Concept demonstrator: Holding site location, ambulance allocation, and relocation decision support tool.

A.C. Oosthuizen

Department of Industrial Engineering

Stellenbosch University

Thesis presented in fulfilment of the requirements for the degree of Master of Engineering in the Faculty of Engineering at Stellenbosch University.

M.Eng. (Research) Industrial

Supervisor: Louzanne Bam
Co-Supervisor: Prof. Jan H. van Vuuren

March 2017

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work; that I am the sole author thereof (save to the extent explicitly otherwise stated); that reproduction and publication thereof by Stellenbosch University will not infringe any third-party rights; and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 2017

Acknowledgements

I would first like to acknowledge my thesis supervisor Louzanne Bam. She provided me with the needed guidance, but allowed this paper to be my own research and work. Also, my language editor Liezel Visser, who was so patient with my repeated grammar mistakes and extremely thorough.

I would like to thank Herman van der Westhuizen for sending my thesis proposal to acquaintances at the Western Cape Emergency Control Centre. This project would not have been possible without this introduction. I also need to acknowledge the experts at the Western Cape Emergency Control Centre who saw the worth of this project and helped me in gaining access to the required data: Dr. Shaheem de Vries, Prof. Lee Wallis, Dr. Julian Fleming, and Calvin Naidoo. Without their participation and input, this project would not have been successful.

Finally, I have to convey my gratitude to my parents, my boyfriend, and my flatmates for supporting and believing in me. The continuous encouragement throughout this entire process of researching and writing helped to keep me going. I would not have been as calm and collected throughout without your support. Thank you.

A.C. Oosthuizen

Abstract

Before the start of a shift, the dispatchers at the Western Cape Emergency Control Centre (WC ECC) decide where to place holding sites and how many ambulance to allocate to each holding site. During a shift they decide when and where to relocate ambulances. At present, dispatchers make these decisions based solely on their experience and intuition.

In this project a concept demonstrator decision support tool (DST) is developed which produces solutions for the near-optimal placement of holding sites per shift, ambulance allocation, and relocation per hour of that shift based on predicted ambulance demand rates. The DST is developed with the aim of assisting the dispatchers at the WC ECC with holding site placement, ambulance allocation, and relocation decisions.

The real-world instance utilised during the development of the concept demonstrator DST consists of six months' historical call data from the City of Cape Town and the Cape Winelands municipalities. Singular spectrum analysis is used to forecast ambulance demand according to incident priority. The extended queuing maximum availability location problem model is adapted to fit the real-world instance. The model aims to simultaneously maximise expected ambulance coverage and minimise ambulance relocations by manipulating holding site placement, ambulance allocation, and relocation. The solution method implemented for the model as a whole is the artificial bee colony algorithm.

The DST was solved for four planning week instances, at 95% service reliability. Predicted demand for the planning week is predicted using historical demand that precedes the planning week and a recommended schedule of holding site placement, ambulance allocation, and relocation is generated for the predicted ambulance demand. The performance of this schedule is evaluated using the observed historical demand for the planning week.

Different approaches for the classification of calls – consider all calls to be life-threatening, or calls to be life-threatening or non-life-threatening – as well as for the implementation of the model constraints are considered. The results indicate that the WC ECC can improve ambulance coverage with the current, or even smaller, ambulance fleet size if decisions are made with the assistance of the DST that anticipates the probable future ambulance demand.

The concept demonstrator DST's solutions' expected percentage coverage compared to the actual percentage coverage exceeds 150%. However, it is invalid to compare these values like-for-like as a significant number of real-world factors, including the specific road conditions at the time of each call, the responsiveness of both the ECC operator handling the call and the ambulance team involved, and the communication connection between the ECC call operator and the ambulance team, influence the real-world response rate and could not be modelled in the DST. However, even when these factors are taken into account, the discrepancy between the actual and the predicted performance is sufficient to convincingly demonstrate the potential of the concept demonstrator DST to assist the WC ECC in further improving their response time.

Opsomming

Voor die aanvang van 'n skof besluit die ambulaansversenders by die Wes-Kaapse noodbeheersentrum (WC ECC) waar om wagstasies te plaas en hoeveel ambulanse om by elkeen te plaas. Tydens 'n skof besluit hulle wanneer en waarheen ambulanse geskuif moet word. Tans, maak die versenders staat slegs op hul eie ervaring en intuïsie om hul besluitneming te lei.

In die projek is 'n konsep demonstreerder besluitsteunstelsel (DST) gebou wat oplossings vir die naas-optimale plasing van wagstasies per skof, ambulaansplasing en -rondskuiwing per uur van daardie skof bepaal gebaseer op voorspelde ambulaansaanvraag. Die konsep demonstreerder DST is ontwikkel met die doel om die versenders by die WC ECC te help met die besluitneming aangaande wagstasieplasing, ambulaansplasing en -rondskuiwing.

Die werklikheidsgeval, waarvoor die DST ontwikkel word, bestaan uit ses maande se historiese oproepdata van die Stad Kaapstad en die Kaapse Wynland munisipaliteite. 'Singular spectrum analysis' is gebruik om die ambulaansaanvraag volgens voorvalprioriteit te voorspel. Die uitgebreide 'queuing maximum availability location problem' model is aangepas om by die werklikheidsgeval te pas. Die model streef om die maksimum verwagte ambulaansdekking en die minimum rondskuiwingskoste deur middel van verbeterde wagstasieplasing, ambulaansplasing en -rondskuiwing te vind. Die oplossingsmetode wat gebruik is vir die algehele model is die 'artificial bee colony' algoritme.

Die DST is vir vier gevalle opgelos met 'n 95% diensbetroubaarheidsvlak. Die ambulaansaanvraag vir die beplanningsweek is voorspel gebaseer op historiese ambulaansaanvraag, wat nie die beplanningsweek se historiese ambulaansaanvraag bevat nie. Daarna is'n aanbevole wagstasieplasing, ambulaansplasing en -rondskuiwing skedule gegenereer vir die voorspelde ambulaansaanvraag. Die skedule is geïmplimenteer vir die beplanningsweek se historiese ambulaans aanvraag. Die resultate is gebruik om die skedule se prestasie the evalueer. Verskillende benaderings vir die hantering van die oproepe volgens voorvalprioriteit – ag alle oproepe as lewensbedreigend, of ag hulle as lewensbedreigend of nie-lewensbedreigend – en twee implementerings van die ambulaansplasingsbeperking word oorweeg. Die resultate dui aan dat die WC ECC die ambulaansdekking kan verbeter met die huidige, of selfs kleiner, ambulaansvloot as besluite geneem word met behulp van die konsep demonstreerder DST in afwagting van die waarskynlike ambulaansaanvraag.

Die DST se oplossings se verwagte persentasie ambulaansdekking oorskry die werklike persentasie ambulaansdekking wat bepaal is vir die historiese oproepdata met 150%. Dit moet inaggeneem word dat hierdie waardes nie dieselfde is nie. Beduidende gevalle van die werklikheidsgeval se faktore, insluitend die spesifieke toestand van die paaie tydens elke oproep, die fluksheid van die noodbeheersentrum se telefoonoperateur en die ambulaansbemanning, en die kommunikasie tussen die telefoonoperateur en die ambulaansbemanning, beïnvloed die werklike reaksietyd en kon nie gemodelleer word nie. Tog, selfs wanneer die faktore inaggeneem word, is die verskil tussen die waargenome en voorspelde prestasies voldoende om oortuigend die potensiaal van die konsep demonstreerder DST te demonstreer as hulpmiddel vir die WC ECC om hul reaksietye verder te verbeter.

Contents

List of figures

List of tables

Nomenclature

Acronyms

ABC	Artificial Bee Colony
ALS	Advanced Life Support
BACOP1	Backup Coverage model 1
BACOP2	Backup Coverage model 2
BI	Business Intelligence
BLS	Basic Life Support
CAD	Computer-aided Dispatch
CPM	Corporate Performance Management Systems
DACL	Dynamic Available Coverage Location model
DADP	Dominican Ambulance Deployment Problem
DoH	Department of Health
DSM	Double Standard Model
DST	Decision Support Tool
DW	Data Warehousing
DYNAROC	Dynamic Ambulance Relocation Model
ECC	Emergency Control Centre

EIS	Executive Information Systems
EMS	Emergency Medical Services
EOR	Emergency-related Operational Research
FIFO	First-in first-out
FLEET	Facility-location Equipment-emplacement Technique
GA	Genetic Algorithm
GIS	Geographical Information System
GPS	Global Positioning System
GSS	Group Support System
HOSC	Hierarchical Objective Set Covering problem
IDST	Intelligent Decision Support Tool
IFT	Inter-facility Transport
IS	Information Systems
IT	Information Technology
KMDST	Knowledge Management-based Decision Support Tool
MALP	Maximal Availability Location Problem
MAPE	Mean Absolute Percentage Error
Matlab	Matrix laboratory
MCLP	Maximal Covering Location Problem
MDG	Millennium Development Goals
mDSM	Multi-period Double Standard Model
MECRP	Maximal Expected Relocation Problem

MERLP	Minimum Expected Response Location Problem
MESLMHP	Maximal Expected Survival Location Model for Heterogeneous Patients
Metro	Medical Emergency Transport and Rescue Organization
MEXCLP	Maximum Expected Covering Location Problem
MIFT	Maternal Inter-facility Transport
MIT	Massachusetts Institute of Technology
MMLCP	Maximal-multiple Location Covering Problem
MMR	Maternal Mortality Rate
MOFLEET	Variant of MEXCLP
MOTEM	Multi-objective Equipment allocation Model
MPBDCM	Multi-period Backup Double Covering Model
MS	Microsoft
NaN	Not a number
NSS	Negotiation Support System
OR	Operational Research
P1	Priority 1
P2	Priority 2
PDST	Personal Decision Support Tool
PLSCP	Probabilistic Location Set Covering Problem
PROFLEET	Probabilistic version of FLEET
Q-MALP	Queuing Maximum Availability Location Problem
Q-PLSCP	Queuing Probabilistic Set Covering Problem

RMSE	Root-mean-square error
RP^t	Ambulance location problem based on DSM
RtMvGcRM	Real-time Multi-view Generalised-cover Repositioning Model
SA	Simulated Annealing
SCLP	Set Covering Location Problem
SQL	Structured Query Language
SQM	Spatial Queuing Model
SSA	Singular Spectrum Analysis
SVD	Singular Value Decomposition
TEAM	Tandem Equipment Allocation Model
TIMEXCLP	Maximal Expected Coverage Location Model with time variation
VNS	Variable Neighbourhood Search
WC	Western Cape

Greek Symbols

α	Service reliability value.
α^t	Service reliability value during time period t.
β	Repositioning penalty.
η^t	Average system busy probability during time period t.
γ	Penalty for the number of relocations between time periods.
$\lambda_1, ..., \lambda_L$	Eigenvalues of $\mathbf{XX}^T$, with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L \geq 0$.

λ_i^t	Total demand rate in the neighbourhood N_i^t during time period t.
μ_i^t	Service rate of ambulances in the neighbouhood N_i^t during time period t.
ϕ_i^j	Random value ranging between -1 and 1.
ρ	Congestion or traffic intensity rate.
ρ_i^t	Congestion or traffic intensity rate in the neighbourhood N_i^t during time period t.
τ	Time horizon.
θ	Penalty for each holding site location used during at least one time period.
$\mathit{\Lambda}$	Diagonal matrix of eigenvalues of $\mathbf{XX}^T$, where $\mathbf{XX}^T = P\mathit{\Lambda}P^T$ and $\mathit{\Lambda} = \mathrm{diag}(\lambda_1, \cdots, \lambda_L)$.

Roman Symbols

$(n-1)$	Degrees of freedom.
$\bar{x}$	Arithmetic mean of all the data points of a data set.
$\tilde{\mathbf{X}}$	Approximation of $\mathbf{X}$.
$\tilde{y}_t$	Predicted value.
$M/G/s/s$	Erlang loss model.
V_i	Principal components of X_i.
$(\sqrt{\lambda_i}, P_i, V_i)$	This collection is called the i^{th} eigen-triple of the matrix X.
$\mathbf{X} = X_1, \cdots, X_K$	Trajectory matrix of the multidimensional series of L-lagged vectors to which Y_T is mapped.

Nomenclature

a_j^{bt} Binary variable indicating whether ambulance b is located at holding site node j during time period t.

$alloRelax$ Variable which is 1 if the relaxed ambulance allocation constraint is to be implemented and 0 if the original ambulance allocation constraint is to be implemented.

amb Set of available ambulances in the system.

$ambSpd$ The average ambulance speed.

b_i^t Total number of ambulances in the neighbourhood N_i^t during time period t.

bee_i Represents the Z-dimensional vector of food source i.

$bigM$ Large constant.

c^t Minimum expected coverage requiremend during time period t.

C_i^{wkt} Incremental coverage obtained by increasing the number of ambulances of type k in the neighbourhood N_i^t from $(w-1)$ to w during time period t.

C_i^{wt} Incremental coverage obtained by increasing the number of ambulances in the neighbourhood N_i^t from $(w-1)$ to w during time period t.

$c_{jj'}$ The cost of moving one unit from j to j' at each edge $(j, j') \in E$.

$ColSiz$ The colony size of the artificial bee colony.

d_i^{kt} Ambulance demand rate for ambulance of type k at demand node i during time period t.

d_i^t Ambulance demand rate at demand node i during time period t.

D	Vertex set representing the demand nodes.
d	The rank of X, where X is the maximum X_i such that $\lambda_i > 0$ and $X_i = \sqrt{\lambda_i} P_i V_i^{'}$.
dem_j	The demand at each node $j \in V$ of a minimum cost flow problem.
E	Edge set of network graph representing travel time or distance between nodes.
$E(w)$	Probability of coverage for a random call inside N_i^t with w ambulances.
f	Number of outliers that can be determined with Rosner's test when there are 25 or more data points.
fit_i	The fitness value of food source i.
G	Generally distributed service times.
$Gr = (V, E)$	Network graph with V the set of nodes and E the edge-set.
H	Vertex set representing potential holding site nodes.
$hcapacity$	Integer variable which indicates the ambulance capacity of the holding site nodes.
$hnum$	Variable used to determine the number of holding site nodes created, when number of holding site nodes is equal to $(hnum - 2)^2$.
$I = I_1, \cdots, I_m = \{i_1, \cdots, i_p\}$	Disjointed subsets of X_i.
K	An integer value equal to the number of data points minus window length plus one, i.e. $T - L + 1$.
L	Embedding parameter, called the window length, an integer value that is less than or equal to $T/2$.

$Limit$ Solution abandonment limit.

M_i^{kt} Minimum number of ambulances of type k required inside N_i^{kt} to adequately cover demand node i with an α service reliability level during time period t.

M_i^t Minimum number of ambulances required inside N_i^t to adequately cover demand node i with an α service reliability level during time period t.

M Poisson distributed call arrival rate.

m Indexing variable for the disjointed subsets of X_i.

MCN Maximum cycle number used as the stopping criteria for the ABC algorithm.

MR Modification rate value between 0 and 1.

N_i^{kt} Subset of demand nodes $z \in D$ that can be reached from holding site node j in less than r^k minutes during time period t.

N_i^t Subset of demand nodes $z \in D$ that can be reached from holding site node j in less than r minutes during time period t.

N Number of reconstructed components to use for forecasting with SSA.

n Number of nodes in the system.

$newbee_i$ Represents the mutated Z-dimensional vector of food source i.

num Variable used to determine the number of demand nodes created, when number of demand nodes is equal to $(num-2)^2$.

$NumAmbDay$	The number of available ambulance during the day shift, which is 70 for this real-world instance.
$NumAmbNight$	The number of available ambulance during the night shift, which is 55 for this real-world instance.
$NumSelect$	Integer variable limiting the number of holding site nodes that can be selected per shift.
$Obj(bee_i)$	The calculated objective function value for the solution represented by bee_i.
P_0	Likelihood of all servers being free at the same time.
p_A	Number of available ambulances.
p_A^k	Number of available ambulances of type k.
p_A^t	Number of available ambulances during time period t.
P_w	Likelihood of all servers being busy at the same time when there are w servers in the system.
p_Z	Number of holding site nodes that can be selected, i.e. number of available holding site nodes.
$P = (P_1, ..., P_L)$	Orthogonal matrix of eigenvectors of $\mathbf{XX}^T$, where $\mathbf{XX}^T = P \Lambda P^T$.
pop_i	Population associated with demand zone $v_i \in V$.
$prob_i$	The probability of food source i being selected by an onlooker bee.
$Q(p_A^t, \eta, i)$	Correction factor for Jarvis's (1985) algorithm which adjusts the probability for server cooperation in models.
q^t	Time-dependent busy probability.
q_i	Busy probability of ambulances in the system.

q_i^t	Busy probability of ambulances in the neighbourhood N_i^t during time period t.
r^k	Coverage time of an ambulance type k.
$r_{jj'}^{tk}$	Integer decision variable representing the number of ambulances of type k that should be repositioned from location j to location $j^{'}$ $(j, j^{'} \in H)$ between time period $t-1$ and t.
$r_{jj'}^{t}$	Integer decision variable representing the number of ambulances that should be repositioned from location j to location $j^{'}$ $(j, j^{'} \in H)$ between time period t and $t+1$.
r	Coverage time of an ambulance.
R_j	Uniformly distributed random real number between 0 and 1.
$rP1$	The response time target for P1 calls.
$rP2$	The response time target for P2 calls.
s_{ij}^t	Travel time between demand node i and holding site node j during time period t.
S	Coverage limit in terms of travel time.
s	Number of servers in the neighbourhood and the number of calls that can be serviced at one time.
$Scenario$	An integer that can have the value of 1 or 2.
SD	Standard deviation of the data set, calculated with $n-1$ degrees of freedom.
SN	Artificial bee colony size.
SPP	Scout production period value.

T_1 Low outlier test criterion for a single high outlier with Grubb's test.

t_{call} Average call duration.

T_n High outlier test criterion for a single high outlier with Grubb's test.

T Number of data points in time series Y_T.

$u_{jj'}$ The capacity of the flow $r_{jj'}$ on each edge.

v_i^j Represents the j^{th} mutated parameter of food source i.

V_i^t Subset of holding site nodes that can cover demand node i, i.e. the holding site nodes are less than r minutes away, during time period t.

V_j^{kt} Subset of demand nodes i that can be reached from holding site node j in less than r^k minutes during time period t.

V_j^t Subset of demand nodes i that can be reached from holding site node j in less than r minutes during time period t.

V Nodes of network graph representing the demand nodes and potential ambulance station or holding site nodes.

v Subset of V, nodes of network graph Gr.

$violation_i$ Constraint violation value of solution bee_i.

W_i^{kt} Subset of holding site nodes j that can be reached from demand node i in less than r^k minutes during time period t.

W_i^t Subset of holding site nodes j that can be reached from demand node i in less than r minutes during time period t.

x_1 Data point suspected of being a low outlier when using Grubb's test.

x_i^{wt} Binary decision variable that is equal to 1 if and only if demand node i is covered by at least w ambulances during time period t.

x_n Data point suspected of being a high outlier when using Grubb's test.

y_j^t Integer decision variable representing the number of ambulances located at holding site node j during time period t.

Y_T Real-valued non-zero one-dimensional time series with T data points.

y_t Data point t of Y_T, i.e. observed data.

z_j Binary decision variable equal to 1 if and only if a holding site node at node j is selected for use during time period t.

Z Represents the size of the Z-dimensional vector, i.e. the food source.

zp_i^j Represents the j^{th} parameter of food source i.

zp_{max}^j The maximum bound of parameter zp_i^j of food source i.

zp_{min}^j The minimum bound of parameter zp_i^j of food source i.

Subscripts

g Subscript g refers to index of data points.

h Subscript h refers to index of data points.

i Subscript i refers to index of demand nodes.

j Subscript j refers to index of holding site nodes.

k Subscript k refers to index of SVD coefficients.

m Subscript m refers to index of disjointed subsets I.

p Subscript p refers to index of subset of I.

T Subscript T refers to index of data points.

t Subscript t refers to index of time periods of the planning horizon.

z Subscript z refers to index of demand nodes.

Terminology

Ambulance coverage A call is classified as covered if it can be served within the set response time standard, meaning that an idle ambulance is close enough to the incident.

Bipartite graph A set of graph vertices separated into two unconnected sets such that no two graph vertices within the same set are adjacent.

Deterministic This states that the behaviour of an algorithm, model, procedure, or process is entirely determined by its initial state and inputs; i.e. it is not random.

Discrete data Data that can only be specific values and can be counted.

Drainage area An area around a hospital.

Dynamic model A model based on data that changes over time.

Eigenvalue Each of a set of values of a parameter for which a differential equation has a non-zero solution (an eigenfunction) under given conditions (Oxford University Press, 2016).

Eigenvector A vector which when operated on by a given operator gives a scalar multiple of itself (Oxford University Press, 2016).

Orthogonal matrix	A square matrix which is invertible, unitary, and normal. The product of an orthogonal matrix and its transpose returns an identity matrix.
Parametric model	Models that require restrictive distributional and structural assumptions, such as the assumption that the data is static (Vile *et al.*, 2012).
Periodicity	Refers to a tendency for something to happen repeatedly at certain intervals (Oxford University Press, 2016).
Probabilistic	A model that is based on a theory of probability (Oxford University Press, 2016).
Response time	The time between the taking of the call and the arrival of an ambulance (Fitzsimmons, 1973; Poulton & Roussos, 2013).
Spatio-temporal dynamics	The data changes in terms of changes in time and space (Oxford University Press, 2016).
Static model	A model based on stationary data.
Stochastic	This states that the behaviour of an algorithm, model, procedure, or process is entirely random.
Univariate data	Data which consists of one data type.
Window length	It is an embedding parameter used for the SSA algorithm; it is an integer that has to be less than or equal to $T/2$ (Golyandina *et al.*, 2001a; Hassani, 2010; Vile *et al.*, 2012).

