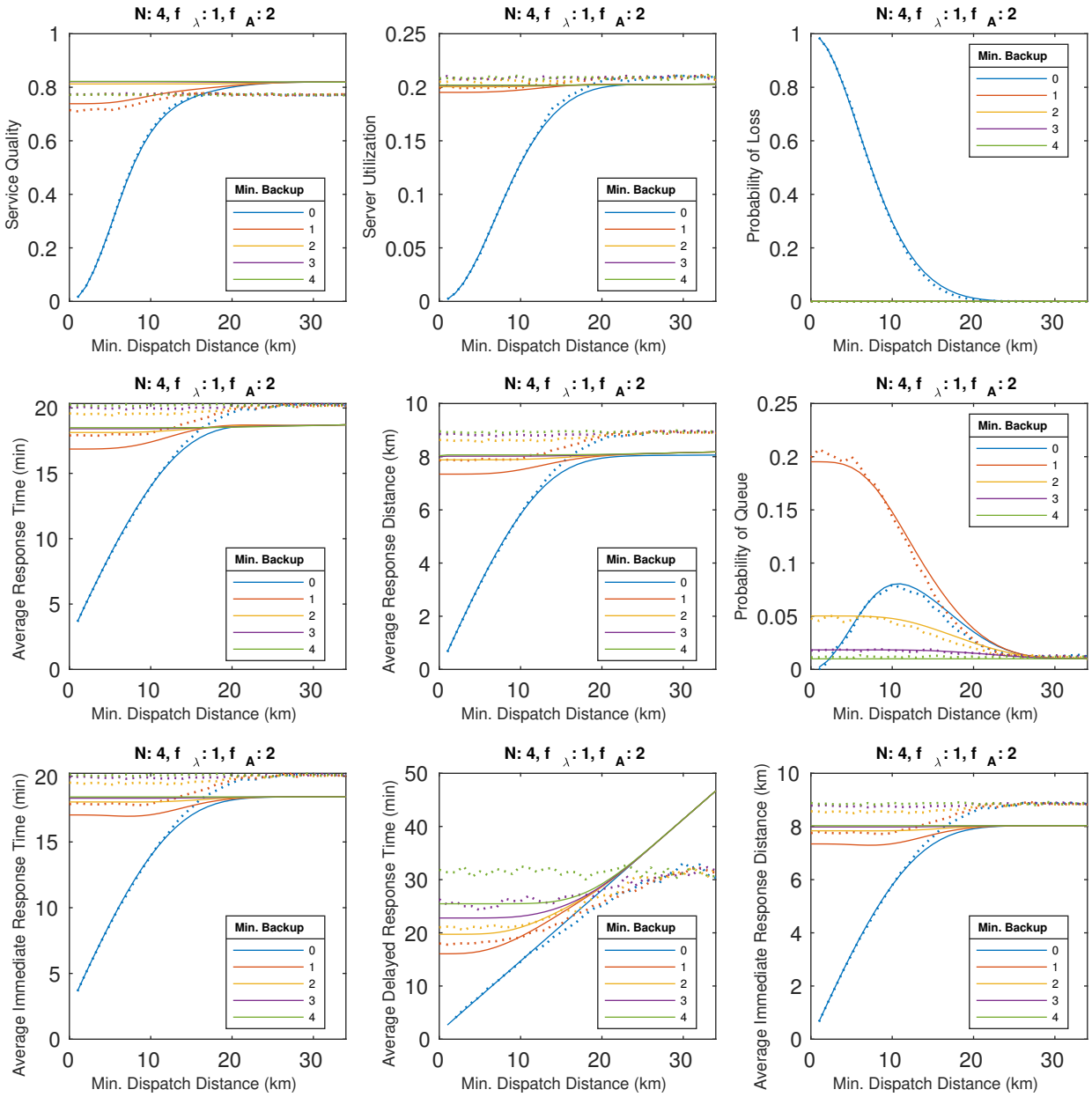


Figure C.11 Simulation versus model: queuing system, $N = 4$, $f_\lambda = 1$, $f_A = 2$.

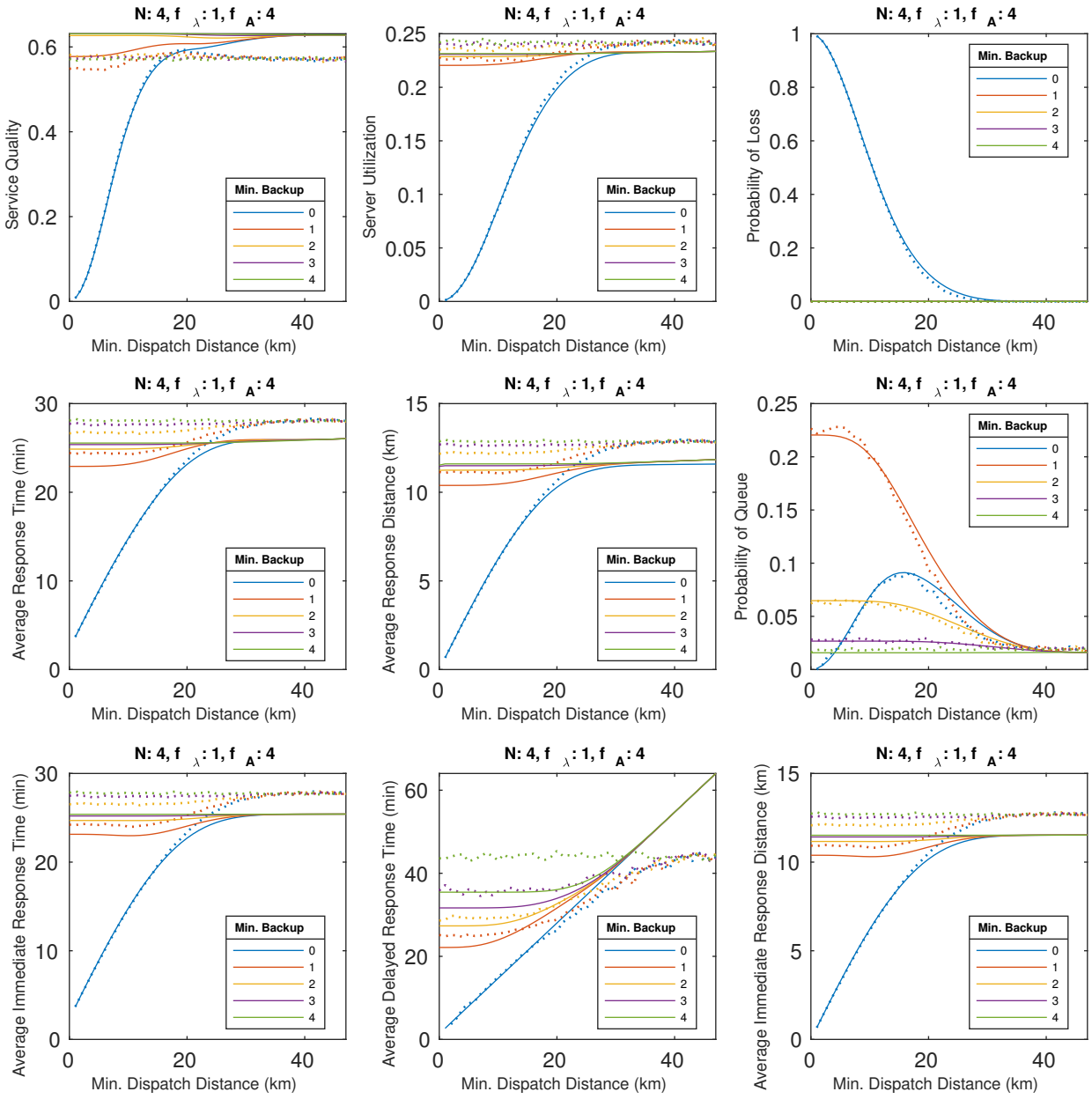


Figure C.12 Simulation versus model: queuing system, $N = 4$, $f_\lambda = 1$, $f_A = 4$.