Superscripts

$'$	Superscript $'$ refers the variable's altered state.
b	Superscript b refers to the index of the available ambulances.

k	Superscript k refers to the index of the ambulance type.
T	Superscript T refers to the variable's transposed state or the index of the last time period of the planning horizon.
t	Supersript t refers to the index of time periods of the planning horizon.
w	Superscript w refers to the index of the number of ambulances.

Chapter 1

Introduction

In Chapter 1 background information is provided to introduce the focus of the project. Thereafter, the problem statement and objectives are introduced and the research approach and structure of the report described. The need for the project and the process followed to solve the problem are also explained.

1.1 Background

Two studies concerning Inter-facility Transport (IFT) in South Africa, with a specific focus on maternal emergencies, brought the real-world problem faced by South Africa's ambulance system into focus. Summaries of the two studies are included in Appendix A. The studies evaluate the impact of guidelines and policies on decreasing response time and loss of life. The findings are significant to any professional working in the management, design, and operation of healthcare services. It demonstrates that decreasing emergency ambulance response through the implementation of simple operational rules has a significant, positive impact on the maternal mortality rate, which is an important healthcare indicator that South Africa struggles to improve.

1.1.1 Ambulance service

In the Emergency Medical Services (EMS) component of the healthcare system, as in any system, there exists critical paths, which cause damage to the whole system if lengthened. Knight *et al.* (2012) stated that the ambulance service is just such a critical path of the healthcare system. The ambulance service consists of a specific chain of events that lead to the arrival of the ambulance and intervention by the on-board personnel (Bélanger *et al.*, 2015; Brotcorne *et al.*, 2003), which may be described as follows:

- Step 1: the incident detection, recognition of need for emergency assistance, and call placement by the public;
- Step 2: the screening of the call at the Emergency Control Centre (ECC) by call-takers;
- Step 3: the dispatching of an ambulance from its holding site by dispatchers at the ECC;
- Step 4: the arrival of the ambulance and the intervention by the on-board personnel; and
- Step 5: patient transportation and responsibility transfer to a healthcare facility.

Step 1 can only start once an emergency is recognised and a call has been placed by a bystander (Andersson & Värbrand, 2006). The ECC therefore has no control over Step 1. The managers, call-takers, and dispatchers at the ECC have some control over Step 2 and Step 3.

Step 2 starts when the call is answered by a call-taker. The call is then screened to determine the severity and degree of urgency of the emergency, and prioritised according to a prioritisation rule (Andersson & Värbrand, 2006). Step 3 starts when the call-takers assign prioritised calls to dispatchers in charge of the area the call originates from. It ends when an available ambulance is dispatched to the call site according to a dispatching rule that determines which of the available ambulances should be dispatched (Kergosien *et al.*, 2011).

Only the ambulance's on-board personnel have some control over Step 4 and Step 5. Step 4 is the arrival of the ambulance at the scene, the assessment of the emergency, and the intervention by the on-board personnel. Step 5 is the transportation of the patient to a relevant healthcare facility, if necessary (Kergosien *et al.*, 2011). The ambulance becomes idle and available for dispatch as soon as care of the patient is taken over by the healthcare facility personnel (Kergosien *et al.*, 2011).

1.1.2 Ambulance service efficiency

Ambulance efficiency is determined by implementing statistical analysis on historical call data. The efficiency of an ambulance is generally indicated by the average response time (Schmid, 2012) – the time between the taking of the call and the arrival of the ambulance (Fitzsimmons, 1973; Poulton & Roussos, 2013), i.e. Step 2 and Step 4, as indicated in Section 1.1.1 – and ambulance coverage. A call is classified as covered if it can be served within the set response time standard, meaning that an idle ambulance is close enough to the incident to reach it within the specified response time standard. Fleet size, holding site location, and ambulance allocation are critical factors that ECC managers can control in anticipation of demand in order to improve ambulance efficiency (Lim *et al.*, 2011). These factors are generally planned and based on static historical or predicted ambulance demand data.

1.2 Problem statement and objectives

The problem statement, objectives, importance, limitations, assumptions, and ethical implications of the research problem, being considered in the project, are described in the following sections.

1.2.1 Problem statement

Since the end of 2014 the Western Cape (WC) ECC has used a Computer-aided Dispatch (CAD) system, CareMonX, to streamline the dispatching of ambulances to calls. The system uses algorithms to determine the closest ambulance to a logged call in terms of travel time. A WC ECC dispatcher then dispatches the ambulance closest to the call, based on the information from CareMonX, which is their dispatching rule.

At the start of every shift the dispatcher determines the location of a number of holding sites and the number of ambulances to place at each site. The premise is that the placement of ambulances at specific sites should increase ambulance coverage. During a shift, ambulances are also relocated to other holding sites to keep the coverage level constant and increase it if possible. Decisions regarding holding site placement, ambulance allocation, and relocation are currently based solely on the dispatcher's knowledge, experience, and intuition. The WC ECC does not forecast ambulance demand and has no policy in place that the dispatchers can follow and base their decision on.

The development of a Decision Support Tool (DST) with an ambulance location model to plan (i) holding site placement per shift; and (ii) ambulance allocation and relocation per hour, based on predicted ambulance demand data, would provide decision support to dispatchers. The DST's output could help increase coverage and decrease average ambulance response time through the near-optimal placement of holding sites and the allocation and relocation of ambulances. The DST's output is to be merely a guideline to inform the decision-making process of dispatchers and not to make the decision for them.

1.2.1.1 Aim

The research project is undertaken to create a concept demonstrator DST which can be utilised to produce a solution for the near-optimal (i) holding site placement per shift, and (ii) ambulance allocation and relocation per hour of that shift, for ambulance demand from the City of Cape Town and the Cape Winelands municipalities. The DST will contain an ambulance location model which will be the mathematical representation of the problem. The ambulance location model will require call rate prediction as an input. A forecasting method will be used, and coded into the DST, to predict the probable future call rates and call classification rates based on historical call data. An algorithm or heuristic will then be chosen and coded to determine the near-optimal holding site locations per shift, and ambulance allocations and relocations per hour of that shift. The DST is to be used as a planning tool to inform the dispatcher's decision-making process. The DST will be tailor-made to be used by the WC ECC for application for the City of Cape Town and the Cape Winelands municipalities, i.e. the real-world instance. Once the DST has been created, it will be validated and verified.

The primary goal of the project is to develop a concept demonstrator DST for the real-world instance, provided by the WC ECC, and to prove the need for the DST. The DST will be developed to be used in conjunction with the CareMonX system. Integrating the concept demonstrator DST into the WC ECC's system is not part of the project's scope. A secondary goal is to learn from the development of the DST for the WC ECC's real-world instance and to make suggestions for adapting the DST to be used in other ECCs.

1.2.1.2 Project objectives

The project has to fulfil certain primary objectives in order to achieve its aim. These primary objectives are to:

1. gather, analyse, and study the historical call data for the real-world instance;
2. choose an ambulance demand forecasting method;
3. choose an ambulance location problem model on which to base the model for the real-world instance;
4. choose a solution method to solve the model in order to produce a near-optimal solution for the (i) holding site placement per shift, and (ii) ambulance allocation and relocation per hour of that shift;
5. choose software to program the DST concept demonstrator in;
6. learn the programming language and code the DST concept demonstrator;
7. run the DST concept demonstrator for the real-world instance; and
8. validate and verify the DST concept demonstrator during the development and testing processes.

1.2.2 Importance of the problem

The results of this research will be important to the WC ECC, provincial health departments, and the national Department of Health (DoH). The outcome of this research can assist in improving response times in South Africa through improved holding site placement and ambulance allocation and relocation, i.e. through improving the efficiency of resource utilisation.

1.2.3 Limitations and assumptions of the study

A limitation of this study is that the performance of the DST will only be as good as the predicted demand rates that the ambulance location model uses as input. Therefore, the quality of the historical call data, the forecasting method, and the model all play a role.

The ambulance location problem model that will be used in the DST cannot be a dynamic model, since integrating the DST into the WC ECC's system is not part of the project's scope, as it is only meant to be a concept demonstrator. Therefore, the DST can only be coded to use historical data as input and not dynamic data, i.e. data in real-time.

1.2.4 Ethical implications of the research

The data that will be used in this study is to be collected from the WC ECC at Tygerberg Hospital in Cape Town. The project is to be undertaken in association with the WC ECC. The data does not contain personal information of any individual, and therefore the use of the data does not have ethical implications. However, the use of the DST to determine near-optimal holding site locations and the requisite allocation and relocation of ambulances may have ethical implications. If implemented, the DST may have a positive or negative effect on patient survival. The likelihood of positive results will be increased by: informing users that the model is not dynamic; choosing an accepted mathematical model for the real-world problem; applying a forecasting method that has previously been used in this context; and solving the problem by making use of solution algorithms or heuristics that have worked in similar circumstances.

1.3 Proposed research approach and strategy

In this section information on the research design and the procedure that will be followed to complete the proposed research problem, as stated in Section 1.2, is provided.

1.3.1 Research design

The problem statement considers an empirical question. The question concerns a real-world problem and the plan is to solve it by analysing historical data (Mouton, 2013). The type of historical data that will be used is numerical data aggregated from data collected by the WC ECC. Thus, the design type of this project can be identified as statistical modelling and computer simulation studies. This type of design focusses on the development and validation of accurate models for real-world circumstances (Mouton, 2013).

1.3.2 Research methodology

The method that will be used to gain the knowledge required for this project is a literature study. A number of topics will be researched:

1. ambulance efficiency;
2. operational research (OR) in the EMS context;
3. overview of DSTs;
4. ambulance demand forecasting methods;
5. overview of ambulance location problem models; and
6. ambulance location problem solution methods.

1.4 Structure of the report

In Chapter 2, significance of ambulance efficiency for trauma incidence and the role OR has played, and still plays, in its management are described. Chapter 3 contains summaries on the concepts that were researched in order to provide the knowledge required for the creation of the DST for the WC ECC's real-world instance. The real-world instance, upon which the project is based, as well as the choice of model and solution methods are explained in Chapter 4. The data gathering, data analysis, and data flow through the DST are explained in Chapter 5. Chapter 6 contains descriptions of the two scenarios, their corresponding DST solutions, and the verification and validation of the solutions against historical call data. The project summary, research findings, research contributions, and opportunities for further work are provided in Chapter 7.

1.5 Conclusion: Introduction

This chapter provided explanations on the origin of the project and background information on ambulance service. The problem statement and objectives were explained, and the research approach, strategy, and report structure was provided. Chapter 2 contains information on ambulance efficiency, and its significance, and the role OR has played, and still plays, in the management of EMS.

Chapter 2

Contextualisation: The management of emergency medical services

The research project was introduced in Chapter 1. In Section 1.1.1 an overview on the ambulance service was provided, Section 1.2 contained the problem statement, the project aim, and its objectives, while in Section 1.3 the research design and methodology were described. This chapter contains an introduction to the necessity of ambulance efficiency. Thereafter, the role that OR has played and still plays in the management of EMS is explained.

2.1 The significance of ambulance efficiency for high severity incidents

The probability of death or lasting disability after incidents of high severity (or trauma) can be described as a function of time until treatment, often indicated by response time (Fitzsimmons, 1973; Lee, 2012; Poulton & Roussos, 2013). High severity incidents include most cardiovascular incidents, head injuries, car crashes, obstetric emergencies, and other life-threatening injuries.

The treatment of some types of cardiovascular diseases, such as stroke and myocardial infarction, are time dependent (Cantwell *et al.*, 2015; The National Institute of Neurological Disorders and Stroke rt-PA Stroke Study Group, 1995), so ambulance efficiency, i.e. quick intervention, is crucial to minimise mortality. The International Guidelines 2000 Conference on Cardiopulmonary Resuscitation and Emergency Cardiovascular Care recommend that ambulances aim to achieve a response time of 8 to 10 minutes from the time of collapse. This would provide the maximum potential for successful cardiac and cerebral resuscitation. Also, the use of a defibrillator within 5 minutes would be preferable (Blackwell & Kaufman, 2002).

According to Trunkey (1983) head-injury patients require surgical intervention within four hours of injury, while patients who suffered severe haemorrhage require surgical intervention within 20 minutes. Hoffman (1976) stated that the availability of and quick access to medical care are early factors that affect the probability of surviving a car crash. Therefore, the length of the response time affects the survival probability of patients who suffered trauma.

As shown, the link between high severity incidents and shorter response times, has been used to state that the reduction of response time would improve patient survival in many categories of illness and injury, not only for trauma incidents. This is known as the "golden hour" philosophy. Lerner & Moscati (2001) attempted to determine the validity of the philosophy and to determine the term's origin. While they were able to determine that it was most likely originated by one of the fathers of trauma surgery and design, R.A. Crowley; they could not prove that it was based on explicit research. However, no large-scale, well-controlled study based on a civilian populations has been done that strongly supports or refutes the philosophy that shorter response time is required for all types of emergency care situations. Therefore, the general and intuitive philosophy still stands and is still used to determine ambulance efficiency. The "golden hour" philosophy is accepted and utilised by most ECCs, including the WC ECC.

A single type of high severity incident, obstetric emergencies, will be described as an example to substantiate the need for more efficient ambulances.

2.2 Obstetric emergencies

In this section the example of obstetric emergencies, and the utilisation of the "golden hour" rule, will be used to show the need for better response times in general in South Africa. Obstetric emergencies were chosen since the lowering of obstetric emergency mortality due to long response time is of critical importance to the South African DoH.

2.2.1 Maternal mortality

The obstetric risk factor used in literature is the Maternal Mortality Rate (MMR). MMR is the number of maternal deaths per 100,000 live births. The World Health Organization (2013) defines maternal death as the death of a pregnant woman or of a woman within 42 days of terminating the pregnancy, due to any cause related to, or aggravated by, the pregnancy or the management of the pregnancy. A live birth is the complete extraction of a baby from its mother; the baby must be breathing or showing signs of life, such as a beating heart, pulsating umbilical cord or movement of voluntary muscles, regardless of whether the umbilical cord has been cut or the placenta is attached. The duration of the pregnancy and where the foetus is located in the womb are not taken into account in either definition (World Health Organization, 2013).

In the eighth report on perinatal care in South Africa, Pattinson (2013) stated that there are five major categories of obstetric causes of death: sudden unexplained intrauterine deaths; spontaneous pre-term birth; intrapartum asphyxia and birth trauma; complications of hypertension; and antepartum haemorrhage[1] (Pattinson, 2013).

It is generally assumed that the obstetric risk factor (thus, the MMR) due to the aforementioned causes is higher in areas of lower socio-economic conditions. However, it has been shown that these types of complications can occur during pregnancy and labour even in the best of socio-economic conditions, at the best hospitals. Therefore, in recent years, focus has been directed away from primary prevention of maternal deaths, i.e. merely improving socio-economic conditions, to secondary prevention, i.e. to prevent death once the complication has occurred (Pattinson, 2013).

Secondary prevention considers factors that affect the time interval between the onset of obstetric complications and their outcome. The outcome tends to be highly affected by a delay in treatment, i.e. the length of ambulance response time (Thaddeus & Maine, 1994), meaning that the outcome tends to be satisfactory if prompt, adequate treatment is provided. This factor is the second delay of the three delays model created

[1] Sudden unexplained intrauterine death: the ceasing of foetal movement, absence of foetal heart beat (by stethoscope or sonic-aid), absence of foetal cardiac activity (real-time ultrasound); Spontaneous pre-term birth: babies that are born alive before the end of 37 weeks of pregnancy (World Health Organization, 2014); Intrapartum asphyxia and birth trauma: brain injury caused by oxygen deprivation (CerebPalsy.org, 2015); Hypertension: high blood pressure (Nordqvist, 2012); Antepartum haemorrhage: genital bleeding after week 28 of pregnancy and before the end of the second stage of labour (EI-Mowafi, 2008).

by Thaddeus & Maine (1994): the delay of the decision to seek care, the delay of the arrival at a healthcare facility, and the delay of the provision of adequate care.

These three factors are used to understand the gaps in access to adequate management of emergencies. This model also attempts to show that, even if a patient does choose to seek care in a timely fashion, the patient can still experience delay due to inaccessibility of healthcare services, such as transport, and the lack of provision of trained personnel (Thaddeus & Maine, 1994). This tends to be a bigger problem in the developing world – specifically in rural areas – as a patient might find the closest healthcare facility to be only equipped for basic treatments and education, and there may be no way for the patient to reach a facility that does have the required resources (Thaddeus & Maine, 1994). According to Pattinson (2013), the main purpose of the model's development was to create a mechanism for identifying where the required and appropriate interventions could be made.

2.2.2 Importance of the maternal mortality rate

The leading cause of death and disability among women of reproductive age in developing countries, such as South Africa, is complications during or after pregnancy, child birth, and/or termination of pregnancy (World Health Organization, 2013). The MMR is therefore deemed important, as it represents obstetric risk – the risk associated with each pregnancy – in the country.

It is not surprising that South Africa's MMR is of concern to government. This concern is evident in the fact that the rate was specifically published in the population policy and the National Service Delivery Agreement of 2010-2014 (South African Government, 2013). The National Development Plan 2030 (South African Government, 2015) also indicates the reduction of the MMR as one of their objectives for improving the health of South Africans (Statistics South Africa, 2013). The focus on MMR is not limited to South Africa; it was also chosen as an indicator for the Millennium Development Goals (MDG), demonstrating that it is an important international indicator of the quality of healthcare provision (World Health Organization, 2013).

Between 1994 and 2010, the DoH put a number of policy initiatives in place to reduce MMR and to improve the general quality of healthcare. South Africa is also a signatory to a number of treatises and conventions, including the Millennium Declaration, which includes the MDG, which promote the improvement of maternal health (Statistics South Africa, 2013).

2.2.3 Millennium declaration

The United Nations Millennium Summit took place between 6 and 8 September 2000. This summit brought together 149 heads of State and government, as well as high-ranking officials from approximately 40 countries. During the summit, a document known as the Millennium Declaration was signed by the 189 countries present. The Declaration consists of 8 MDG and their targets, with a deadline of 2015. The targets and goals are inter-related and represent a partnership between developed and developing countries. By signing the Declaration, the leaders committed their countries to reducing extreme poverty on a national and global level (United Nations, 2000).

As one of the signatories, South Africa has committed to: (1) eradicate extreme poverty and hunger; (2) achieve universal primary education; (3) promote gender equality and empower women; (4) reduce child mortality; (5) improve maternal health; (6) combat HIV/AIDS, malaria and other diseases; (7) ensure environmental sustainability; and (8) develop a global partnership for development (United Nations, 2000). The fifth MDG is the improvement of maternal health. The MMR target set for South Africa to reach by 2015 was 38 maternal deaths per 100,000 live births (Statistics South Africa, 2013, 2015). South Africa's final MDG country report was released in 2015, and the last documented MMR, calculated in 2013, was 141 maternal deaths per 100,000 live births (Statistics South Africa, 2015). South Africa, therefore, did not succeed in reaching the fifth MDG target.

2.3 Operational research in emergency medical services

The logistics of World War II is seen as the birth of OR, and this is also part of the reason for its rich history in the context of emergency response services. Today emergency response services consist of fire, police, and EMS (Simpson & Hancock, 2009).

OR models and methods are used to improve the performance of organised environments, but emergencies are by definition the disruption of organised environments. The evolution of EMS and the advancement of OR relating to emergency response have led to changes in the literature published on OR since 1970. (Simpson & Hancock, 2009). Due to the ever-increasing pressure on the healthcare system to provide efficient service, a major application of OR in the emergency response services lies in healthcare. Coping with the growing demand for healthcare and the volatile nature of the number of arrivals at a healthcare facility makes modelling the resource usage for healthcare one of the most challenging fields of OR (Gillard & Knight, 2014).

A variety of traditional OR methods and techniques have been implemented to ensure the efficiency of healthcare systems. These include the optimisation of surgery schedules (Cardoen *et al.*, 2010), the optimal location of healthcare clinics and facilities (Smith *et al.*, 2009), and the modelling of capacity requirements in critical care (Griffiths, 2005). Unscheduled care is one of the many aspects of healthcare that makes it an un-organised environment because a hospital has little control over it (Gillard & Knight, 2014). According to Simpson & Hancock (2009), the most used OR approach to solve the difficulties of emergency response is mathematical programming, then probability and statistics, and lastly simulation.

The public expects efficient service from any emergency response organisation, especially EMS. In order to provide efficient service a number of decisions regarding the employed strategies have to be made. The number of resources to use, the management of the fleet, and the location and relocation of resources all need to be discussed and decided on. These decisions can be classified in three classic decision-making levels, namely strategic, tactical, or operational (Bélanger *et al.*, 2015; Simpson & Hancock, 2009).

Strategic decisions are concerned with the location of ambulance depots and the size of the ambulance fleet. Tactical decisions are concerned with the location of potential stations or holding sites and what areas it should cover, the crew pairing, scheduling, and management of the fleet. Finally, operational decisions, that commonly need to be considered in real-time, involve short term decisions, such as ambulance dispatching, i.e. which ambulance should be sent to a call, and relocation (Aartun & Leknes, 2014; Bélanger *et al.*, 2015).

Initially, Emergency-related OR (EOR) work was focussed on strategic and tactical decisions, i.e. the static location problem. This is seen in practice in Brotcorne *et al.*'s (2003) review, which focussed on static location problem models. The static location problem seeks to determine the set of holding sites where ambulances should be positioned between missions and the number of ambulances that should be placed at each. Once solved and implemented, the location plan remains unchanged, so each ambulance returns to its designated holding site after each mission. However, it can be beneficial to change ambulances' locations during the day to account for the evolution of the situation faced by the ambulance service or EMS. According to Bélanger *et al.* (2015), a significant effort has been made in recent years to develop approaches that explicitly consider the uncertainty and dynamism that form part of the EMS context; this has led to a considerable number of new models and management strategies. The new models, specifically those devoted to vehicle relocation, have mainly been developed to deal with situations where ambulances can be deployed to a large set of locations over the service area, which includes hospitals or medical facilities. In this context, the designated holding site of an ambulance can be changed during the course of a day without necessitating significant cost or modifications to the system itself (Bélanger *et al.*, 2015).

2.4 Ambulance location model: Operational rules

As stated in Section 1.1.2 fleet size, holding site placement, and ambulance allocation are critical factors that ECC managers can control to improve the efficiency of the ambulance service. The required fleet size, holding site placement, and ambulance allocation are generally determined with the help of ambulance location models, along with other aspects if required. Ambulance location problem models are explained in more detail in Section 3.6. To create an ambulance location problem model, information concerning the operational rules for the real-life problem considered is required. The most important operational rules are the service quality criteria, dispatch policy, and relocation policy.

2.4.1 Service quality criteria

An ambulance location problem model contains an objective function that indicates what aspect(s) of the problem the model seeks to improve. For any ambulance location problem the two most important aspects will be the ambulance coverage and response time (Schmid, 2012). The two also have an effect on each other, as good coverage implies that there are enough ambulances positioned in an area to react to requests in that area; and, the fact that they are already positioned in the area means that the response time target can be met and can even be lower than the target. Generally, the objective function seeks to (i) minimise the average response time and keep the area coverage above an acceptable level; or (ii) increase the expected coverage and minimise the required relocations. The response time standard depends on the country, and the coverage standard is a time or distance standard based on the response time standard.

Different countries have different response time standards and most also have service reliability standards. In Montreal, Canada, the standard is that 90% of calls should be served within 7 minutes. In the United States, 95% of calls in urban areas should be served within 10 minutes, whereas in rural areas it should be served within 30 minutes. The United Kingdom has different response times for different categories of calls; 75% of category-A calls should be served within 8 minutes, and 95% of category-B and -C calls should be served within 14 minutes in rural areas, and 19 minutes in urban areas, respectively (Lim *et al.*, 2011). In Germany the response time standard differs from one area to another; it generally ranges between 10 to 15 minutes. The service reliability value is generally that 95% of calls should be reached within a standard response time (Erkut *et al.*, 2008). In Wales the response time standard is to reach 65% of category-A calls within 8 minutes, and to reach 95% of category-B and -C calls within 14 minutes in urban, 18 minutes in rural, and 21 minutes in sparsely populated areas (Vile *et al.*, 2012). The response time standard in the Netherlands is that 95% of calls should be reached within 15 minutes (van den Berg *et al.*, 2015).

2.4.2 Dispatch policy

Ambulance dispatch is defined by Lim *et al.* (2011) as the process of assigning an ambulance to answer an emergency call. The dispatch policy often also contains the method of call prioritisation or queueing and the process of assigning an ambulance to an emergency call in the queue (Lim *et al.*, 2011).

Call prioritisation is the process of prioritising calls in a queue. However, according to Lim *et al.* (2011) call prioritisation is not appropriate for developing countries with resource-limited EMS, as additional resources are needed for implementation. The WC is one of the few provinces that has enough resources to implement call prioritisation, as well as a CAD system that streamlines the dispatching process.

There are a number of ways in which calls can be prioritised. Calls can be prioritised based on:

- centrality, most central call first (Lee, 2012);
- a first-in first-out (FIFO) basis (Schmid, 2012; Thakore *et al.*, 2002); or
- incident type (Poulton & Roussos, 2013).

Calls that are prioritised based only on centrality would cause ambulances to be excessively repositioned without fulfilling nearby requests (Lee, 2012). FIFO prioritisation causes life-threatening calls to wait if a non-life-threatening call was first and only one ambulance is available, purely because another call was first (Schmid, 2012; Thakore *et al.*, 2002). Incident type prioritisation is difficult, as generally call-takers have only basic medical training and the caller has none (Yancey & Mould-Millman, 2015). Also, in most ECCs no protocol exists to help the call-takers determine incident type. Even though incident type prioritisation is difficult and usually impossible to do accurately it is still the method most often used.

The dispatch policy is what the dispatcher uses to determine which ambulance to dispatch to the prioritised call. A general rule for any dispatch policy is that an ambulance can only be assigned to one call (Schmid, 2012). The following are some dispatch policies that are implemented for assigning ambulances to calls:

- closest dispatch: dispatch the idle ambulance that is the closest to the incident location or that can get there the fastest (Lim *et al.*, 2011; Poulton & Roussos, 2013);
- non-closest dispatch: determine what ambulance to assign by combining coverage with probability (Lim *et al.*, 2011; Poulton & Roussos, 2013);
- reroute enabled dispatch: used along with another policy to improve response time of urgent calls (Lim *et al.*, 2011);
- pseudo priority dispatch: used along with another policy to upgrade lower priority calls with long waiting times (Lim *et al.*, 2011);
- priority update enabled dispatch: used along with another policy to be able to update call priority (Lim *et al.*, 2011); and

- pre-arrival instructions: used along with another policy to provide ambulance crew with instructions (Lim *et al.*, 2011).

Appendix C contains tables that show the advantages and disadvantages of the dispatch policies mentioned, along with some that were not mentioned. According to Lim *et al.* (2011) and Poulton & Roussos (2013), the most commonly used dispatch policy is closest dispatch with incident call prioritisation. That is also what WC ECC uses.

Euclidean distance, not road route distance, is the most used distance measure employed by ECC dispatchers to determine the nearest ambulance (Poulton & Roussos, 2013). The reason is that it is the easiest measure to implement and to replicate, allowing for fast computation and reaction. However, this method is inaccurate. It is possible to use a geographic information system (GIS) to calculate the road distances. The GIS environment has improved and become more and more accessible, but a costly software download and license is still required to make use of it. The cost and accuracy has to be weighed and balanced to determine which is more important. The project could not afford such a license and will use the Haversine distance method. It is similar to the Euclidean method, but the Haversine method does not assume that the earth is flat and will therefore be more accurate over longer distances.

2.4.3 Relocation policy

A relocation action can be triggered by an area's coverage falling below the minimum (Andersson & Värbrand, 2006) or by an ambulance that becomes idle (Schmid, 2012). The first trigger is not used often because ECCs generally have a regulation that does not allow ambulances to be relocated directly between holding sites (Schmid, 2012); however, it is allowed at the WC ECC. This regulation is usually in place to ensure that ambulances are not driving around empty. The second trigger is used more often. However, the first trigger would allow for a better average coverage over the planning horizon than the second trigger.

The purpose of the relocation policy is to keep the ambulance coverage of the area high (Andersson & Värbrand, 2006). Looking at it from a global perspective, if all available ambulances are located specifically to cover future demands more effectively, i.e. the relocation policy is implemented in an anticipatory manner whereby the current situation and potential future requests are taken into account, the performance of emergency services can be improved (Schmid, 2012).

2.5 Conclusion: Contextualisation: The management of emergency medical services

In Chapter 2 the necessity of ambulance efficiency and the role that OR has played and still plays in the management of EMS were explained. Chapter 3 contains summaries of the research done on DSTs, ambulance demand forecasting, resource deployment, and solution methods.

Chapter 3

Demand forecasting and resource deployment methods

The need for an efficient ambulance service was explained, and the use of OR in the management of EMS systems was described in Chapter 2. This chapter contains research on DSTs, demand forecasting, and resource deployment methods and their solution methods, all in the context of the EMS system.

3.1 Overview: Decision support tools

In the 1960s organisations started to computerise most of their operational processes, i.e. processing, billing, inventory control, and accounts payable. Information systems (IS) were developed to assist this process. Management IS (MIS) was the first program type of its kind; its purpose was to make information in transaction processing systems available to management for decision-making purposes, but in reality most failed. According to Pervan & Arnott (2005) the failures were due to the Information Technology (IT) department not understanding the nature of the work that the managers wanted to base on the output of MIS, which lead to the creation of large and inflexible systems that produced reports that were typically dozens of pages thick and contained little information of worth.

DST is an area of IS that was created to be an environment that would enable decision makers and IT to work together to solve problems, and initially a lot of the work was experimental. The idea was that the decision-maker would deal with the complex unstructured parts of the problem and the computer system would provide assistance by automating the structured parts. In general, the problems that require the use of a DST are impossible or inappropriate to solve purely using a computer system (Pervan & Arnott, 2005). DSTs have grown radically since its experimental start and have become part of everyday business practices. Most are used to facilitate and improve the effectiveness and efficiency of management, planning, and/or staff activities (Alter, 1977; Dahl & Derigs, 2011).

Today's DSTs can be described as interactive computer-based systems that support decision makers in solving semi-structured problems. Generally, a DST consists of three parts: the database, the analytical tool/model, and the user-interface (Dahl & Derigs, 2011; Rönqvist, 2012). The way the three parts are implemented and used differ for the different types. The DST discipline is still growing, and new types are still being created. The following is a short list of some of the types of DSTs that are used in business:

- the personal DST (PDST) is a small-scale system used by one manager, or a small number of managers, to support a decision-making process (Arnott & Pervan, 2008; Pervan & Arnott, 2005);
- the group support system (GSS) combines communication and DST technologies; it is used to facilitate effective group work (Arnott & Pervan, 2008; Pervan & Arnott, 2005);
- the negotiation support system (NSS) is used by a group when the work focusses on the negotiations between opposing parties (Arnott & Pervan, 2008);
- the intelligent DST (IDST) uses the application of artificial intelligence techniques to support decision-making processes (Arnott & Pervan, 2008);
- the knowledge management-based DST (KMDST) aids in knowledge storage, retrieval, transfer, and application. It is used to support individual and organisational decision-making (Arnott & Pervan, 2008);
- data warehousing (DW) provides large-scale data infrastructure often required for decision-making (Arnott & Pervan, 2008; Pervan & Arnott, 2005); and

- the enterprise reporting and analysis system is an enterprise focussed DST, which includes executive IS (EIS), business intelligence (BI), and corporate performance management systems (CPM) (Arnott & Pervan, 2008).

Many DST types make use of OR techniques in the analytical tool/model. In order to implement these techniques Rönqvist (2012) proposes that a specific process should be followed. The process is usually followed by an analyst and should be repeated even after the DST is finished in order to revise and develop the DST, keeping it relevant and useful. The steps of the process are:

1. describe the problem and any necessary assumptions and/or simplifications (Sections 4.1 and 4.2);
2. translate the simplified problem into a suitable model (Section 4.2);
3. choose and develop a method which solves the model within the time limits and with sufficient solution quality (Section 4.3 and Chapter 5); and
4. evaluate the model and method by running tests and analysing the results (Chapter 6).

The first step is based on existing written material or verbal communication with the decision maker who is to use the DST or is responsible for it. The description of the problem then allows the analyst to articulate the problem's objectives, to identify the components and classify them as relevant or irrelevant, and to determine the structure of the problem. This step also helps the analyst to determine whether it is appropriate to formulate an optimisation model or if there are alternative methods. An optimisation model can only be used if the relevant aspects of the problem are quantifiable (Rönqvist, 2012). Ideally, the problem can be defined as structured, i.e. all the problem's dimensions are known, or a semi-structured problem which can be made to be structured with help from the decision maker. The models for semi-structured problems can be unsatisfactory if the analyst has to determine most of the dimensions of the model without the decision maker in order to transform it into a structured problem (Densham, 1991).

The second step requires that it be determined how the proposed DST is to interact with other systems currently in use, the solution time available, the required solution quality, and the collection or generation plan for the data. The simplified problem is then created based solely on the problem description. The optimisation model can then be mathematically based on the simplified problem. The optimisation model consists of the decision variables, objective function, and constraints. The decision variables reflect the decisions that are possible, and the objective function represents the goals that the solution need to meet and balance, while the constraints aim to create solutions that are feasible (Rönqvist, 2012).

The third step starts with the collection (or generation) and cleaning of the data to ensure that it does not contain erroneous data points (Rönqvist, 2012). The data analysis process is usually implemented to fulfil this need. An added functionality of the DST would be to include routines to filter any data before it is added to the system, otherwise it the data has to be filtered by the decision maker. The solution method chosen during this step depends on the complexity of the model, the solution time, and quality requirements – all of which are determined during the previous step. During the fourth step the model is tested and the solution produced is evaluated and transferred into a form that the decision maker can use in the decision-making process (Rönqvist, 2012).

3.2 Importance of forecasting ambulance demand

Many studies have sought to create and/or use models to improve the efficiency of the EMS, i.e. ambulance service. A majority of these models make use of OR to improve their ambulance deployment with the hope of minimising response time and increasing demand coverage. These models all differ in terms of complexity, but in order to be effective they all require accurate ambulance demand data. Depending on the type of the model, the data can be historical, predicted, or dynamic. Models that use historical call data look to the past to plan for the future which can be extremely inaccurate, as the future is not identical to the past. The ideal would be to create and use models that can accommodate dynamic data and that have to be solved as the ambulance demand data changes in real-time in order to make immediate decisions, but these models are computationally intensive and not always feasible. The best option after dynamic data is to forecast ambulance demand based on the trends seen in the historical data, which will allow a model to plan resource deployment for the probable future. According to

Vile *et al.* (2012), less study has been invested in this important field even though the ability to predict ambulance demand accurately at short time intervals and in terms of their location is critical for the management and near-dynamic deployment of the ambulance fleet.

3.3 Overview: Ambulance demand forecasting

According to Vile *et al.* (2012), ambulance demand forecasting models are comparable to those designed for and used by fire and police services. The models initially created and used were extremely simplistic and had many shortcomings, as they did not account for daily or weekly trend data or other causal factors. These early models were also parametric, requiring restrictive distributional and structural assumptions, such as that the data is stationary. These models did prove to be useful for capacity planning and budgeting, but recent advances in location analysis – which allow for more flexible and dynamic ambulance deployment strategies – require more responsive demand predictions and model-free methods to forecast ambulance demand volumes (Vile *et al.*, 2012).

In Section 3.3.1, ambulance forecasting models are discussed, and in Section 3.3.2 a forecasting-accuracy metric is chosen.

3.3.1 Forecasting models

The method most often used for forecasting ambulance demand is a simple averaging formula, called the MEDIC method, which is sensitive to how the area is divided. This means that typically the one hour demand in a 1 km^2 region is predicted by averaging a small number of historical data points from the same region over the corresponding hours from previous weeks or years (Zhou & Matteson, 2015).

The challenges experienced with forecasting ambulance demand can be explained by looking at Zhou & Matteson's (2015) Toronto EMS study. The Toronto EMS averages four historical data points in the same hour of the preceding four weeks for the past five years for which they want to forecast ambulance demand, i.e. the MEDIC method. The identified ambulance demand forecasting challenges:

1. the demand data at the requisite resolutions are often thinly dispersed: Toronto receives, on average, only 23 high priority calls per hour, and 96% of the 1 km^2 spatial regions have no events in any hour;
2. the demand often exhibits complex time-related dynamics and some are location-specific: weekly and daily seasonality and serial dependencies of a few hours, which were more pronounced in crowded neighbourhoods; and
3. the emergency call or ambulance demand data is generally large scale: Toronto dispatches for 200,000 priority calls every year.

The aforementioned challenges and the use of the MEDIC method, by the Toronto EMS, caused extremely noisy and fluctuating predictions, as most of the data points will be zero. So, dispatch based on these predictions will tend to be haphazard and inefficient. There are a number of studies that have researched methods that are better suited to forecasting ambulance demand as a time-related and location-specific process. The methods used in some of these studies were:

- autoregressive moving average models, where the predicted points are weighted moving averages of the past few forecasting errors (Channouf *et al.*, 2007);
- singular spectrum analysis (SSA), which is a non-parametric method, i.e. model-free technique, which outperforms traditional forecasting methods (Hassani, 2010; Vile *et al.*, 2012);
- artificial neural networks, which forecasts demand in discrete time and space, but does not give higher predictive accuracy than traditional forecasting methods due to the sparseness of the data (Setzler *et al.*, 2009);
- time-varying Ghaussian mixture model, which forecasts ambulance demand in discrete time and continuous space by incorporating location-specific time-related patterns in the demand, yielding higher predictive accuracy than traditional forecasting methods (Zhou & Matteson, 2015).

Of all the methods listed only SSA and the time-varying Ghaussian mixture method have been proven to outperform the industry methods. SSA was chosen for the DST because, unlike the time-varying Ghaussian mixture model, SSA does not require statistical knowledge and is easy to implement (Vile *et al.*, 2012; Zhou & Matteson, 2015). Also, SSA has been used to predict ambulance demand before by Vile *et al.* (2012).

3.3.2 Forecast-accuracy metric

The root-mean-square error (RMSE) is a commonly used forecast-accuracy metric in time series analysis, used to report how close the predicted data points are to the corresponding known data points (Channouf *et al.*, 2007; Matteson *et al.*, 2011; Vile *et al.*, 2012). The RMSE equation can be seen in (3.1). Let y_t be the observed value and $\tilde{y}_t$ the predicted value, with T fitted points in the time series.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T} (y_t - \tilde{y}_t)^2}{T}}. \tag{3.1}$$

Mean Absolute Error or Mean Absolute Percentage Error (MAPE) can also be used to gather conclusions similar to RMSE. RMSE is chosen for this project, however, since it overcomes the common problem encountered with MAPE – that the percentage error may become inflated if the actual value in the denominator is relatively small compared to the forecasting error. The RMSE metric also gives relatively higher weight to large errors, i.e. large difference between the predicted and the observed, which are particularly undesirable when forecasting EMS demand (Vile *et al.*, 2012). The higher weight will ensure that these errors are not missed.

3.4 Overview: Singular spectrum analysis

The origin of SSA is usually associated with the publication of papers by Broomhead & King (1986) and Broomhead *et al.* (1987). SSA is a novel and powerful technique used in the field of time series analysis. It is used for many practical problems, such as classical time series, multivariate statistics, multivariate geometry, dynamical systems and signal processing. Therefore, it can be applied in mathematics, physics, economics, financial mathematics, meteorology, oceanography, social sciences, market research, and medicine, among others. Essentially, SSA can be used for any seemingly complex time series (Hassani, 2010). At its core, SSA is a model-free technique which can be used without any statistical training (Golyandina *et al.*, 2001a). The basic capabilities of SSA can be shown by looking at the problems that can be solved (Hassani, 2010), such as:

- finding the trends of different resolutions of data;

- data smoothing;
- extraction of seasonality components;
- simultaneous extraction of cycles with small and large periods;
- extraction of periodicities with varying amplitudes;
- simultaneous extraction of complex trends and periodicities;
- finding structure in short time series;
- causality test; and
- forecasting.

Real time series usually go through structural changes during the time period under consideration; therefore it is crucial that the forecasting method is not sensitive to these variations. SSA is just such a method. Unlike the methods traditionally used for time series forecasting, such as autoregressive and structural models that assume normality and that the series is stationary, SSA is a non-parametric technique, i.e. model-free. It makes use of some probabilistic and statistical concepts, but no statistical assumptions, such as the stationariness of the series or normality of the residuals, are made (Hassani, 2010; Vile *et al.*, 2012). Hassani (2010) found SSA to be superior to classic techniques because of the complex structure of real-time series. SSA also works well for both large and small sample sizes. According to Vile *et al.* (2012), SSA is more flexible in approach and produces superior long term forecasts and comparable short term forecasts compared to well established methods.

SSA forecasting can be embedded into and used alongside a number of OR methods, i.e. queueing theory, simulation and optimisation models, in order to be used to deploy resources to achieve an improvement of ambulance service efficiency and quality. SSA has been shown to be an effective method of time series analysis, but Vile *et al.* (2012) is seen as the first to have used it to forecast ambulance demand. However, since then Gillard & Knight (2014) have also used SSA to forecast emergency demand and to analyse EMS staffing level requirements.

3.5 Singular spectrum analysis forecasting

According to Hassani (2010), the basic method of forecasting with SSA consists of two complementary stages, each with two separate steps, as seen in Figure 3.1. During stage one the time series is decomposed, and during stage two the original series is reconstructed without the noise components, and the reconstructed series is used to forecast new data points. The main concept in studying the properties of SSA is 'separability'; this defines how well different components – which show the seasonality of the time series – can be separated from each other.

SSA decomposes a time series into a sum of time series components, where each component can be classified as trend, periodic, quasi-periodic, or noise (Golyandina *et al.*, 2001a; Vile *et al.*, 2012). The steps of the SSA algorithm shown in Figure 3.1 will now be explained.

$$\mathit{Stage}\ 1\colon \mathit{Decomposition} \left\{ \begin{array}{c} \mathit{Step}\ 1\colon \mathit{Embedding} \\ \mathit{Step}\ 2\colon \mathit{Singular\ value\ decomposition} \end{array} \right.$$

$$\mathit{Stage}\ 2\colon \mathit{Reconstruction} \left\{ \begin{array}{c} \mathit{Step}\ 1\colon \mathit{Grouping} \\ \mathit{Step}\ 2\colon \mathit{Diagonal\ averaging} \end{array} \right.$$

Figure 3.1: The main stages of SSA.

Consider a real-valued non-zero one-dimensional time series $Y_T = (y_1, \cdots, y_T)$ with T data points. During the first step, *Embedding*, the trajectory matrix is computed. The time series, Y_T, is mapped to a multidimensional series of L-lagged vectors, $X_1, \cdots, X_K$, with vectors $X_i = (y_h, \cdots, y_{(h+L-1)})' \in \Re^L$ for $h = 1, 2, \cdots, K$ and $K = T - L + 1$. The parameter L is the embedding parameter, often called the window length. It is an integer that has to be less than or equal to $T/2$ and greater than two. It is generally selected that its value is proportional to the expected periodicity within Y_T (Golyandina *et al.*, 2001a; Hassani, 2010; Vile *et al.*, 2012).