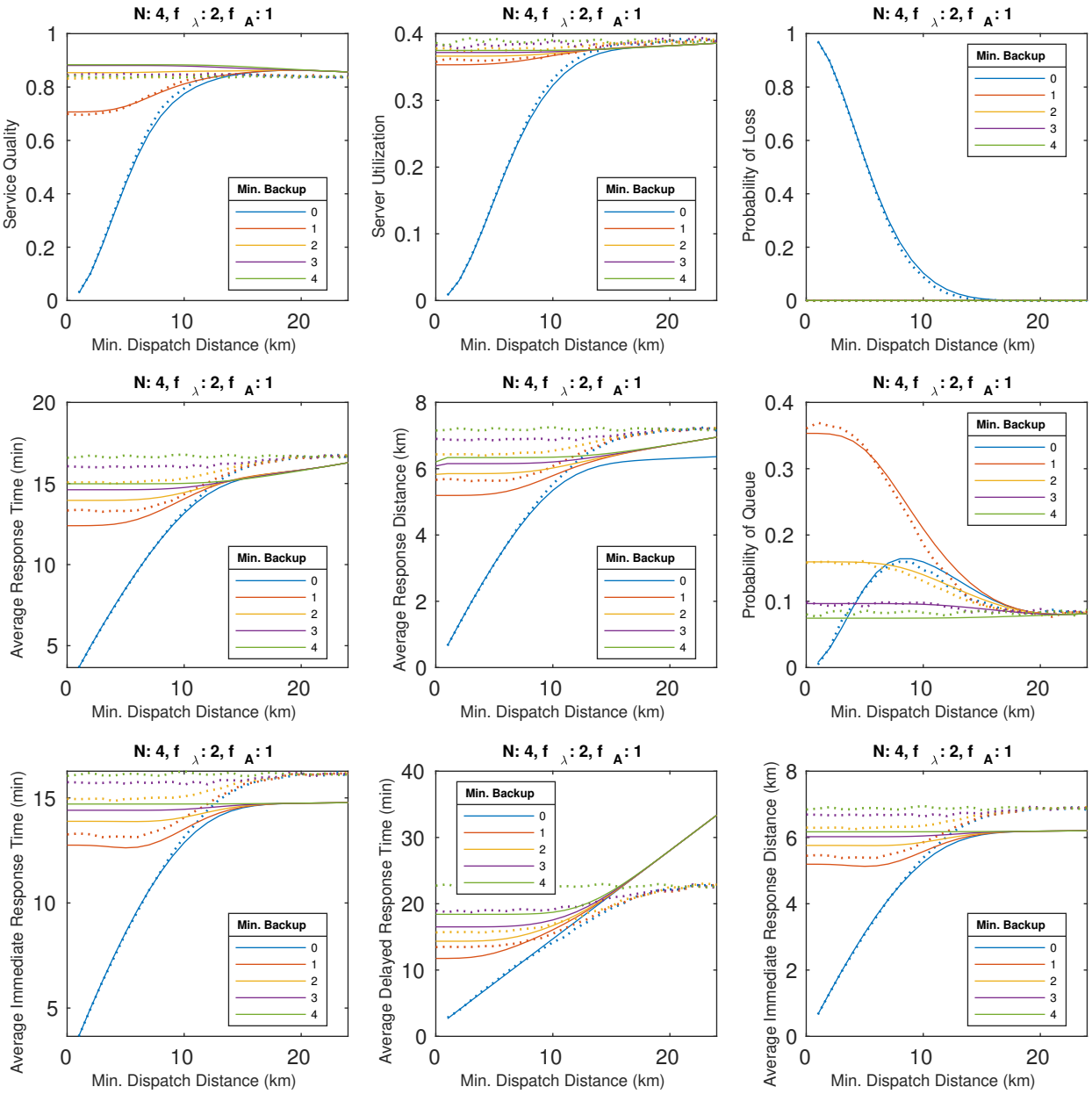


Figure C.13 Simulation versus model: queuing system, $N = 4$, $f_\lambda = 2$, $f_A = 1$.