The trajectory matrix $\mathbf{X} = [X_1, \cdots, X_K]$, shown in (3.2), is a Hankel matrix. This means that it is a square matrix, and all the elements along the anti-diagonal are identical (Golyandina *et al.*, 2001a; Hassani, 2010; Vile *et al.*, 2012).

$$\mathbf{X} = (x_{hg})_{h,g=1}^{L,K} = [X_1 : \cdots : X_K] = \begin{pmatrix} y_1 & y_2 & y_3 & \cdots & y_K \\ y_2 & y_3 & y_4 & \cdots & y_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_L & y_{L+1} & y_{L+2} & \cdots & y_T \end{pmatrix} \in \Re^{L \times K}. \tag{3.2}$$

In the second step, *Singular value decomposition*, the $\mathbf{XX}^T$ is computed and its eigenvalues and eigenvectors are determined. The eigenvalues and eigenvectors are then represented in the form $\mathbf{XX}^T = P\Lambda P^T$. $\Lambda = \text{diag}(\lambda_1, \cdots, \lambda_L)$ is the diagonal matrix of eigenvalues of $\mathbf{XX}^T$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L \geq 0$. $P = (P_1, P_2, \cdots, P_L)$ is the corresponding orthogonal matrix of eigenvectors of $\mathbf{XX}^T$ (Golyandina *et al.*, 2001a; Hassani, 2010; Vile *et al.*, 2012).

If $V_i = X_i' P_i / \sqrt{\lambda_i}$ (often called the 'principal components') for $i = 1, \cdots, d$, then the singular value decomposition (SVD) of the trajectory matrix can be written as shown in (3.3). Here $d = \text{rank}\,(X) = \max\,(i,$ such that $\lambda_i > 0)$ and $X_i = \sqrt{\lambda_i} P_i V_i'$. The matrices $\{X_i, i = 1, \cdots, d\}$ have a rank 1. They are therefore elementary matrices. The collection $\left(\sqrt{\lambda_i}, P_i, V_i\right)$ is called the i-th eigen-triple of the matrix X (Golyandina *et al.*, 2001a; Hassani, 2010; Vile *et al.*, 2012).

$$X = X_1 + \cdots + X_d. \tag{3.3}$$

The third step is called *Grouping*, and it is the splitting of the elementary matrices X_i into several groups and summing the matrices within each group. Sets of the matrices within $\{X_i, i = 1, \cdots d\}$ are selected; these represent various trend or periodic components of Y_T. The grouping procedure partitions the set of indices $\{1, \cdots, d\}$, seen in (3.3), into m disjointed subsets $I_1, \cdots, I_m$. Let $I = \{i_1, \cdots, i_p\}$; the resulting matrix X_I is defined as $X_I = X_{i_1} + \cdots + X_{i_p}$. This is calculated for $I = I_1, \cdots, I_m$ and leads to the decomposition $X = X_{I_1} + \cdots + X_{I_m}$ (Golyandina *et al.*, 2001a; Hassani, 2010; Vile *et al.*, 2012).

The final step, *Diagonal averaging*, is the reconstruction of the one-dimensional time series. The selection of $I_1, \cdots, I_m$ and the computation of $X = X_{I_1} + \cdots + X_{I_m}$ result in a matrix that does not have a Hankel structure. In order to find the approximate time series X has to be transformed into a Hankel matrix (Golyandina *et al.*, 2001a; Hassani, 2010; Vile *et al.*, 2012).

The matrix $\tilde{\mathbf{X}} = ||\tilde{x}_{h,g}|| = \sum_{k=1}^{l} P_{h_k} P_{h_k}^T \mathbf{X}$ is computed as an approximation to $\mathbf{X}$. It is then averaged over the diagonals of the matrix $\tilde{X}$ to transition to the one-dimensional series (Golyandina *et al.*, 2001a; Hassani, 2010; Vile *et al.*, 2012).

3.6 Overview: Ambulance location problems

The ambulance location problem forms part of a wide set of diverse problems from the public and private sectors that are solved by location analysis. The location analysis field's main focus is locating facilities across a nodal network (Knight *et al.*, 2012). Consider a graph $Gr = (V, E)$ where $V = D \cup H$, D and H are two vertex-sets that represent, respectively, the demand nodes and potential facility location nodes, i.e. ambulance station or holding site nodes, and E is the set of edges (Andrade & Cunha, 2015; Başar *et al.*, 2012). The edges connect the nodes and represent the roads that can be taken to move between the nodes. The edges can be used to calculated ambulance travel times, i.e. ambulance response times from the facility nodes to the demand nodes (Knight *et al.*, 2012).

Ambulance location problem models generally simplify the problem by using discrete demand points as input. These models can be classified as covering, p-median, or p-center models. Covering models focus on maximising demand coverage, where a demand point is covered if it can be reached within a predefined standard distance or response time by at least one ambulance. The p-center models aim to minimise the maximum service distance for all demand points and the p-median models aim to minimise the total or average service distance for all demand points (Li *et al.*, 2011).

There are a number of review articles on the progress of ambulance location problem models, and the majority have mainly considered covering models, as seen in Brotcorne *et al.* (2003), Başar *et al.* (2012), and Bélanger *et al.* (2015). The content of Brotcorne *et al.*'s (2003) review article focussed on static models, whereas the review articles of Başar *et al.* (2012) and Bélanger *et al.* (2015) focussed more on time-dependent models. The research in the following sections is focussed on covering models, since, as stated in Chapter 1, the DST's aim is to decrease response time and therefore increase coverage.

3.6.1 Static models

In this section deterministic single coverage, deterministic multiple coverage, and probabilistic and stochastic models are considered. These types of models strive to find either the minimum number of ambulances required to adequately cover an area or the maximum coverage that could be obtained given a finite fleet size. The two types of objectives formed two types of static models that are complementary, but used for different planning purposes. The minimisation model is used to decide on the size of the ambulance fleet, whereas the maximisation model provides an estimate of how well the system could perform over the planning horizon with a set fleet size. These models are used to provide solutions for the long term and demand is generally assumed to be constant for all the periods of the planning horizon, and the models are therefore static (Rajagopalan *et al.*, 2008). An overview on the three types of static ambulance location problem models are provided along with examples of each type.

3.6.1.1 Deterministic single coverage models

Deterministic location problem models ignore all stochastic (random) considerations of the ambulance location problem and are meant to be used in the planning phase. This type of static ambulance location problem model also does not take into account that ambulance coverage diminishes as soon as an ambulance is dispatched to a call, i.e. sent from the ambulance station or holding site (Brotcorne *et al.*, 2003).

Toregas *et al.* (1971) were the first to explicitly formulate the ambulance location problem using coverage. Simply defined, coverage is when a demand node can be reached by at least one ambulance within a time or distance standard. This definition was later expanded when queueing theory was used to determine the minimum number of ambulances that a demand node requires within a set response time or distance standard to be classified as covered for a specific service reliability (Farahani *et al.*, 2012; Li *et al.*, 2011). Toregas *et al.* (1971) defined the set covering location problem (SCLP), which strives to minimise the cost of locating ambulance stations or holding sites while satisfying a coverage limit (Farahani *et al.*, 2012). However, the number of ambulances that might be needed to achieve such complete coverage may be unrealistic. Generally, the EMS managers only want to determine the best usage of the ambulance fleet (Bélanger *et al.*, 2015). This is done by defining the problem in the form of the maximal covering location problem (MCLP), as proposed by Church & ReVelle (1974), which aims to maximise the population covered using a set ambulance fleet size (Farahani *et al.*, 2012).

Both SCLP and MCLP consider only one ambulance type. Realistically, it is common for there to be more than one type of ambulance in use. Therefore, in 1979 three models were proposed that looked at maximising population coverage that is simultaneously covered by two types of vehicles: tandem equipment allocation model (TEAM), multi-objective equipment allocation model (MOTEM), and facility-location, equipment-emplacement technique (FLEET) (Bélanger *et al.*, 2015; Brotcorne *et al.*, 2003; Schilling *et al.*, 1979).

3.6.1.2 Deterministic multiple coverage models

Deterministic single coverage models assume that an ambulance is always available when a call comes in. This is not a practical assumption. An ambulance may be unavailable due to it still serving a call when a new call in its area comes in. Therefore, deterministic single coverage models cannot provide robust solutions in real-life situations (Bélanger *et al.*, 2015). To overcome this drawback two methods were used: multiple coverage or including ambulance busy probabilities and ambulance station reliabilities (Li *et al.*, 2011). This section contains information on models that have incorporated multiple coverage. In Section 3.6.1.3 information is provided on models that explicitly consider busy probabilities and ambulance station reliabilities.

Multiple coverage models aim to increase the number of ambulances available to cover a demand node in order to increase the probability of having a node covered by one available ambulance at all time. These models indirectly consider the random nature of emergency demands through vehicle availability (Bélanger *et al.*, 2015).

Daskin & Stern (1981) proposed the hierarchical objective set covering problem (HOSC), which aims to minimise the number of ambulances to ensure complete coverage and to maximise the number of ambulances that can cover a demand node. Since HOSC does not consider the population of each demand node, it will tend to regroup ambulances around demand nodes that can be covered easily, and harder demand nodes might then only be covered once. Eaton *et al.* (1986) introduced the Dominican ambulance deployment problem (DADP) to overcome the weaknesses of HOSC, which strives to maximise the population that can be covered by more than one ambulance and minimise the number of ambulances needed to guarantee complete coverage. Hogan & ReVelle (1986) also strived to improve on HOSC by considering population density and giving hierarchical importance to the different coverage levels. These models are the backup coverage model 1 (BACOP1) and backup coverage model 2 (BACOP2), where both attempt to maximise the population covered twice given a fixed number of ambulances (Bélanger *et al.*, 2015; Brotcorne *et al.*, 2003).

Gendreau *et al.* (1997) introduced the double standard model (DSM), which considers the concept of double coverage, as well as different coverage standards. DSM determines the location of a fixed number of ambulances in order to maximise the population covered twice within a standard time frame. However, this model does cause some ambulances to be overused (Bélanger *et al.*, 2015; Brotcorne *et al.*, 2003). Storbeck (1982) introduced the maximal-multiple location covering problem (MMLCP), which aims to locate a fixed fleet of ambulances in order to minimise the population left uncovered, while maximising the number of demand zones covered by more than one vehicle (Bélanger *et al.*, 2015).

3.6.1.3 Probabilistic and stochastic models

Deterministic multiple coverage models presented a significant improvement over single coverage models, but they too have their limitations. For example, double coverage does not guarantee a satisfying service reliability level and might not even be necessary for an uncongested system. Probabilistic and stochastic models have been developed to overcome the limitations and to provide better real-life representation by explicitly considering ambulance busy probabilities (Bélanger *et al.*, 2015; Brotcorne *et al.*, 2003).

Expected coverage location models seek to establish the set of ambulance stations, or holding site locations, which maximises the expected coverage and considers ambulance busy probabilities, and therefore ambulance availability (Bélanger *et al.*, 2015). Daskin & Stern (1981) and Daskin (1983) integrated ambulance availability into MCLP, and were among the first to do so, and so introduced the maximum expected covering location problem (MEXCLP). The model aims to locate a given number of ambulances in order to maximise the expected coverage that depends on the ambulance busy probability, i.e. the probability that an ambulance is unavailable to respond to a call. The value is assumed to be the same for all vehicles, independent of their location, and to be known (Bélanger *et al.*, 2015; Li *et al.*, 2011).

Bianchi & Church (1988) proposed a variant of MEXCLP (MOFLEET) that independently considers the location of ambulance stations or holding sites and the assignment of ambulances to each. Daskin *et al.* (1988) also provided a more general MEXCLP version that could accommodate different coverage levels. The assumptions needed for MEXCLP to work are not generally met in practice, and that can lead to a significant variance between the systems's predicted and actual performance. Also, it considers deterministic travel times, i.e. the travel times are dependent on the initial state and inputs. Goldberg *et al.* (1990) proposed a model that has similar objectives to those of MEXCLP and its variations. This model determines the location of a given number of ambulances that maximises expected coverage, but it considers stochastic travel times. The dispatching decisions for this model are based on a preference list (Bélanger *et al.*, 2015).

ReVelle & Hogan (1988) formulated the probabilistic location set covering problem (PLSCP), which strives to minimise the number of ambulances needed to ensure that at least one vehicle is available for each demand node with a given level of reliability. This means node-specific busy probabilities are used, which are expressed as the ratio between the service time for demands arising in the node and the availability of the ambulances that can ensure their coverage. ReVelle & Hogan (1988) also proposed a probabilistic version of MCLP, the maximal availability location problem (MALP). MALP determines the location of a given number of ambulances that maximises the population covered by at least one available ambulance within the planning horizon, with a given level of service reliability. ReVelle & Marianov (1991) proposed a probabilistic version of FLEET, namely PROFLEET. This model aims to maximise the number of calls that can be covered simultaneously by two types of vehicles with a given level of service reliability (Bélanger *et al.*, 2015).

The MALP, PLSCP and PROFLEET models require the assumption that the busy probabilities are vehicle independent. This could impact the busy probability estimate and consequently the predicted system performances. Marianov & Revelle (1994) and Marianov & ReVelle (1996) relaxed this assumption to provide a more accurate estimate of actual system performance and proposed Q-PLSCP and Q-MALP. These two models are straightforward extensions of the PLSCP and MALP with queuing theory used to compute the number of ambulances required to ensure system reliability for each demand node (Bélanger *et al.*, 2015; Marianov & Revelle, 1994; Marianov & ReVelle, 1996).

Recently, researchers have continued to consider the static ambulance location problem with the aim of addressing some of the issues related to randomness more explicitly. The expected coverage definition was refined by Alsalloum & Rand (2006) to consider random travel times. The explicit consideration of the assignment of ambulances to emergency demands within a stochastic framework was undertaken by Beraldi *et al.* (2004), Beraldi & Bruni (2009) and Zhang & Jiang (2014) (Bélanger *et al.*, 2015). Therefore, static location problem models are still a relevant research area. This is also seen in that the time-dependent models, discussed in Section 3.6.2, are extensions or modifications of static location problem models.

3.6.2 Time-dependent models

The assumption of constant demand, which is required for the static ambulance location models described in Section 3.6.1, is unrealistic. Demand fluctuates throughout the week, day of the week, and even hour by hour within a given day (Rajagopalan *et al.*, 2008). The advancements in OR and time-analysis have moved the focus of ambulance location models towards multi-period and dynamic models. These models make use of time-dependent variables, such as demand rate, travel time, service time, and/or ambulance availability. Ambulance relocation also forms part of some of these models as a method of maintaining sufficient service levels for a system over time. A relocation decision is a dynamic response to the actual realisation of the demand (Bélanger *et al.*, 2015). In the following sections information on multi-period and dynamic models, some of which are adapted to consider relocation or can be adapted, are provided.

3.6.2.1 Multi-period models

While addressing the ambulance location problem in Louisville (Kentucky), Repede & Bernardo (1994) realised that there are no models that consider the variation of demand over time. They then formulated the maximal expected coverage location model with time variation (TIMEXCLP), a multi-period variation of MEXLCP (Bélanger *et al.*, 2015; Brotcorne *et al.*, 2003). According to Bélanger *et al.* (2015), TIMEXCLP is the first multi-period ambulance relocation model. Rajagopalan *et al.* (2008) introduced a multi-period variant of the PLSCP, the dynamic available coverage location model

(DACL). DACL strives to minimise the number of ambulances needed to guarantee the coverage of each demand node with a given level of service reliability, while taking into account several time periods. Neither the TIMEXCLP nor the DACL have any added constraints that take into account the relocation of ambulances (Bélanger *et al.*, 2015).

Schmid & Doerner (2010) introduced the multi-period double standard model (mDSM), which considers period dependent travel times. It is an extension of DSM in that it now considers the travel time variation between time periods due to factors such as traffic, and it adds a penalty term to the objective function to limit the number of relocated ambulances between periods (Bélanger *et al.*, 2015). Başar *et al.* (2012) considered determining the place and time that ambulance stations or holding sites should be open over a multi-period planning horizon. The model is called the multi-period backup double covering model (MPBDCM), and it is a combination of BACOP and DSM. MPBDCM differs from previous multi-period relocation problems, as it considers that, should a station or site be opened at one period, it must remain open until the end of the planning horizon. This addition is justified when changing the status of a station or site involves large costs or inconvenience (Bélanger *et al.*, 2015). Andrade & Cunha (2015) extended the Q-MALP model to a multi-period model and incorporated ambulance relocation. The model aims to maximise expected coverage and minimise relocation cost per hour over a shift.

Multi-period models consider the changes in the system that occur over time during the planning horizon, which is an improvement over the use of constant demand in static models. Multi-period models revise the values of the variables for pre-defined time periods of the planning horizon (Rajagopalan *et al.*, 2008; Schmid & Doerner, 2010). During each revision, some adjustments are required to keep each part of the model consistent with the other parts. The fact that the revisions are done at predefined time periods cause any change occurring between two consecutive time periods to make the model inconsistent with the dynamic setting of the system. This inconvenience lead to the development of dynamic models (Moeini *et al.*, 2014).

3.6.2.2 Dynamic models

Gendreau *et al.* (2001) were the first to propose an ambulance relocation model that explicitly accounts for the dynamic nature of EMS. This model, called the ambulance location problem (RP^t), is based on the DSM. The RP^t still strives to maximise the population covered by at least two vehicles within the standard time frame, but also to minimise relocation costs. The objective function contains a penalty term that takes into account the relocation history of ambulances. The penalty term ensures the avoidance of overly long and round-trips, as well as moving the same ambulance repeatedly; it is also updated each time a relocation is performed. RP^t needs to be solved every time an ambulance is dispatched, i.e. the system's state changes. However, in the EMS context relocations need to be decided on in real-time, and the computational time needed to solve such a relocation problem may be too long to consider each time an ambulance is dispatched. Gendreau *et al.* (2001) suggested that the time between two calls should be used to determine the relocation plan associated with each possible dispatching decision. Therefore, as soon as the dispatched ambulance becomes known the corresponding relocation plan can be applied immediately (Bélanger *et al.*, 2015).

Andersson & Värbrand (2006) proposed a dynamic ambulance relocation model, DYNAROC, which differs from Gendreau *et al.* (2001) in the way that it assesses the system's performance. Andersson & Värbrand (2006)'s model considers the preparedness measure – the capacity of the system to answer future calls – rather than a coverage measure. In practice, the level of preparedness for each demand node is regularly checked, and when it drops below a certain value the relocation of ambulances start (Bélanger *et al.*, 2015).

Gendreau *et al.* (2006) proposed the maximal expected relocation problem (MECRP). It determines the appropriate relocation plan for each possible system's states. This approach is only possible for problems with a relatively small number of ambulance vehicles. Nair & Miller-Hooks (2009), in a similar way to Gendreau *et al.* (2006), presented a location-relocation model that considers the evolution of the system's state over time. The system's state at a given time is defined by the incoming call probability distributions, the number of available ambulances, and the travel time within the road network at that particular time. This model has two objectives: to maximise the double coverage and to minimise the location-relocation costs (Bélanger *et al.*, 2015).

Maxwell *et al.* (2010) considered dynamic programming to formulate the dynamic ambulance relocation problem. The problem is limited to the relocation of vehicles that have just completed their mission, so it significantly reduces inconvenience and the number of possible decisions. When an ambulance completes its mission the relocation problem considered assists in determining the next holding site for that ambulance in such a manner that the number of calls that can be reached within the given time frame is maximised (Bélanger *et al.*, 2015). Schmid (2012) also proposed using dynamic programming to formulate the dynamic relocation problem. The relocation decisions are considered when an ambulance has completed its mission and the only vehicle available for relocation is the newly idle ambulance. Its objective is to minimise the average response time over a finite planning horizon, while considering the variation of the travel times and demand density with respect to time (Bélanger *et al.*, 2015).

Naoum-Sawaya & Elhedhli (2013) considered addressing the dynamic ambulance relocation problem by means of a two-stage stochastic programming approach. The first stage is the vehicle location and the second the assignment of vehicles to emergency demands upon actual reception of the emergency demands. The objective is firstly to minimise the number of relocated ambulances and then the number of demands that cannot be served within the prescribed delay (Bélanger *et al.*, 2015).

Mason (2013) proposed the real-time multi-view generalised-cover repositioning model (RtMvGcRM), which is implemented within the EMS management software called Optima Live. Optima Live provides real-time relocation recommendations. It strives to determine the location of available ambulances in such a manner that service quality is maximised and relocation costs are minimised. Jagtenberg *et al.* (2015) proposed a dynamic variant of MEXCLP to address a real-time relocation problem with the goal of minimising the expected fraction of late arrivals. It assumes that an ambulance is only allowed to be relocated at the end of a mission. Despite its simplicity, it has been shown that this model performs better than the static policy where an ambulance always returns to its home base (Bélanger *et al.*, 2015).

3.7 Location-allocation and relocation models

It was stated in Section 1.2.1.1, that this project's primary goal is to create a concept demonstrator DST that can be utilised to produce a solution for the near-optimal (i) holding site placement per shift, and (ii) ambulance allocation and relocation per hour of that shift, for ambulance demand from the City of Cape Town and the Cape Winelands municipalities. Therefore, a location-allocation and relocation model is required. It was also stated in Section 1.2.3 that integrating the DST into the WC ECC's system is not part of the project's scope. Thus, dynamic models will not be considered. This leaves static ambulance location models or multi-period models.

The ambulance location-allocation and ambulance relocation problems are extremely similar; both aim to find the optimal locations for ambulances as a function of the system's demand (Nguyen, 2015). Ambulance location-allocation problems are strategic and allow for off-line computation, while relocation problems require more realistic data or the implementation of procedures in real-time (Nguyen, 2015). It is possible to formulate the two problems together by considering time-dependent models, i.e. time-dependent demand and/or travel times. The relocation of ambulances can then be preplanned by solving for ambulance location per time period of the planning horizon (Nguyen, 2015; Rajagopalan *et al.*, 2008; Schmid & Doerner, 2010). This is exactly what is possible with multi-period models. The following sections briefly describe three multi-period models that were considered for this project. These models consider relocation or can be adapted to consider ambulance relocation. The WC ECC's real-world instance and the choice of model will be described in Chapter 4.

3.7.1 Time-dependent MEXCLP with start-up and relocation cost

Van den Berg & Aardal (2015) introduced a time-dependent MEXCLP with start-up and relocation cost. They based the model on Repede & Bernardo (1994)'s extension of MEXCLP, called TIMEXCLP. The TIMEXCLP model aims to maximise the expected coverage over the day by partitioning the day into multiple time periods. The fact that there is no relation between the time periods allows the problem to be solve independently for each period. Therefore, there can be huge differences in the location of ambulances between the different time periods, resulting in high relocation costs.

Schmid & Doerner (2010) introduced an extension of DSM, a time-dependent model which assumes time-dependent travel times. DSM had the same issue as TIMEXCLP with high relocation costs, and to overcome it Schmid & Doerner (2010) added a penalty to the number of relocations between periods. Van den Berg & Aardal (2015) decided to adopt the same practice to control the movement of ambulances between periods to different locations. Van den Berg & Aardal (2015) did not have a time-dependent travel time model and therefore used average drive time as an indication of speed during a time period. However, they did use time-dependent demand and ambulance availability.

The model was solved using CPLEX 12.5 with a time limit of 5 minutes. To test the model it was applied to two different instances. The input for the first instance was randomly generated with 500 demand points and 50 possible locations. The input for the second instance was real-life data for a region of Amsterdam, Netherlands, with 161 demand points and a population of 1.2 million. The demand nodes were created based on 4-digit postal codes. It is assumed that the demand nodes are also available ambulance station locations. In the Netherlands 95% of all high priority calls must be reached within 15 minutes. This response time target can only be met if ambulances are placed in order to provide adequate coverage (Van den Berg & Aardal, 2015).

3.7.1.1 Problem model formulation

Let d_i^t denote the demand rate at demand node i during time period t, and p_A^t denote the number of available ambulances during time period t. The decision variables are y_j^t, the number of ambulances located at location j during time period t, and x_i^{wt}, which indicates whether demand node i is covered by at least w ambulances during time period t. Due to the time-dependent demand, service time, and ambulance availability, the model also has a time-dependent busy probability, q^t. The set of ambulance stations or holding site location nodes that can cover demand node i during time period t is denoted by V_i^t. The relation between consecutive time periods are taken into account by adding a penalty, γ, for the number of relocations between the time periods. The integer variable $r_{jj'}^t$ represents the number of ambulances that should be repositioned from location j to location j' between time period t and $(t+1)$. Another penalty variable, θ, indicates the cost of using an ambulance station location during at least one time period, and is added to each location used. A binary variable z_j is used to indicate whether a location j is used during at least one time period. Van den Berg & Aardal (2015) made use of big-M constraints to ensure correct values for these variables. These constraints use a constant, $bigM$, which has a very large value.

The model can then be formulated as follows:

$$[max] \sum_{t\in\tau}\sum_{i\in D}\sum_{w=1}^{p_A^t} d_i^t\left(1-q^t\right)\left(q^t\right)^{(w-1)} x_i^{wt} - \theta\sum_{j\in H} z_j - \gamma\sum_{i\in D}\sum_{j\in H}\sum_{t\in\tau} r_{jj'}^t. \tag{3.4}$$

Subject to:

$$\sum_{j\in V_i^t} y_j^t \geq \sum_{w=1}^{p_A^t} x_i^{wt} \quad i \in D, t \in \tau; \tag{3.5}$$

$$\sum_{j\in V_i^t} y_j^t \leq p_A^t \quad t \in \tau, \tag{3.6}$$

$$\sum_{j\in V_i^t} y_j^t \leq (bigM)(z_j) \quad j \in H, \tag{3.7}$$

$$y_j^t + \sum_{i\in D} r_{jj'}^t - \sum_{i\in D} r_{j'j}^t = y_j^{(t+1)} \quad t \in \tau, j, j' \in H; \tag{3.8}$$

$$y_j^T + \sum_{i\in D} r_{jj'}^T - \sum_{i\in D} r_{j'j}^T = y_j^{(t+1)} \quad T \in \tau, j, j' \in H; \tag{3.9}$$

$$y_j^t, r_{jj'}^t \text{ integers}; \tag{3.10}$$

$$x_i^{wt}, z_j \in \{0, 1\}. \tag{3.11}$$

The first term of the objective function, (3.4), calculates the expected coverage over all demand nodes and all time periods. The second term considers the number of locations used and penalises the use of each based on their cost. The third term penalises the number of ambulances relocated between time periods.

Constraint (3.5) ensures that a demand node is covered by at least w ambulances if at least w ambulances can reach this demand node within the response time or coverage standard. Constraint (3.6) ensures that no more than p_A^t ambulances are used during time period t. For this constraint to be valid $bigM$ should be at least the value of the left hand side. However, a too high value can result in a weak linear program relaxation and an increase in computation time. Constraint (3.7) indicates that the number of ambulances at a location can only be positive if the location is opened. Constraint equations (3.8) and (3.9) ensure that the values of $r_{jj'}^t$ are correct and balanced for every location. If the number of ambulances is not constant throughout the day, a dummy location is required, off-duty ambulances are then sent to this location.

3.7.2 Dynamic available coverage location model

Rajagopalan *et al.* (2008) formulated the multi-period model for dynamic demand environments, called DACL, in order to minimise the required number of ambulances while meeting the predetermined ambulance availability requirements. DACL allows solving for multiple periods and incorporates ambulance busy probabilities by using Jarvis's (1985) hypercube approximation algorithm. Both the multi-period and ambulance busy probabilities enhance the realism of this model (Rajagopalan *et al.*, 2008).

The model was applied to data from 2004 for Mecklenburg County, North Carolina, a region spread over approximately 1,398.59 km^2 with a population of 801,137. Mecklenburg county received 77,292 calls during 2004, of which 61,630 were classified as emergencies requiring an ambulance to be dispatched. The data for every Monday of 2004 for Mecklenburg County was used in the model, for which there were 8,742 ambulance dispatched. The time periods were defined to consists of three hours each. However, the time value of the time periods can easily be changed. The county was divided into square nodes of 5.18 km wide and 5.18 km in length, which provided Rajagopalan *et al.* (2008) with 168 demand nodes. It was assumed that the ambulances could be stationed in any of the nodes except those that are part of the boundary. A tabu search heuristic was developed to solve the model, and the solutions were validated with a comprehensive simulation model (Rajagopalan *et al.*, 2008).

3.7.2.1 Problem model formulation

Let amb be the set of available ambulance and a_j^{bt} be 1 if ambulance b ($b \in amb$) is located at node j during time period t. Then, p_A^t is the number of available ambulances during time period t, d_i^t the demand rate at node i during time period t, n the number of nodes in the system, and c^t the minimum expected coverage requirement during time period t. q_i^t indicates the busy probability of an ambulance at node i during time period t, η^t the average system busy probability during time period t, P_0 the probability of having all servers free, and P_w the probability of having all servers busy in an $M/M/w/0{-}loss$ system with w ambulances. $Q(p_A^t, \eta^t, i)$ is a correction factor for Jarvis's (1985) algorithm, which adjusts the probabilities for server cooperation in the models. Let

$$Q(p_A^t, \eta^t, i) = \frac{\sum_{l=i}^{p_A^t - 1} \left(p_A^t - i - 1\right)! \left(p_A^t - l\right) \left(\left(p_A^t\right)^l\right) \left(\left(\eta^t\right)^{l-i}\right) P_0}{(l-i)! \left(1 - P_{p_A^t}\right)^i p_A^t! \left(1 - \eta^t \left(1 - P_{p_A^t}\right)\right)} \qquad \forall\, i = 0, 1, \cdots, m-1. \tag{3.12}$$

Also, x_i^t is equal to 1 if demand node i is covered by at least one ambulance with α^t service reliability during time period t, else 0. If demand node i is within a set distance from an ambulance at node j during time period t, let dis_{ji}^t be equal to 1, else 0.

The model can then be formulated as follows:

$$[min] \sum_{t=1}^{T} \sum_{l=1}^{n} \sum_{b \in amb} a_j^{bt}. \tag{3.13}$$

Subject to:

$$\begin{aligned} \Bigg[\Bigg\{ 1 - \prod_{b=1}^{p_A^t} (q_i^t)^{\sum_{j=1}^{n} dis_{ji}^t a_j^{bt}} Q \left(p_A^t, \eta^t, \sum_{i=1}^{n} \sum_{b=1}^{p_a^t} dis_{ij}^t a_j^{bt} - 1 \right) \Bigg\} & \\ -\alpha^t \Bigg] x_i^t \geq 0 \quad \forall\, i, t; & \qquad (3.14) \\ \sum_{i=1}^{n} d_i^t x_i^t \geq c^t \quad \forall\, t; & \qquad (3.15) \\ x_i^t, a_j^{bt} = \{0, 1\} \quad \forall\, i, j, b, t. & \qquad (3.16) \end{aligned}$$

Objective function (3.13) minimises the number of ambulances deployed. Constraint (3.14) keeps track of the nodes that are covered and ensures it is done with the required service reliability. Constraint (3.15) ensures that the coverage of the total system will be greater than c^t but together with constraint (3.14) it allows only the demand nodes covered with α service reliability to be included in the system wide expected coverage (Rajagopalan *et al.*, 2008).

3.7.3 Extended queuing maximum availability location problem model

Andrade & Cunha (2015) modified the Q-MALP model, which was created by Marianov & ReVelle (1996). The modifications turned the static model into a multi-period model which considers ambulance relocation. The modified model is called the extended Q-MALP model. The extended Q-MALP model was created to be used as an optimised-based DST to guide ambulance station location and ambulance allocation decisions, while also determining the best relocation of ambulances in order to cope with the varying demand for the São Paulo EMS. A secondary purpose was to create a generic and flexible model that can be adapted to other large real-world problems (Andrade & Cunha, 2015).

According to Andrade & Cunha (2015), they were the first to apply the Q-MALP to such a large real-world problem. In order to incorporate relocation in the model a few of the concepts developed by Schmid & Doerner (2010) regarding the relocation of ambulances were included. The following concepts were added:

1. moveable stations;
2. multiple time periods, with optimal allocation for successive periods (not a purely static solution);
3. time-dependent variations in travel times and the resulting changes with respect to coverage and required vehicle relocation;
4. independent location decisions for stations and ambulances;
5. multiple types of vehicles with different coverages, basic life support (BLS), and advanced life support (ALS); and
6. capacity constraints for each potential site location.

São Paulo is the largest city in South America with a population of up to 11.9 million people in 2014. It is Brazil's financial and business centre, housing the headquarters of the largest national and multi-national corporations. São Paulo therefore attracts daily commuters who live in the neighbouring cities. The neighbouring cities have a population of up to 20 million people. The commuters and large population already in São Paulo has caused traffic congestions to extend past peak driving hours. In 2007 the best feasible response time for 98% of the calls was within 27 minutes. This lead to a restructuring in 2009, which tried to reduce the total response times and to bring it closer to an acceptable standard. During the process it became evident that more ambulance stations were required to reduce the travel time between the closest ambulance and the incident. This realisation lead to an optimisation-based DST to guide the São Paulo EMS in its strategic decisions involving the location of ambulance stations and the allocation and relocation of ambulances in order to cope with the varying demand of different time periods. An integer programming model was used to represent the problem. The solution method implemented was a meta-heuristic based on the artificial bee colony (ABC) algorithm (Andrade & Cunha, 2015).

3.7.3.1 Problem model formulation

Let $i \in D$ represent the set of demand nodes and $j \in H$ the set of potential location nodes for ambulance stations. The planning horizon consists of time periods indicated by $\tau = \{1, 2, \cdots, t, \cdots, T\}$. The model also considers the use of two ambulance vehicles, ALS and BLS, denoted by $k \in \{1, 2\}$, respectively. The stochastic nature of the problem model is indicated by three different neighbourhoods, W_i^{kt}, V_j^{kt}, and N_i^{kt}. W_i^{kt} represents the subset of location nodes that can be reached from demand node i during time period t in less than r^k time units. r^k is the time standard possible, i.e. coverage standard in terms of time, for an ambulance of type k. V_j^{kt} represents the subset of demand nodes that can be reached from location node j during time period t in less than r_k time units. N_i^{kt} represents the subset of all demand nodes, $z \in D$, that can be reached from demand node i during time period t in less than r_k time units. The neighbourhood N_i^{kt} is assumed to be an $M/G/s{-}loss$ queuing system, with Poisson distributed call arrival rate, generally distributed service times, s ambulances in the neighbourhood, and up to s calls being serviced at the same time. Queuing theory solutions for steady-state equations are used to estimate the likelihood that all servers are busy at the same time; this allows for the calculation of ambulance busy probabilities and the minimum number of ambulances required inside N_i^{kt} to provide coverage to demand node i.

The binary decision variable x_i^{wkt} is equal to 1 if, and only if, demand node i is covered by at least w ambulances of type k during time period t; otherwise it is equal to 0. y_j^{kt} is an integer variable that denotes the number of ambulances of type k located at location j during time period t, and the binary decision variable z_j is equal to 1 if, and only if, an ambulance station is located at location node j and 0 otherwise. The integer variable $r_{jj'}^{kt}$ denotes the number of ambulances of type k that are to be repositioned from location j to location j' $(j, j' \in H)$ between time period $(t-1)$ and t.

The objective function strives to determine the best p_Z locations for ambulance stations, as well as to allocate p_A^k ambulances of type k among the selected stations at each time period t such that the expected coverage over all time periods is maximised and simultaneously the total time spent in relocating ambulances is minimised. Queuing theory is used to determine the minimum number of ambulances of type k, M_i^{kt}, required to adequately cover demand node i with an α service reliability level during time period t.

The model can then be formulated as follows:

$$[max] \sum_{t \in \tau} \left[\sum_{k=1}^{2} \sum_{i \in D} \sum_{w=0}^{M_i^{kt}} d_i^{kt} \cdot C_i^{wkt} \cdot x_i^{wkt} - \beta \cdot \sum_{j \in H} \sum_{j' \in H} s_{jj'}^{t} \cdot r_{jj'}^{kt} \right]. \tag{3.17}$$

Subject to:

$$\sum_{j \in W_i^{kt}} y_j^{kt} \geq 1 \quad i \in D, t \in \tau, k \in \{1,2\}\,; \tag{3.18}$$

$$\sum_{j \in W_i^{kt}} y_j^{kt} \geq \sum_{w=0}^{M_i^{kt}} x_i^{wkt} \quad i \in D, t \in \tau, k \in \{1,2\}\,; \tag{3.19}$$

$$x_i^{wkt} \leq x_i^{(w-1)kt} \quad i \in D, t \in \tau, k \in \{1,2\}\,, w \in \left\{1, 2, \cdots, M_i^{kt}\right\}; \tag{3.20}$$

$$p_k \cdot z_j \geq y_j^{kt} \quad j \in H, t \in \tau k \in \{1,2\}\,; \tag{3.21}$$

$$y_j^{kt} + \sum_{i \in D} r_{jj'}^{kt} - \sum_{i \in D} r_{j'j}^{kt} = y_j^{k,(t+1)} \quad j, j' \in H, t \in \tau - \{T\}\,, k \in \{1,2\}\,; \tag{3.22}$$

$$y_j^{kT} + \sum_{i \in D} r_{jj'}^{kT} - \sum_{i \in D} r_{j'j}^{kT} = y_j^{k1} \quad j, j' \in H, k \in \{1,2\}\,; \tag{3.23}$$

$$\sum_{j \in H} z_j = p_Z; \tag{3.24}$$

$$\sum_{j \in H} y_j^t = p_A^1 \quad t \in \tau; \tag{3.25}$$

$$\sum_{j \in H} y_j^t = p_A^2 \quad t \in \tau; \tag{3.26}$$

$$y_j^{1,t} + y_j^{2,t} \leq C_j \quad j \in H, t \in \tau; \tag{3.27}$$

$$y_j^{kt} \geq 0 \text{ and integer} \quad j \in H, t \in \tau, k \in \{1,2\}\,; \tag{3.28}$$

$$x_i^{wkt} \in \{0,1\} \quad i \in D, t \in \tau, k \in \{1,2\}\,, w \in \left\{0, 1, 2, \cdots, M_i^{kt}\right\}; \tag{3.29}$$

$$z_j \in \{0,1\} \quad j \in H; \tag{3.30}$$

$$r_{jj'}^{k,t} \geq 0, \text{ and integer} \quad \left(j, j^{'}\right) \in H, t \in \tau, k \in \{1,2\}\,. \tag{3.31}$$

Constraint (3.18) ensures that every demand node i is covered at least once within

r^k time units, during time period t by ambulance type k. Constraint (3.19) implies that a demand node, i, is covered M_i^{kt} times only if at least M_i^{kt} ambulances of type k are stationed within the given response time standard, r^k, from the demand node i (Marianov & ReVelle, 1996). Constraint (3.20) ensures demand node i is covered by w ambulances, only if it is firstly covered by $(w-1)$, i.e. it ensures continuity. Constraint (3.21) specifies that all available ambulances must be assigned to selected location site nodes. Constraint equations (3.22) and (3.23) ensure that the ambulance relocations between consecutive time periods take place consecutively; this is based on work by Schmid & Doerner (2010). Constraint (3.23) also guarantees that the relocation plan is connected. Constraint (3.24) ensures that the number of selected stations is equal to p_Z. Constraint equations (3.25) and (3.26) ensure that p_A^k ambulances of type k are located in all time periods. Constraint (3.27) ensures the capacity restriction at each holding site location j is not exceeded. Constraint (3.28) ensures that the number of ambulances of type k, y_j^{kt}, to be located at each selected site j is positive and integer. Constraint (3.29) ensures that the x_i^{wkt} variable modelling the coverage is binary. Constraint (3.30) ensures that the z_j variable indicating the location site nodes selected is binary. Constraint (3.31) indicates the number of ambulances of type k, $r_{jj'}^{kt}$, to be repositioned between time period $(t-1)$ and t from j to j' ($\{j, j' \in H\}$) and ensures that the value is integer and greater than zero.

3.8 Ambulance location problems: Solution methods

An ambulance location problem can be classified as a discrete facility location problem, where the solution space for locating p_A ambulances in n nodes is $n \times p_A$ (Basu *et al.*, 2015). There are various optimisation techniques that can be used to solve these types of problems; the three main techniques are: heuristic and meta-heuristic algorithms, simulation, and exact methods.

In general ambulance location problem models are formulated as integer programming problems, and it is possible to apply exact methods, i.e. the branch-and-bound algorithm, to obtain optimal solutions but only for small scale problems. This is because this type of problem tends to be NP-hard, for which exact methods are limited (Basu *et al.*, 2015; Rajagopalan *et al.*, 2008). Simulation is usually used to evaluate the system's performance, or it is combined with heuristics and/or meta-heuristics to provide near optimal solutions. Therefore, if the problem is too large and simulation is not required heuristics and/or meta-heuristics are the preferred solution technique (Li *et al.*, 2011).

Heuristics are generally problem specific, whereas meta-heuristics tend to have a problem independent structures. Meta-heuristics consist of different components which exploit problem related information, and can be implemented in many different ways to solve a problem (Basu *et al.*, 2015). The reasoning behind choosing one meta-heuristic and the method of implementing its components over another is not purely objective as is the case with exact methods. Generally, the meta-heuristic and the implementation of its components are chosen randomly or because it was the method used in literature to solve the basic variant of the model (Basu *et al.*, 2015).

The solution methods that were implemented to solve the ambulance location problem models described in Section 3.6 were mostly meta-heuristics. The following is a list of heuristics and meta-heuristics that have been used to solve these and some other ambulance location problem models:

- integer linear programming, the MEXCLP (Jagtenberg *et al.*, 2015);
- tabu search, minimum expected response location problem (MERLP) (Rajagopalan & Saydam, 2009);
- artificial bee colony algorithm (ABC), the extended Q-MALP (Andrade & Cunha, 2015);
- genetic algorithm (GA), the Maximal Expected Survival Location Model for Heterogeneous Patients (MESLMHP) (Knight *et al.*, 2012) and DSM (Liu *et al.*, 2014);
- ant colony optimisation, the double coverage model (Su *et al.*, 2015) and the modified double coverage model (Luo *et al.*, 2013); and
- simulated annealing (SA) and GA, the spatial queuing model (SQM) and MCLP (Mohammadi *et al.*, 2014).

Simulation is not part of the scope of the project, and the WC ECC's real-world instance is too large for the use of exact methods. The above list shows that the preferred method for solving ambulance location problem models is meta-heuristics.

3.9 Meta-heuristics

There are a myriad of real-life hard, NP-hard, optimisation problems, such as the ambulance location problem being considered in this project (Rajagopalan *et al.*, 2008), that require high quality solutions. This led to the creation of many algorithms, of which heuristics and meta-heuristics are the most popular.

Heuristics are experience-based techniques (Muritiba, 2010). Therefore, they tend to be the first thing used to solve a new optimisation problem. Meta-heuristics were developed in 1970 to replace or improve on heuristics, as they tended to easily become stuck in local optima, which were not the global optimum. The idea was that meta-heuristics would guide lower-level procedures, i.e. heuristics, to find good-quality (near-optimal) solutions (Blum *et al.*, 2008a; Lei *et al.*, 2015).

A good meta-heuristic algorithm can be identified by how well it uses and balances exploration and exploitation techniques to find the global optimum. A balance is required between the two techniques, as they are contradictory. Exploration is the investigation of new and unknown areas of the solution search space, and exploitation is the use of knowledge of old solutions to find better solutions (Busetti, 1983).

Meta-heuristics can be classified in more than one way. The different classification types refer to different ways in which researchers differentiate between meta-heuristics. The following list shows the three most prominent classifications types (Blum & Roli, 2003):

1. nature-inspired versus non-nature inspired;
2. memory-based versus memory-less; and
3. single-point versus population-based meta-heuristics.