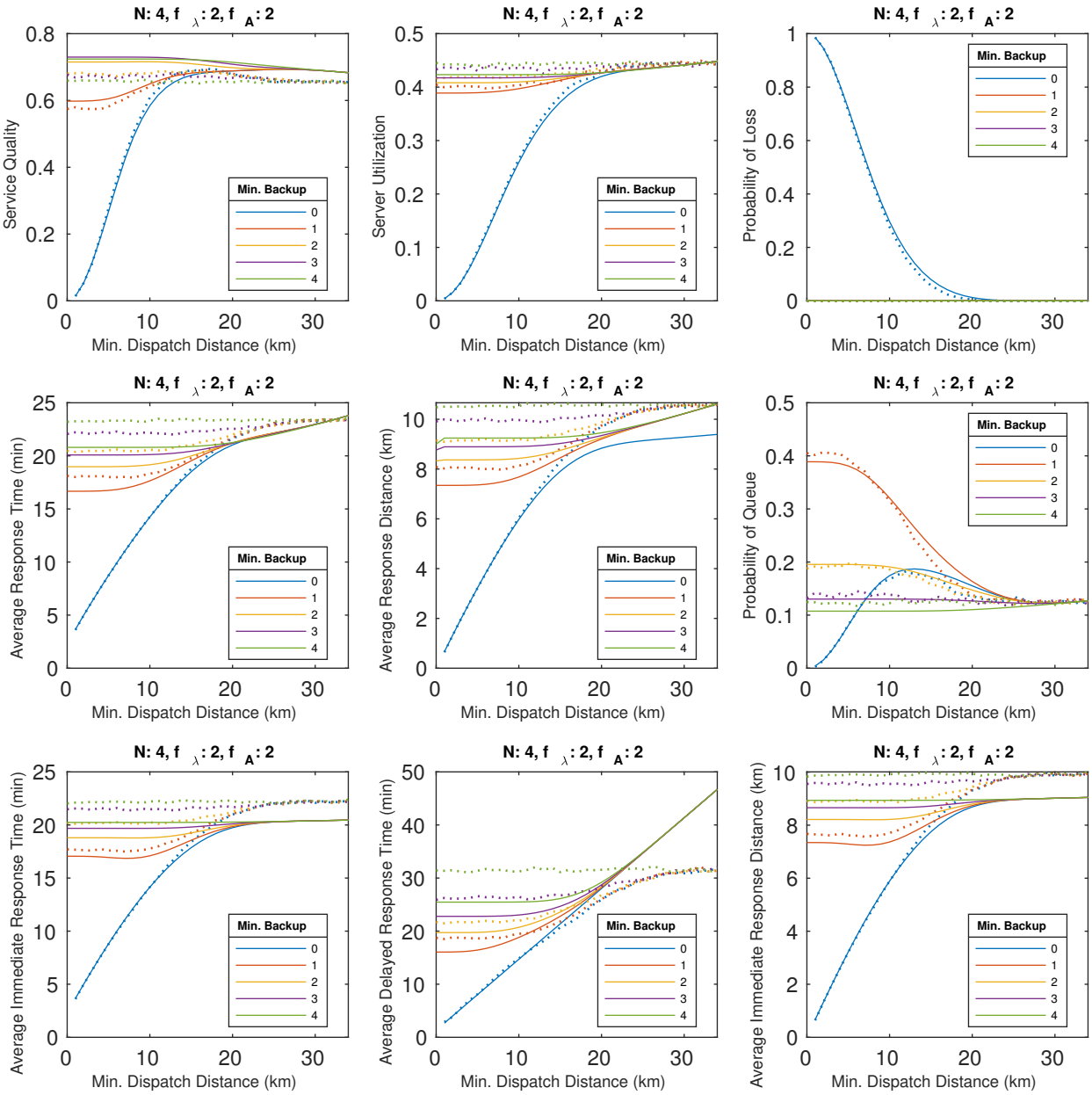


Figure C.14 Simulation versus model: queuing system, $N = 4$, $f_\lambda = 2$, $f_A = 2$.