The third classification type is the one most often used. It refers to the number of solutions that is considered by the optimisation algorithm during any iteration. The single-point algorithms consider only one possible solution at a time and are also referred to as trajectory methods. Population-based algorithms consider more than one possible solution at a time (Blum *et al.*, 2008a).

3.10 Single-point meta-heuristic algorithms

Single-point algorithms are an extension of the iterative improvement local search procedures, i.e. a type of heuristic. This type of heuristic tends to provide unsatisfactory solutions, as the quality of the solution depends greatly on the starting position. Single-point algorithms have added an exploration component to the iterative improvement local search procedure. This component guides exploration of the search space, i.e. all possible solutions, to find better and new solutions, but the algorithm still requires a termination criteria, such as maximum computation time and/or number of iterations. Examples of single-point algorithms are SA (Mohammadi *et al.*, 2014), tabu search (Chanta *et al.*, 2014; Rajagopalan *et al.*, 2008), iterated local search, and variable neighbourhood search (VNS) (Blum *et al.*, 2008a). The two most popular algorithms are SA and tabu search.

3.10.1 Simulated annealing

SA was proposed by Kirkpatrick *et al.* (1983) to be a probabilistic method for finding the global minimum of an objective function that might have some local minima. It is one of the oldest meta-heuristics and one of the first to explicitly incorporate a strategy to escape local optima, called hill-climbing (Bertsimas & Tsitsiklis, 1993).

The SA algorithm imitates the metallurgic process of annealing, when metal is heated until its melting point and slowly cooled until it freezes into a minimum energy crystalline structure which has fewer crystal defects. If the cooling schedule is slow enough, the end result would be a solid with superior structural integrity (Blum *et al.*, 2008b; Busetti, 1983; Henderson *et al.*, 2003).

During an iteration two possible solutions are evaluated, the current and a newly selected solution. The objective function value of these two solutions are compared. If the new solution has a higher objective function value then it replaces the current solution. If the new solution is worse it can still replace the current solution based on a probability algorithm, i.e. hill-climbing; this allows a fraction of non-improving solutions to be accepted with the hope of escaping local optima. The probability of accepting a non-improving solution decreases as the ‘temperature’ parameter is decreased with each iteration. As the temperature parameter is decreased to zero, and hill-climbing happens less and less, the solution distribution converges until all the probability is concentrated on the set of globally optimal solutions. This is only possible if the algorithm is convergent; otherwise the algorithm will converge to a local optimum, which may or may not be globally optimal (Henderson *et al.*, 2003). The choice of the next trial solution depends exclusively on the current solution, since SA does not use memory (Blum *et al.*, 2008b; Henderson *et al.*, 2003).

SA can handle highly non-linear models, chaotic and noisy data, and constrained optimisation. It can also easily be adapted for use in more than one problem. However, SA requires the user to make a lot of choices concerning the values of control parameters and the cooling schedule. These decision influence the quality of the solution and the length of the computation time required for convergence. The process of changing the algorithm for all the different types of constraints, and to tune the parameters can be extremely delicate work. Furthermore, the SA algorithm tends to be computationally intensive (Busetti, 1983).

3.10.2 Tabu search

Tabu search is a local search based improvement meta-heuristic. The initial solution is important for any local search based heuristic, as it sets up the initial search space, and this holds true for tabu search. The strategies that are usually employed for generating initial solutions for tabu search include randomly generating initial solutions, greedy approach, and initial solution based on some problem characteristic (Basu *et al.*, 2015). The tabu search (Glover & Greenberg, 1989) framework can be used as a general framework for a variety of iterative local search strategies for discrete optimisation.

The algorithm starts with an initial solution, calls it the current solution, and then searches for the best solution in an acceptably defined neighbourhood around the current solution. The best solution found in the neighbourhood then becomes the current solution, and the search process starts again. The process terminates when the termination condition is met. The most typical termination criteria are: maximum number of iterations, maximum number of iterations without any improvement in solution value, and maximum execution time (Basu *et al.*, 2015; Blum *et al.*, 2008b; Henderson *et al.*, 2003).

During the search process, the algorithm keeps track of the current solution and the best solution found thus far. To prevent the algorithm from returning to previous solutions, or getting stuck, a list of forbidden moves are kept, and the solutions that can only be reached through these tabu moves are removed from the neighbourhood. A move stays on the tabu list until the list is full; then the oldest move is deleted. The tabu list adds a memory aspect to the tabu search algorithm and helps it to avoid getting stuck in local optima (Blum *et al.*, 2008b; Henderson *et al.*, 2003).

The length of the tabu list controls the memory of the algorithm. If it is small the search will be concentrated on small areas of the search space, and if it is large it will force the algorithm to explore larger areas of search space. The length can be kept constant, changed at discrete intervals, or be continuously changed during the execution of the search (Basu *et al.*, 2015; Blum *et al.*, 2008b). Tabu lists are generally implemented in a FIFO manner. The lists store the features of the recently visited solutions. Different lists may be used for each type of solution feature, each of which is initialised at the start of the algorithm as empty lists (Blum *et al.*, 2008b). Often an aspiration criteria is added which if met will allow tabu moves to be used to reach a solution. The most common criterion states that if the solution's quality is the best so far then the tabu status is withdrawn (Basu *et al.*, 2015).

Unlike the SA algorithm, tabu search uses memory but there is still no proof of convergence existing in the literature for the general tabu search algorithm (Blum *et al.*, 2008b; Henderson *et al.*, 2003). The process of creating and maintaining the tabu list or lists can also be intensive.

3.11 Population-based meta-heuristic algorithms

Population-based algorithms consider a set of potential solutions at each iteration, and the next set of solutions is created by applying one or more operators to the current set of solutions. The quality of the new set of solutions and the efficiency of the method are dependent on the operator(s) used to manipulated and change the solution set (Blum *et al.*, 2008a). Two of these types of algorithms that are used often for ambulance location problems are GA, an evolutionary computation algorithm (Knight *et al.*, 2012; Lim *et al.*, 2011) and ABC, a particle swarm optimisation algorithm (Andrade & Cunha, 2015).

3.11.1 Genetic algorithm

GAs were created and developed in the 1960s at the University of Michigan by John Hollard, his colleagues, and his students (Melanie, 1999). Hollard did not originally plan to develop GA as an optimisation algorithm but to use it to formally study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanisms of natural adaptation might be used in computer systems. The basic GA, therefore, does not offer any statistical guarantee of convergence to a global optimum when solving for any optimisation problem (Busetti, 1983; Melanie, 1999). However, GAs are still used, as there is a need for an algorithm that can search through a huge number of possibilities for good solutions. Also, many computational problems require a program that is able to adapt and perform well in a changing environment (Melanie, 1999).

The GA algorithm is a direct, parallel, stochastic search technique based on the process of natural selection and genetics in evolution theory (Basu *et al.*, 2015; Blum *et al.*, 2008b). The algorithm considers a set of feasible solutions, called the population, at every iteration, called a generation. The individual solutions in the population are called chromosomes. These chromosomes evolve through successive generations in order to produce ever improving solutions.

Three operators are used to change the solutions: crossover, mutation, and selection. The crossover operator merges two chromosomes from the current generation to create one or two new chromosomes, i.e. off-spring, for the next generation. The mutation operator modifies an existing chromosome to create a new chromosome. Crossover and mutation operators produce a new chromosome with the help of existing ones, but the selection operator chooses the best chromosome between two existing chromosomes for the next generation. The number of chromosomes on which the operators are applied are chosen randomly. The termination criteria for GA include maximum number of generations, maximum number of generations without any improvement, and/or maximum computation time (Basu *et al.*, 2015; Melanie, 1999).

The simple GA is the basis on which most of the GAs are built. However, even the simple GA decisions have to be made concerning the size of the population and the values of the probabilities for crossover and mutation. The success of the algorithm depends on these chosen values. The more complicated versions of GA can work with representations of the solution that are not strings, or they may have different types of crossover and mutation operators (Melanie, 1999). Even though GA is quite restrictive in terms of its model it has been shown that GA is better suited for some optimisation problems than SA (Busetti, 1983).

3.11.2 Artificial bee colony algorithm

Particle swarm optimisation algorithms are inspired by, and aim to imitate, the biological behaviour found in swarms, colonies, or any cooperative group of living organisms. The ABC algorithm, which forms part of the particle swarm optimisation family, imitates the foraging behaviour of a honey bee colony. In a honey bee colony the foraging tasks are divided, and each task is done by a specialised group of bees that self-organise. This division of labour and organisation is essential to maximise the amount of nectar brought to the food store in the hive (Karaboga & Basturk, 2007).

Karaboga (2005) proposed the ABC algorithm for the optimisation of unconstrained numerical problems, where it tended to outperform other meta-heuristics, such as GA, differential evolution, and other particle swarm optimisation algorithms (Brajevic *et al.*, 2011; Karaboga & Basturk, 2007). The ABC algorithm can also be modified to handle continuous, combinatorial, constrained, multi-objective, and large-scale optimisation problems (Karaboga *et al.*, 2014).

In the ABC algorithm a possible solution is represented as a food source, and the amount of nectar of a food source is the quality of the associated solution. There are three groups of artificial bees: employed, onlooker, and scout bees. The artificial bee colony generally consists of 50% employed and 50% onlooker bees. Scout bees are employed bees that have become inactive. The number of solutions in the solution population is the same as the number of employed bees. Thus, each employed bee has an associated solution.

Every cycle of the ABC algorithm follows a sequence of phases aimed at finding the best possible solution (Karaboga, 2005). Firstly, during the initialisation phase every employed bee generates a solution, i.e. finds a food source, randomly. During the employed bees phase the employed bees explore the neighbourhood of their solution to find a new and better solution. This exploration is the application of a local search improvement heuristic. If a newly found solution is better than an old solution the old solution is replaced by the new solution. An employed bee will abandon the neighbourhood of its solution if no new solution is found after a number of explorations; this is called the limit. The employed bee then becomes a scout bee. The employed bees share information about the quality of their solutions and the current best solution, with the onlooker bees through a waggle dance. During the onlooker bees phase every onlooker bee chooses a solution based on the waggle dance and explores its neighbourhood using the search improvement heuristic. The probability of a solution being chosen by an onlooker bee is proportional to its quality relative to the quality of the colony's current best solution. The best solution in the colony is compared to the current best solution found so far; if an improvement is seen then the current best solution is replaced. During the scout bees phase the employed bees that have turned into scout bees randomly find new solutions and the process starts again with the employed bees phase (Karaboga, 2005). The phases are repeated until the stopping criteria is met, i.e. maximum number of iterations (Andrade & Cunha, 2015).

The variables that affect the computation length and the efficiency of the ABC algorithm are the colony size, the exploration failure limit, and the maximum number of cycles. If the solution population size, i.e. 50% of the colony size, is too small it can cause the best solution to be missed, but if it is too large it can cause the cycles to take too long. Similar arguments can be made for the other two variables (Andrade & Cunha, 2015; Karaboga, 2005).

It has been shown that ABC can be implemented to efficiently solve for unconstrained and constrained optimisation problems (Brajevic *et al.*, 2011). Also, its flexibility allows it to be easily adapted and modified to fit the needs of any optimisation problem.

3.12 Conclusion: Demand forecasting and resource deployment methods

In Chapter 3 summaries on the research conducted regarding DST, demand forecasting, and resource deployment methods were provided. Chapter 4 will start by providing a description of the WC ECC's real-world instance and describe how the chosen model is to be altered to fit the real-world instance. The decisions concerning which mathematical model and meta-heuristic to use will also be explained.

Chapter 4

The real-world instance: Western Cape emergency control centre

Summaries on the knowledge gathered and processed were provided in Chapter 3. The topics that were researched were DSTs, ambulance demand forecasting, and resource deployment methods in the EMS context. In Chapter 4 the project's real-world instance is introduced, the choice of problem model and its required alterations are discussed, and the choice of solution method to solve the problem is explained.

4.1 The Western Cape emergency control centre

Before the start of a shift, dispatchers at the WC ECC decide which holding sites to use and how many ambulances to place at each, to best cover demand for their drainage area. A drainage area is an area around a hospital. This allows for ambulances to be placed to increase the likelihood of meeting the response time targets, which are the WC ECC's service quality criteria. The calls are prioritised as either Priority 1 (P1), life-threatening, or Priority 2 (P2), non life-threatening, according to the urgency of the incident type, i.e. incident prioritisation. The response time target for urban P1 calls is 15 minutes, for rural P1 calls is 40 minutes, and no response time target is set for P2 calls. The holding site placement and ambulance allocation decisions are based only on the dispatchers' experience and intuition, since no demand forecasting mechanism or decision-making protocol is utilised.

At the WC ECC a day is divided into two shifts. A shift consists of 12 hours; the two shifts are from 7 A.M. to 7 P.M., i.e. the day shift, and then from 7 P.M. to 7 A.M., i.e. the night shift. Cross-over of dispatchers is not allowed during the shift change. This can cause an hour or more of uncertainty, as no protocol or plan explains the choices made during the previous shift or helps with the choices for the new shift. Also, the simultaneous movement of all holding sites and ambulances for the new shift may cause unnecessary delays in meeting coverage and therefore decrease the likelihood of meeting the response time targets, i.e. decrease the efficiency of the ambulance system.

The WC ECC dispatchers generally use petrol stations as holding sites, as they are public and have parking and restroom facilities. Only a certain number of ambulances and crews can be at any one holding site, as space is limited. Other options for holding sites are hospitals, the fire department, the police station, or restaurants.

The average number of available ambulances for the City of Cape Town and Cape Winelands municipalities is 70 for the day shift and 55 for the night shift. Back-up ambulances are available to replace the ambulances that are unavailable due to mechanical problems. The historical call data provided by the WC ECC for this project is from the City of Cape Town and the Cape Winelands municipalities. The data does not contain any information or provide any insight into the time-dependent speed of the ambulances, but it is known that the maximum speed is 120 km/hr. The WC ECC does use different types of ambulances, but no information is available stating what type of ambulance is used for what type of incident. Also, the closest ambulance is dispatched regardless of the priority type of the call or the type of ambulance.

The ambulance service chain of events described in Section 1.1.1 is seen in practice at the WC ambulance service, and it is a critical path for the WC ECC. The ambulance service is initiated during a shift when calls come in.

Step 1, call placement, is out of the control of the WC ECC; therefore its impact on the ambulance response time, i.e. the WC ECC's service quality criteria, can merely be observed and noted. However, the data collected from the chain of events, including that from Step 1, can be used to plan resource deployment.

During Step 2, call taking, call-takers answer the incoming calls, screen and prioritise the calls according to P1 and P2, and then assign calls to the dispatcher(s) in charge of the relevant drainage area. According to the Head of Emergency Medical Services, the WC ECC also handles non-urgent transfer calls, which make up 44-45% of the workload, but they are generally planned in advance.

After the call has been assigned to a dispatcher Step 3, ambulance dispatching, starts. The CareMonX system is used to determine the ambulance closest to the logged call. The dispatcher makes the final choice and dispatches an ambulance. The dispatched ambulance then arrives at the incident location; the on-board personnel assess the patient's state, i.e. Step 4, and determine whether the patient needs to go to hospital, and to which hospital the patient should be taken, if necessary (Schmid, 2012). On-board personnel take care of the patient until hospital personnel take over (Lee, 2012), i.e. Step 5. Hospital allocation is not considered in this research project. As soon as a patient's care is taken over by hospital personnel the ambulance is idle, and the dispatcher can send the ambulance back to its previous holding site or relocate it to another to increase overall coverage and increase the likelihood of meeting response time targets.

4.2 The location-allocation and relocation model

In Section 3.2 it was stated, that SSA will be used in the DST to forecast the probable future ambulance demand. The location-allocation and relocation model and solution method still have to be chosen. The ambulance location problem model choice was based on two factors, namely the flexibility of the model and the similarity between the previous real-world instance for which the model was used and the WC ECC's real-world instance. All three models described in Section 3.7 can be adapted to work for different real-word instances, but only the extended Q-MALP model was tested on a real-world instance of similar scale and description. Also, the problems solved for are similar. The similarities of the real-world instances are explained in Section 4.2.1 and the problem model and the changes to its formulation to fit the WC ECC's real-world instance are explained in Section 4.2.2.

4.2.1 The real-world instances: The São Paulo EMS and the WC ECC

The São Paulo EMS is public and provides out-of-hospital acute medical care, timely transport of a patient to a facility, and other medical transport of patients with serious and life-threatening illnesses and injuries. Both South Africa and Brazil are developing countries. Cape Town and São Paulo are two large cities faced with socio-economic equalities. Both cities have areas of distinct wealth and extreme poverty and ambulances have to serve all areas. The roads also range from good to bad in the respective areas for both cities (Andrade & Cunha, 2015).

In São Paulo ambulance stations are generally located in buildings, some of which are owned by the city and others rented. There are 77 ambulance stations, and for many years none have been added to the list or removed, as installing new ambulance stations can be time consuming and expensive. Building availability and rising real-estate costs have also discouraged the installation of new stations (Andrade & Cunha, 2015).

São Paulo is the largest city in South America and has kept on growing, causing the roads to become increasingly congested. It is Brazil's financial and business core and attracts a significant number of commuters daily. Traffic congestion is also no longer limited to peak time periods. This affects ambulance response times dramatically, regardless of call priority. While the ambulances are allowed to break ordinary traffic rules to reach calls as fast as possible, the constant traffic congestion causes impassable gridlocks (Andrade & Cunha, 2015). Cape Town also has a great number of commuters, which causes extensive traffic congestion on a daily basis. Ambulance drivers in South Africa are also allowed to break traffic rules, as long as they do not endanger the lives of the public while doing so.

Analysis of the ambulance station locations and vehicle allocations in 2007 led Andrade & Cunha (2015) to conclude that the best feasible response time in São Paulo was within 27 minutes, with a service reliability rate of 98% for all calls. In 2009, a comprehensive restructuring and improvement took place in order to bring the response time closer to international standards. The consensus during this process, according to Andrade & Cunha (2015), was that more ambulance stations were required in order to reduce the travel time between the ambulance station and the incident. Andrade & Cunha (2015) believed that using moveable ambulance stations – non-permanent buildings that can be set-up in public areas such as parks and squares – and relocating them at intervals to ensure good coverage would help reduce response time.

The WC ECC already makes use of public areas, specifically petrol station parking lots, for their holding sites, but they do not set-up any sort of structure and only make use of a permanent building for their ambulance depot. The ambulances only return to the depot after a shift or at the end of the day, depending on how long they are used. The holding site locations can therefore be changed during the day for every shift without considering the moving of any sort of structure.

Similar to Andrade & Cunha (2015), the focus of this project is to create an optimisation-based DST to guide decision-making to help in the choice of holding site locations, ambulance allocation, and relocation in order to improve the WC ECC's ability to accommodate varying demand of the different time periods of a shift.

4.2.2 The extended Q-MALP model for the WC ECC's real-world instance

The extended Q-MALP model was developed by Andrade & Cunha (2015) and applied to the real-world instance seen at the São Paulo EMS. The model incorporates various ideas introduced by Marianov & ReVelle (1996) and Schmid & Doerner (2010). The relevance of this model is that the modifications added to transform the Q-MALP model into the extended Q-MALP model to allow for both a stochastic and dynamic view of the problem. Therefore, the model is generic and flexible enough to deal with many real-world large EMS problems, including practical aspects of ambulance services, such as multilayer service levels, independent base location and vehicle allocation, and coverage probabilities, among others (Andrade & Cunha, 2015).

The implemented location-allocation and relocation model, for the WC ECC's real-world instance, generally follows the same formulation as the extended Q-MALP, briefly described in Section 3.7.3. However, changes had to be made for the model to fit the real-world instance. These changes are explained in Section 4.2.2.3. The extended Q-MALP model was also checked against the original Q-MALP model, created by Marianov & ReVelle (1996), and a modified and used version of the Q-MALP model, by Ghani (2012).

4.2.2.1 Assumptions and simplifications

Mathematical models may be used to represent real-world problems. These models are then solved and their solutions are used in part, or as a whole, to solve the real-world problems. These models are not 100% accurate, as the majority of the problems are complex and not all the aspects can be modelled with ease. Therefore, the majority of models require a number of assumptions and simplifications. In order to model this project's real-world instance mathematically a number of assumptions and simplifications had to be made. Some of these assumptions and simplifications have already been mentioned. The assumptions and simplifications will be provided in this section, along with those that have not yet been discussed.

The extended Q-MALP model is a multi-period model which requires a number of variables to be time-dependent. As stated in Section 3.7.1, travel time, the number of available ambulances, service time, and demand change over the course of the day for any ambulance service. Ignoring these variations and using averaged values may cause aspects of reality to be ignored. It is not always possible to get time-dependent values for all four of these. This has forced many researchers to assume stable values (averages) for some if not all of the variables (Van den Berg & Aardal, 2015). The only time-dependent variables that were available and could be used for this project were demand rate, with the help of the SSA forecasting method, and ambulance availability, with the help of queuing theory. Average values for the service time had to be used, as the service time is made up of different times that are all time-dependent, and their time-dependencies are not known in advance and cannot be determined, i.e. time to register call, time to dispatch ambulance, response time, time spent on scene, drive time to hospital, and time spent at hospital. Marianov & ReVelle (1996), Ghani (2012), and Andrade & Cunha (2015) all used an averaged service time for similar reasons when they utilised the Q-MALP model.

For this project, no data was available on ambulance travel time variations in the City of Cape Town and the Cape Winelands municipalities, therefore time-dependent travel times could not be determined. Also, the amount of data that would be required to predict ambulance travel times in advance are not generated by the WC ECC, or probably by any ECC. Therefore, an average speed will be used. The chosen average speed value is explained in Section 6.2. In Section 2.4.2, it was stated that the travel distance will be calculated using the Haversine distance method for two coordinate locations. Along with the average speed of an ambulance these distances will then be used to calculate travel time estimates.

The WC ECC ambulance fleet consists of more than one type of emergency vehicle, of which an ambulance is one. The grouping term generally used for these types of vehicles are ambulances, though the more accurate term would be emergency vehicles. The WC ECC, however, does not have accurate data concerning when what type was used and exactly how many of each is in use per shift. Since resources are limited, there are also no rules concerning which type of vehicle to dispatch to what type of emergency. Therefore, the problem had to be relaxed to assume only one type of ambulance with an average speed. Also, back-up ambulances are available, and the number of available ambulances can be assumed to stay the same for the whole of the shift, as stated in Section 4.1.

4.2.2.2 Queuing theory

The queuing theory implemented for the extended Q-MALP model was briefly mentioned in Section 3.7.3. This section will provide more information on how it was incorporated and used by Andrade & Cunha (2015) to transform the Q-MALP into a multi-period model.

Andrade & Cunha (2015) defined three neighbourhoods in order to illustrate the stochastic nature of the location-allocation and relocation problem. Since the ambulances are considered to be the same type for the WC ECC's real-world instance the neighbourhoods will be described without k, which indicated the ambulance type for Andrade & Cunha (2015)'s extended Q-MALP model. The neighbourhoods can then be defined as:

- for a given demand node $i \in D$ and a time period $t \in \tau$, W_i^t denotes the subset of all holding site nodes, $j \in H$, which can be reached from i in less than r minutes, shown in (4.1) (Andrade & Cunha, 2015);
- for a given holding site node $j \in H$ and a time period t, V_j^t denotes the subset of all demand nodes, $i \in D$, which can be reached from j in less than r minutes, shown in (4.2) (Andrade & Cunha, 2015; Ghani, 2012; Marianov & ReVelle, 1996); and
- for a given demand node $i \in D$ and a time period t, the subset of all the demand nodes, $z \in D$, which can be reached from i in less than r minutes is given by N_i^t, shown in (4.3) (Andrade & Cunha, 2015; Ghani, 2012; Marianov & ReVelle, 1996).

$$W_i^t = \{j \in H | s_{ij}^t \leq r\} \quad \forall\, i \in D, t \in \tau; \tag{4.1}$$

$$V_j^t = \{i \in D | s_{ij}^t \leq r\} \quad \forall\, j \in H, t \in \tau; \tag{4.2}$$

$$N_i^t = \left\{ z \in D | s_{iz}^t \leq r \right\} \quad \forall\, i \in D, t \in \tau. \tag{4.3}$$

The neighbourhood, N_i^t, around i is a $M/G/s - loss$ or $M/G/s/s$ queuing system (Andrade & Cunha, 2015; Ghani, 2012; Marianov & ReVelle, 1996). This type of queuing system has a Poisson distributed call arrival rate, generally distributed service times, s ambulances in the neighbourhood, and up to s calls being serviced at the same time.

This division of the area into neighbourhoods means that it is not necessary to track the state of each ambulance in the system (Marianov & ReVelle, 1996). The demand rate d_i^t in any neighbourhood i is assumed not to differ significantly from the demand rate in the neighbourhoods that border i. This proposes a rough equivalence between the calls that originate outside N_i^t that require servers inside N_i^t, and the number of calls inside N_i^t that require servers from adjacent neighbourhoods. Also, the travel times within a neighbourhood are assumed to be small compared to service times, as it is normal for the on-board personnel to have to wait for hospital personnel to become available to take over the responsibility of the patient after reaching the hospital. This allows the assumption that the flow of ambulances bound to and from N_i^t are not too different, which justifies treating each neighbourhood N_i^t as an isolated system with a set number of ambulances (Andrade & Cunha, 2015; Ghani, 2012; Marianov & ReVelle, 1996).

The aforementioned queueing assumptions make it possible to estimate the ambulance busy probability, q_i^t, in the neighbourhood N_i^t, shown in (4.4). The sum of the demand rates in a given neighbourhood, N_i^t, are comparable to a demand rate in a queuing system, λ_i^t. Similarly, the service rate, μ_i^t is one over the average duration of a call, i.e. service time. If b_i^t, which is the sum of the number of ambulances y_i^t in neighbourhood W_i^t, is equivalent to the total number of ambulances present in the neighbourhood during time period t then (4.4) can be rewritten to become (4.5) where ρ_i^t is the traffic intensity, or congestion rate, of the system (Andrade & Cunha, 2015; Ghani, 2012; Marianov & ReVelle, 1996):

$$q_i^t \cong \frac{t_{call} \cdot \sum_{z \in N_i^t} d_z^t}{24 \cdot \sum_{j \in W_i^t} y_j^t} \quad \forall\, i \in D, t \in \tau; \tag{4.4}$$

$$q_i^t \longrightarrow \frac{\lambda_i^t}{\mu_i^t \cdot \sum_{j \in W_i^t} y_j^t} \longrightarrow \frac{\rho_i^t}{b_i^t} \quad \forall\, i \in D, t \in \tau. \tag{4.5}$$

The likelihood of all servers being busy at the same time can also be estimated with the queuing theory steady-state equations. Therefore, with the given system there would be w ambulances, and the traffic intensity would be equal to ρ, so the probability can be estimated by (4.6) (Andrade & Cunha, 2015; Ghani, 2012; Marianov & ReVelle, 1996).

$$P\left(w\right)=\frac{\left(\frac{1}{w!}\cdot\rho^{w}\right)}{1+\rho+\left(\frac{1}{2!}\right)\cdot\rho^{2}+\cdots+\left(\frac{1}{w!}\right)\cdot\rho^{w}}. \tag{4.6}$$

The probability of coverage, $E(w)$, for a random call inside N_i^t can be calculated as $E(w)=[1-P(w)]$. Therefore, the incremental coverage, C_i^{wt}, gained when the number of ambulances in the system is increased from $(w-1)$ to w, is given by $[E(w)-E(w-1)]$ resulting in (4.7) (Andrade & Cunha, 2015; Ghani, 2012; Marianov & ReVelle, 1996).

$$\frac{\left(\frac{1}{w!}\right)\cdot\left(\rho_i^t\right)^{w-1}}{1+\rho_i^t+\left(\frac{1}{2!}\right)\cdot\left(\rho_i^t\right)^2+\cdots+\left(\frac{1}{(w-1)!}\right)\cdot\left(\rho_i^t\right)^{w-1}}-$$

$$\frac{\left(\frac{1}{(w-1)!}\right)\cdot\left(\rho_i^t\right)^{w}}{1+\rho_i^t+\left(\frac{1}{2!}\right)\cdot\rho_i^{t2}+\cdots+\left(\frac{1}{w!}\right)\cdot\left(\rho_i^t\right)^{w}}=C_i^{wt}. \tag{4.7}$$

All of the above make it possible to calculate the minimum number of ambulances, M_i^t, required inside neighbourhood N_i^t to provide coverage to a given demand node $i\in D$ during time period $t\in\tau$ with service reliability α. From (4.6) it can be shown that M_i^t is the smallest integer that satisfies $1-P(M_i^t)\geq\alpha$, thus (4.8) can be derived (Andrade & Cunha, 2015; Ghani, 2012; Marianov & ReVelle, 1996):

$$1-P(M_i^t)\geq\alpha\Leftrightarrow\frac{\left(\frac{1}{M_i^t!}\right)\cdot\left(\rho_i^t\right)^{M_i^t}}{1+\rho_i^t+\left(\frac{1}{2!}\right)\cdot\rho_i^{t2}+\cdots+\left(\frac{1}{M_i^t!}\right)\cdot\left(\rho_i^t\right)^{M_i^t}}\leq 1-\alpha. \tag{4.8}$$

4.2.2.3 Problem model formulation

The problem model formulation for the extended Q-MALP as implemented by Andrade & Cunha (2015) was provided in Section 3.7.3.1. This section will describe how the model was adapted to be used for the WC ECC's real-world instance.

A common assumption for this type of problem is $H \subseteq D$ (Andrade & Cunha, 2015). This is at times a valid assumption for the problem model for WC ECC's real-world instance as implemented in this project. For this project the whole area will be divided into blocks that represent the demand nodes and then again be divided into blocks that will represent the holding site nodes. A holding site node might fall within a demand node. However, this is not a mandatory assumption.

The planning horizon can be chosen to be anything from a week, a month, even a year, but the size of the planning horizon does affect the time required to solve the model. Therefore, a smaller planning horizon which would be solved in a shorter period of time is suggested. Also, with a smaller planning horizon updated historical call data can be added at shorter intervals to solve for higher accuracy. Regardless of the chosen planning horizon each day is divided into two shifts (day and night), which are then divided into 12 hours. The hours are represented by $\tau = 1, 2, 3, ..., 12$ for each shift; these are the time periods. The set of time periods represented by τ and indexed by t depend on the shift being considered. If it is the day shift $\tau = 7, 8, 9, ..., 18, 19$, and if it is the night shift $\tau = 20, 21, 22, 23, 24, 1, 2, ..., 6, 7$.

Only one type of emergency vehicle, referred to as ambulance, was considered in the implemented model. Therefore, the k variable will not be used, as it can only ever be assigned a value of 1. The ability of an ambulance to respond to a demand within r minutes is therefore kept constant for all ambulances. The decision and control variables are shown in Table 4.1 and Table 4.2, respectively. The parameters for the implemented version of the extended Q-MALP model are shown in Table 4.3.

Table 4.1: Decision variables.

Symbol	Description
x_i^{wt}	Binary decision variable that is equal to 1 if and only if demand node i is covered by at least w ambulances at time period t
y_j^t	Integer decision variable representing the number of ambulances located at holding site node j at time period t
z_j	Binary decision variable that is equal to 1 if and only if a holding site is located at node j at time period t
$r_{jj'}$	Integer decision variable representing the number of ambulances that should be repositioned from location j to location $j^{'}$ $(j, j^{'} \in H)$

Table 4.2: Control variables.

Symbol	Description
p_Z	Number of stations to be located
p_A	Number of available ambulances to be allocated
α	Service reliability value
β	Repositioning penalty

The objective function (3.17) can be divided into two parts. The first part represents the location-allocation phase, shown in (4.9), and the second part the relocation phase, shown in (4.10). The phases are solved using two different solution methods, discussed in Section 4.3. However, the relocation phase's solution method is called within the location-allocation phase's solution method. This is necessary, since the relocations are what link the location-allocation decisions made for each hour of the planning horizon, i.e. t.

$$[max] \sum_{t\in\tau}\left[\sum_{i\in D}\sum_{w=0}^{M_i^t} d_i^t \cdot C_i^{wt} \cdot x_i^{wt}\right]; \tag{4.9}$$

$$[max] \sum_{t\in\tau}\left[-\beta \cdot \sum_{j\in H}\sum_{j'\in H} s_{jj'}^t \cdot r_{jj'}^t\right]. \tag{4.10}$$

The goal of (4.9) is to maximise the expected coverage for each shift over the entire planning horizon by selecting holding site nodes and allocating ambulances, while (4.10) seeks to minimise the total time spent on relocating ambulances between consecutive hours during each shift, with β as the relocation penalty. The constraints that are purely relevant to the location-allocation phase are (3.18), (3.19), (3.20), (3.21), (3.24), (3.26), (3.27), (3.28) (3.29), and (3.30). The constraints relevant to the relocation phase are (3.22), (3.23), and (3.31). For all the constraints k is set equal to 1 and removed for the WC ECC's real-world instance.

The extended Q-MALP, created by Andrade & Cunha (2015), was verified against the original Q-MALP, by Marianov & ReVelle (1996), and a modification and implementation of the Q-MALP by Ghani (2012). It was taken into account that Andrade & Cunha (2015) made changes to the Q-MALP model in order to extend its use. The

Table 4.3: Problem parameters.

Symbol	Description
$i \in D$	Set of demand nodes
$j \in H$	Set of possible holding site nodes
$t \in \tau$	Set of time periods
$G = (V, E)$	Undirected graph of $V = H \cup D$ nodes and E edges; each edge has a corresponding travel time between the different nodes at different time periods
s_{ij}^t	Travel time between demand node i and holding site node j during time period t
d_i^t	Demand rate at demand node i during time period t
r	Coverage time of ambulance
W_i^t	Subset of holding site nodes j which can be reached from demand node i in less than r minutes during time period t
V_j^t	Subset of demand nodes i which can be reached from holding site node j in less than r minutes during time period t
N_i^t	Subset of demand nodes $z \in D$ which can be reached from demand node i in less than r minutes during time period t
q_i^t	Busy fraction of ambulances in N_i^t at time period t
t_{call}	Average call duration
λ_i^t	Total demand rate in N_i^t at time period t
μ_i^t	Service rate of ambulances in N_i^t at time period t
b_i^t	Total number of ambulances in N_i^t at time period t
ρ_i^t	Traffic intensity in N_i^t at time period t
C_i^w	Incremental coverage obtained by increasing the number of ambulances in N_i^t from $(w-1)$ to w
M_i^t	Minimum number of ambulances inside N_i^t required to provide coverage to this node at time period t with reliability α

comparison between the models was focussed on the aspects of the model which stayed true to the original Q-MALP model. Discrepancies were found for three of the constraints relevant to the location-allocation phase. The discrepancies had no impact on two of the constraints, (3.18) and (3.20), but for the third, (3.19), it caused an increased coverage standard requirement for the possible solutions.

The same discrepancy was found in constraint equations (3.18) and (3.19). Andrade & Cunha (2015) summed the number of ambulances at holding site node j during time period t by considering W_i^t, while Marianov & ReVelle (1996) and Ghani (2012) considered V_j^t. This discrepancy is believed to be due to a typing error that went unnoticed, as Andrade & Cunha (2015)'s results do not show notable errors due to the discrepancy.

Constraint equations (3.19) and (3.20) also share a discrepancy concerning the variable x_i^{wt}. Since x_i^{wt} is defined from $w = 0$, the discrepancies in constraint equations (3.19) and (3.20) could have an effect. Constraint (3.19), with $k = 1$, is meant to be summed from $w = 1$ and to ensure that demand node i is only covered M_i^t times if at least M_i^t ambulances are stationed within the given response time standard, r, from the demand node i (Ghani, 2012; Marianov & ReVelle, 1996). However, Andrade & Cunha (2015) summed (3.19) from $w = 0$ and which changed the purpose of the constraint to ensuring that a demand node i is only covered $(M_i^t + 1)$ times if at least M_i^t ambulances are stationed within the given response time standard, r, from the demand node i. The discrepancy caused the constraint to require a higher coverage level. Therefore, the solutions tended to exceed coverage standards, since x_i^{0t} will always be equal to 1. The fact that x_i^{0t} will always be equal to 1 does not cause a complication with constraint (3.20) where $w \in 1, ..., M_i^{kt}$ when it should be $w \in 2, ..., M_i^{kt}$.

It was decided to change the three constraints to match that which was used by Marianov & ReVelle (1996) and Ghani (2012). The revised constraints are shown in (4.11), (4.12), and (4.13) and are in the form that they will implemented for this project.

$$\sum_{j \in V_j^t} y_j^t \geq 1 \quad i \in D, t \in \tau; \tag{4.11}$$

$$\sum_{j \in V_j^t} y_j^t \geq \sum_{w=1}^{M_i^t} x_i^{wt} \quad i \in D, t \in \tau; \tag{4.12}$$

$$x_i^{wt} \leq x_i^{(w-1)t} \quad i \in D, t \in \tau, w \in \left\{2, \cdots, M_i^t\right\}. \tag{4.13}$$

A few of the constraints had to be relaxed for the WC ECC's real-world instance. The fact that the possible holding site locations are represented by nodes, each of which can contain more than one holding site, caused the required relaxation of constraint (3.24), the relaxed form of the constraint is shown in (4.14). The value of p_Z is also taken to equal the number of available holding site nodes. Therefore, the constraint now only ensures that no more than the available number of holding site nodes are chosen for the location of holding sites.

Holding site nodes rather than specific holding site locations were used, as no information was provided by the WC ECC regarding the locations of their holding sites, and this representation keeps the model flexible. Locations can easily be added or taken from the set of used holding site locations. This also ensures that the final decision concerning the selected holding site locations is made by the dispatcher, this is desirable as the concept demonstrator DST is intended to support, not replace, the dispatchers' discretionary decision-making.

The other constraint that was relaxed was (3.26). In its original form it ensures that all available ambulance are allocated, in its relaxed form, shown in (4.15), it ensures that no more than the available ambulances are allocated. Both forms of the ambulance allocation constraint were coded, but only one was activated at a time. The results for the two implementations will provide knowledge on whether the WC ECC's fleet is sufficient for the task of achieving a high ambulance coverage with a specified service reliability value, or whether the WC ECC would be able to cope with fewer.

$$\sum_{j \in H} z_j \leq p_Z; \tag{4.14}$$

$$\sum_{j \in H} y_j^t \leq p_A \quad t \in \tau. \tag{4.15}$$

The relaxation of the ambulance allocation constraint caused complications with the implementation of the solution method for the relocation problem, as the number of ambulances allocated during a shift do not stay constant. In order to solve this, a dummy holding site node was created to represent the ambulances not allocated to a holding site node during time period t. The Metro Emergency Medical Service

(-33.9348108, 18.4894037) was chosen as the location of the dummy node for this project. At the point that the model provides its output the dispatchers can decide where the ambulances located to the dummy holding sites should be placed. This again places the final decision in the hands of the dispatcher.

4.3 The solution methods

Since the output of this research could have ethical implications if implemented, as stated in Section 1.2.4, a robust, efficient, tried and tested solution method for the model is required. However, the two phases of the objective function each requires its own solution method, since they are two types of problems. The relocation phase's solution method will be called from within the location-allocation phase's solution method. This will be explained in Section 5.6. Therefore, the location-allocation phase's solution method is also the model's solution method, and it was determined in Section 3.8 that it should be a meta-heuristic.

4.3.1 Location-allocation solution method

As stated in Section 3.9, meta-heuristics are problem independent and consist of different components that can be implemented in many ways (Basu *et al.*, 2015). Therefore, most researchers choose a meta-heuristic, and the implementation of its components, based on its ease of use and whether it has been used to solve that type of model before (Basu *et al.*, 2015). Since Andrade & Cunha (2015) used the ABC algorithm, and the real-world instances of the São Paulo EMS and the WC ECC are comparable, a version of the basic ABC algorithm will be implemented. An overview on the ABC algorithm was provided in Section 3.11.2. The following explanation of the basic ABC algorithm provides more detail.

The basic ABC algorithm considers a population of SN solutions. These solutions are called food sources and each consists of a Z-dimensional vector. Each dimension represents a decision parameter, and for each there is a defined minimum and maximum bound (Li & Yang, 2016). Algorithm 1 shows the pseudo code for the basic ABC algorithm (Bansal *et al.*, 2013; Brajevic *et al.*, 2011; Li & Yang, 2016). In Section 3.11.2 it was stated that 50% of the population is employed bees and 50% onlooker bees. A food source, or solution, has Z decision parameters indexed by j. There are

SN number of solutions considered during one cycle. The solutions are indexed by j. Let zp^j_{min} and zp^j_{max} represent, respectively, the minimum and maximum bounds of parameter zp^j_i of food source i. Then v^j_i represents the mutated parameter j of food source i, and ϕ^j_i a random value ranging between -1 and 1. The Z-dimensional vector of food source i is represented by bee_i, and $newbee_i$ represents the mutated Z-dimensional vector of food source i. Then fit_i represent the fitness of food source i and $prob_i$ the probability of food source i being selected by an onlooker bee. $Obj(bee_i)$ is the calculated objective function value for the solution represented by bee_i.

The basic ABC algorithm shown in Algorithm 1 is for unconstrained problems. In Section 5.6.1 it will be described how the model or the ABC algorithm was adapted to solve the constrained problem considered in this project. The pseudo code also shows that the basic ABC algorithm strives to find the solution that provides the minimum value. Thus, a negative sign had to be added to the objective function, (3.17). The summed expected coverage and relocation cost values, in (3.17), are used as the value that has to be minimised by the ABC algorithm.

The relocation cost is added to the expected coverage value to ensure that the coverage lost during relocation is taken into account. The relocation cost is used to represent the loss of coverage. Also, different holding site node locations and ambulance allocations can provide the same coverage but required a higher or lower relocation cost. The relocation cost's influence on the objective function value is controlled with β, seen in (3.17), which is kept small to allow the expected coverage part of objective function to have the greatest influence.

4.3.2 Relocation solution method

The relocation phase of the problem requires its own solution method, as it is a different type of problem. The relocation problem can be defined as a minimum cost flow problem in a bipartite graph (Andrade & Cunha, 2015) and can be written as a mixed-integer linear programming problem.

In Section 3.6 it was explained that ambulance location problem models are generally defined on a graph. The minimum cost flow problem can also be defined on a similar graph. Let $Gr = (V, E)$ where V denotes the graph's vertex-set and E the set of directed arcs (Cunningham, 1976; Kitahara & Matsui, 2012). Every node in the graph

Algorithm 1 ABC algorithm pseudo code.

```
procedure ABC
    Initialise;
    for Food source $i$ until food source $SN/2$ do
        for Parameter $j$ in food source $i$ do
            $zp_i^j = zp_{min}^j + rand(0,1).(zp_{max}^j - zp_{min}^j)$;
        end for
    end for
    while Cycle $\leq MCN$ do
        EMPLOYED BEE PHASE:
        for Food source $i$ until food source $SN/2$ do        ▷ Neighbourhood search
            Randomly select parameter $j$ and a solution $k$, $i \neq k$;
            $v_i^j = zp_i^j + (\phi_i^j \times (zp_i^j - zp_k^j))$;
            if $Obj(newbee_i) <= Obj(bee_i)$ then
                $bee_i = newbee_i$;
            end if
        end for
        for Food source $i$ until food source $SN/2$ do        ▷ Fitness
            if $Obj(bee_i) >= 0$ then
                $fit_i = 1/(1 + Obj(bee_i))$;
            else
                $fit_i = 1 + |Obj(bee_i)|$;
            end if
        end for
        for Food source $i$ until food source $SN/2$ do        ▷ Probability
            $prob_i = \frac{fit_i}{\sum_{j=1}^{SN/2} fit_j}$;
        end for
        ONLOOKER BEE PHASE:
        Select food source based on prob and do Neighbourhood search;
        SCOUT BEE PHASE:
        for Food source $i$ until food source $SN/2$ do        ▷ Reinitialise
            if Food source $i$ has not improved after $trial(i)$ then
                Initialise food source $i$;
            end if
        end for
        Memorise the best solution thus far;
    end while
end procedure
```

has a capacity value, every arc an associated cost, and every node has a demand that has to be met (Goldberg, 1997).