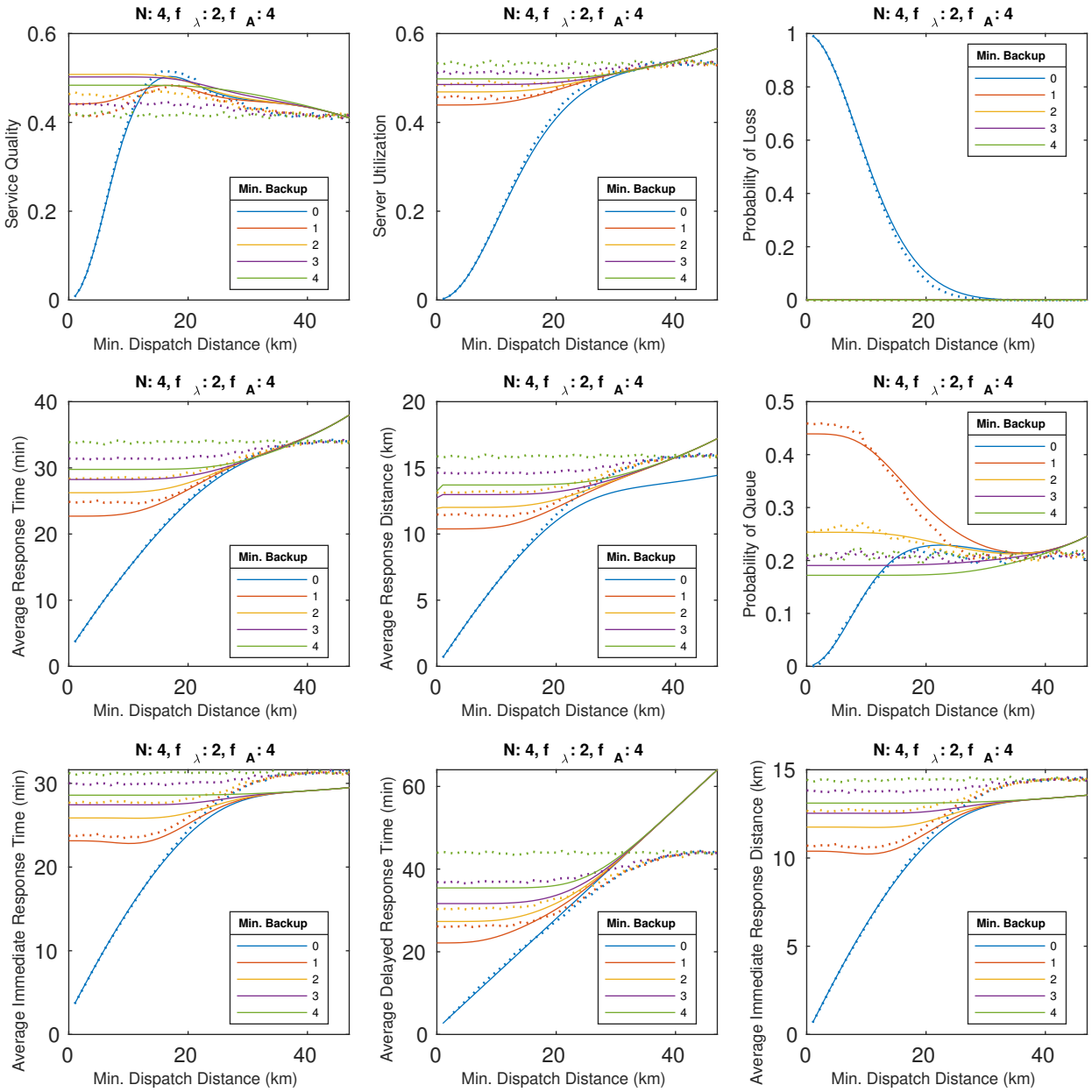


Figure C.15 Simulation versus model: queuing system, $N = 4$, $f_\lambda = 2$, $f_A = 4$.