Let dem_j be the demand at each node ($j \in V$), $c_{jj'}$ the cost associated with moving one unit from j to j' across each edge $(j, j') \in E$, and $u_{jj'}$ be the capacity of unit flow $r_{jj'}$ allowed on each edge. It can be assumed for most minimum cost flow problems, and it is a valid assumption for the relocation problem, that dem_j, $c_{jj'}$, and $u_{jj'}$ are all integers. The problem can then be formulated mathematically as follows:

$$[min] \sum_{(j,j') \in E} c_{j,j'} r_{j,j'}. \tag{4.16}$$

Subject to:

$$\sum_{j':(j,j') \in E} r_{jj'} - \sum_{j':(j',j) \in E} r_{jj'} = dem_j \quad \forall\, j, j' \in V; \tag{4.17}$$

$$0 \leq r_{jj'} \leq u_{jj'} \quad \forall\, (j, j') \in E. \tag{4.18}$$

The minimum cost flow problem seeks to find maximum flow while minimising the cost of the flow (Edmonds & Karp, 1972; Goldberg, 1997). Depending on the problem the graph can be symmetrical, where each arc has a corresponding reverse arc (Goldberg, 1997). Since, the Haversine distance method is to be implemented and the road network is not taken into account, the relocation problem is an undirected minimum cost flow problem. However, the number of relocations between two holding site nodes are limited by the maximum number of ambulances that can be assigned to any holding site.

The simplex method is often used to solve for minimum cost flow problems (Cunningham, 1976; Kitahara & Matsui, 2012). It is an algebraic procedure based on solving the systems of equations that the mathematical model consists of. It was developed by George Dantzig, who is seen as the father of linear programming, in 1946 (Hillier & Lieberman, 2010). The simplex method is an organised approach used to evaluate a feasible region's vertices. This helps to determine the optimal value of the objective function (Technopedia.inc, 2017).

The simplex method is a well researched

field and a popular solution method. Most solution programming software that is available has a simplex method already coded and ready to use. The relocations required between two consecutive time periods will be solved within the ABC algorithm with the help of Matlab's built-in dual-simplex method. The choice of programming software is explained in Section 5.3.

4.4 Conclusion: The real-world instance: Western Cape emergency control centre

The project's real-world instance was introduced in Chapter 4, along with the description of the choice of model, its formulation, and the choice of solution methods. In Chapter 5, the proposed integration of the DST into the WE ECC's operations, the required analysis of the historical call data, and the choice of software will be described. An overview on the processes that comprise the DST will also be provided.

Chapter 5

The concept demonstrator decision support tool

The real-world instance, problem formulation, and solution methods were provided in Chapter 4. In this chapter it is explained where the concept demonstrator DST would fit into the WC ECC's system if implemented, and the data analysis process and the different processes required to finally forecast and solve the problem are described.

5.1 The decision support tool

If implemented, the DST would affect the processes followed at the start of a shift and during a shift. As stated in Section 4.1 resource deployment can be planned in advance with the help of historical call data, i.e. data gathered during the ambulance service chain of events. If this planning is done well it can lead to improvements in ambulance coverage, and response time. This type of planning is a problem area for the WC ECC, as stated in Section 1.2. Therefore, it was deemed necessary to create and determine the usefulness of a concept demonstrator DST to help dispatchers to plan the deployment of resources in anticipation of probable future demand. The use of the DST should improve the management of Step 3 and thus lead to response time standards being met with a specified service reliability value.

It was stated in Section 3.7, that the integration of the concept demonstrator DST into the WC ECC's system is not part of the scope of this project. Therefore, the chosen location-allocation and relocation model was the multi-period extended Q-MALP

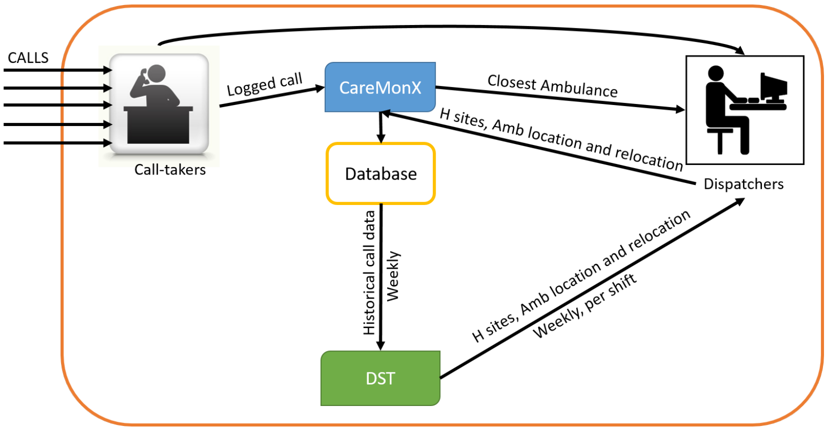


Figure 5.1: Model Interaction.

model. The DST is built to interact with the WC ECC's system as shown in Figure 5.1. The DST, if implemented, would run in parallel with the CareMonX system, since the location-allocation and relocation model is not dynamic, like the CareMonX system. Also, running the two in parallel would prevent unforeseen interactions between the system and the DST which might hinder the ambulance service.

Figure 5.1 shows that the DST requires historical call data, i.e. the data collected during the ambulance service chain of events, from the WC ECC database. Also, the DST requires the data on a weekly basis, but this can easily be changed if managers prefer to plan on a monthly or quarterly basis. Thereafter, the data is processed, demand rates for the planning horizon predicted, and the extended Q-MALP model implemented and solved. The solution is then provided, needs to be interpreted, and used to inform dispatchers' decisions concerning holding site placement, ambulance allocation, and relocation.

5.2 Analysing the historical call data

To fully understand and solve a problem data is required. The data has to provide information on the problem. The data analysis process is generally followed to ensure that the data used provides this information and is measured correctly. The first step is to determined what type of data needs to be measured and what measurement method will be used. Secondly, a method for gathering the measured data must be devised. The data then has to be analysed and filtered. In Section 5.2.1 the data collection process followed for this project is described, and in Section 5.2.2 how the data was evaluated and checked for outliers are explained.

5.2.1 Data collection

It was stated in Section 1.2.4, that the project is to be done in association with the WC ECC. The data was collected and provided by a WC ECC data analyst, and had to be picked up at Tygerberg Hospital in Cape Town. The analyst gathered six months' call data, from 1 August 2015 at 7 A.M. until 29 February 2016 at 06.59 A.M., from the City of Cape Town and Cape Winelands municipalities. The Structure Query Language (SQL) query used by the data analyst to call the data from the WC ECC database can be seen in Appendix B.

The data analyst only provided 6 months' call data for a number of reasons. First, the CareMonX system has only been in use since the end of 2014, and the system is implemented in the emergency vehicles in phases, i.e. a group of vehicles receives the new system at a time. The system is not yet implemented in all the vehicles. Therefore, the data collected using the CareMonX system has become more complete as more implementation phases have passed. The use of the CareMonX system by the dispatchers, call-takers, and on-board personnel has also improved over time, as they gained more experience, and so increased the accuracy of the collected data. A better fleet GPS tracking system, called AVL track, was also brought on-line recently. Thus, the most recent data, believed to be the most accurate, and containing at least one peak holiday season – Christmas and New Year – were provided.

The historical call data was provided in an MS Excel file (78,265 KB) and contained information on 233,391 calls from four EMS divisions which fall within the City of Cape Town and the Cape Winelands municipalities. There were 62 columns with data associated with each call, but not all were relevant for use in the DST. The columns were filtered until only 21 columns were left; 11 of these columns were used to calculate the service time for each call. The call information of interest in the DST was:

- incident number;
- priority;
- case type (emergency or IFT);

- classification (urban or rural);
- incident latitude;
- incident longitude;
- service time;
- shift start date; and
- hour sequence.

The most important aspects were the shift start date, hour sequence, incident longitude and latitude, service time, and case type, i.e. emergency or IFT, of each call. The other columns were kept to provide extra identifiable information for each call, if needed. Each row of data in the MS Excel file represented an emergency call, i.e. ambulance demand, with a priority type and case type, from an urban or rural neighbourhood, at a specific location, logged on a specific date and at a specific hour, and an ambulance that was out of service for a specified service time. The rows of data can be used to determine the demand rate per hour, per day.

5.2.2 Data evaluation

Before any further processing or calculations, i.e. forecasting, could be done, the demand rate per hour as-is had to be considered and evaluated to determine whether there are any outliers that might skew the forecasting and solution algorithm's results. Outliers are data points that lie outside of the observed data pattern (Groulx, 2007). The cause of outliers is linked to either incorrect measurements or correct measurement but rare events that form part of the data. If an outlier is caused by incorrect measurements then it has to be detected and removed, since it can adversely affect the assumptions necessary to run statistical tests and/or other calculations. Outliers that result from rare events are sometimes analysed to investigate these events and are often removed to better calculate descriptive statistics (Seo & Gary M. Marsh, 2006).

The historical data provided by the WC ECC has to be processed to gather information on demand frequency in terms of call priority, time, and locations. The demand rate per hour can easily be determined from the historical data and is the data set to be evaluated for outliers. This is necessary, since the data set will be used to calculate the demand rate per hour, per location, per priority type, which will be the input for the SSA method to forecast ambulance demand.

The SSA method was chosen for its ease of use and because it is robust. SSA also separates the trend and noise of a data set, and then uses the reconstructed time series (consisting of the trend components) to forecast. However, outliers are a form of extreme noise for which SSA might not be robust enough. According to Alexandrov (2009) and Briceño *et al.* (2013), SSA's reconstruction component, which is based on the whole time series, makes it robust enough to handle outliers. De Klerk (2015), on the other hand, stated that the presence of outliers could cause bias results, since SSA makes use of the bootstrap method and outliers should therefore be removed before using SSA. Due to the lack of consensus, a conservative approach was selected and the data set evaluated to identify potential outliers.

The data set, demand rate per hour, is univariate, i.e. it consists of only one data type. There are a number of methods available to detect outliers in univariate data sets. In general these tests are designed to detect and remove one outlier at a time, and then the process starts over until all the outliers have been removed. Three tests are usually suggested for this type of data set: Grubb's test, Dixon's test, and Rosner's test (Dan Dan & Ijeoma, 2013). Dixon's test can only be used on a data set that has fewer than or equal to 25 data points (Solak, 2009), which is not the case here.

This leaves two popular outlier detection methods, Rosner's test and Grubb's test. Rosner's test can detect up to f outliers when there are 25 or more data points. The test is able to identify high and low outliers; i.e. it is a two-tailed test. Grubb's test can be used to detect a single outlier at a time in a univariate data set. It can detect high and low outliers, but not at the same time. Both methods require the data points to be placed in order of increasing magnitude and for the mean and standard deviations to be calculated (Dan Dan & Ijeoma, 2013). Rosner's test and Grubb's test can be used on the data set, but based on inputs received from a subject-matter expert at the Centre for Statistical Consultation at Stellenbosch University, Grubb's test was chosen.

For Grubb's test, if x_n is a data point which is suspected of being a high outlier, then the test criterion, T_n, for a single outlier, as shown in (5.1), should be calculated. $\bar{x}$ is the arithmetic mean of all the data points, and SD is the standard deviation calculated with $(n-1)$ degrees of freedom (Grubbs, 1969):

$$T_n = \frac{(x_n - \bar{x})}{s}. \tag{5.1}$$

If x_1 is the value in doubt due to it being the smallest value, then T_1 for a single outlier, as shown in (5.2) (Grubbs, 1969), has to be calculated.

$$T_1 = \frac{(\bar{x} - x_1)}{s}. \tag{5.2}$$

The criterion value then has to be compared to the critical values shown in the Grubb's tables. The critical values are identified by the number of data points and the level of significance being tested for (Grubbs, 1969). However, these days Grubb's test is a built-in functionality on most statistical software.

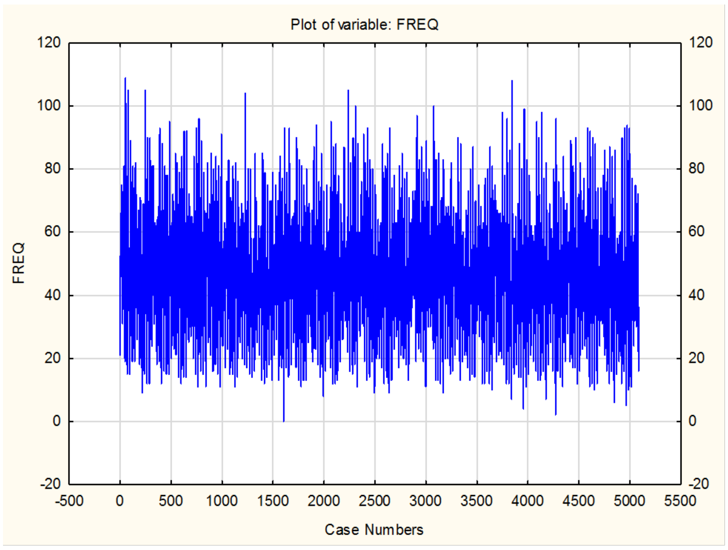
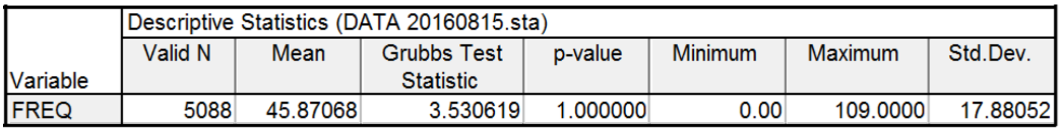


Figure 5.2: Plot of historical data per hour.

The data set was first plotted for a visual inspections for outliers. The plot is shown in Figure 5.2, and from the plot it appears that the data set does not contain outliers, but Grubb's test was still implemented to ascertain this with certainty. The university's Dell Statistica license provided easy access, therefore it was used to run Grubb's test on the demand rate per hour data set. Figure 5.3 shows the results from Grubb's test. A p-value which is equal to 1 indicates that there are no outliers in the data. This results supports the visual inspection, the data set does not contain outliers.



Variable	Descriptive Statistics (DATA 20160815.sta)						
	Valid N	Mean	Grubbs Test Statistic	p-value	Minimum	Maximum	Std.Dev.
FREQ	5088	45.87068	3.530619	1.000000	0.00	109.0000	17.88052

Figure 5.3: Grubb's test results.

5.3 Software selection

The DST was created to be a concept demonstrator and not a software package that the WC ECC can implement as-is. The DST required software that is reliable, can handle the calculations, memory and CPU usage, has tested codes, and a good help forum from which to learn the programming language.

The use of MS Excel was considered, since the historical data was provided in an MS Excel file. However, it became evident that MS Excel would not be the optimal choice for working with the data, modelling the data, and solving the problem. The file took long to open, and any changes to the data caused problems when the file had to be saved. The file crashed more than once when MS Excel struggled to implement and save changes.

To model and solve the real-world instance's problem requires matrix multiplication and manipulation. Therefore, two programs which are capable of doing both were considered: Python and Matlab. Python is open source and uses a programming language which is easy to understand. However, no codes could be found for the implementation of the basic ABC algorithm, or SSA. Matlab is a proven technical computing software application and makes use of a high-performance language. It has codes for the implementation of the ABC algorithm and SSA, which have been used and tested, and a well established help forum. Therefore, though Matlab requires a costly license, it was chosen. The cost of the license is outweighed by the availability of in-house functions and user created and tested codes. The fact that Matlab is capable of importing and exporting files from and to MS Excel easily was also a consideration. Since the historical data was provided in an MS Excel file it was decided that the concept demonstrator DST would be formulated to print the solution to MS Excel files.

5.4 The decision support tool processes

The processes that comprise the DST are called from a single '*.m'-file, *Thesis.m*. The function files of the different processes are: *ImpData.m*, *ProcessData.m*, *ProcessDemand.m*, *DemRate.m*, *SSAmain.m*, *CreateH.m*, *DistanceHH.m*, *CreateNeigh.m*, *ABCcon.m*, *AmbAllo.m*, *AlgthesisAlt.m*, and *Relo.m*. The order in which the processes are called is shown in Figure 5.4.

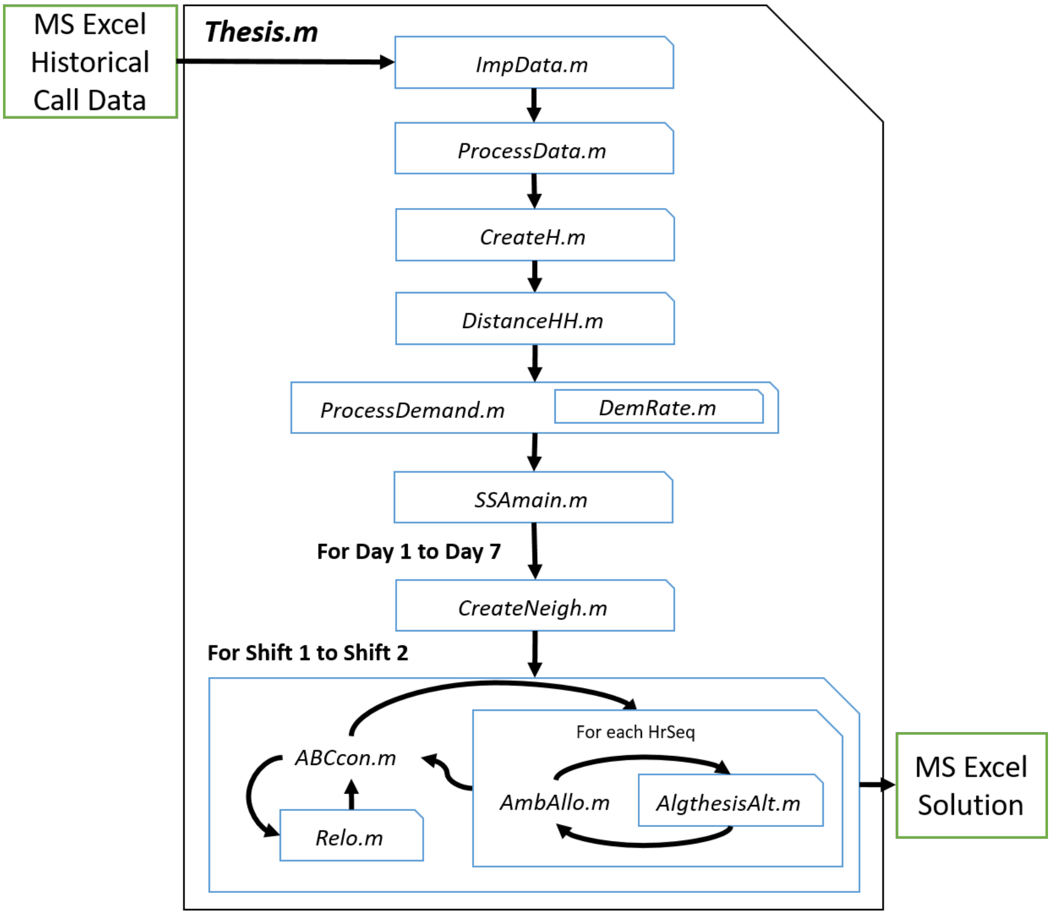
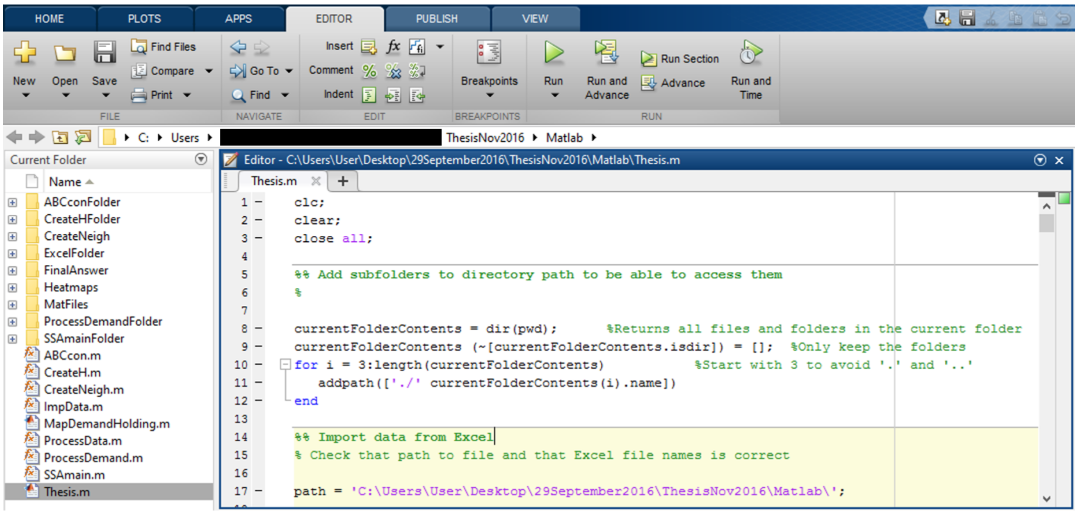


Figure 5.4: Data flow: DST.

The rest of this section will briefly explain how the DST will be used by the user. Figure 5.5 shows a screen shot of the DST's Matlab code for *Thesis.m*. As stated, the processes which comprise the DST are all called from *Thesis.m*. The '*.m'-files shown in the 'Current Folder', to the left in Figure 5.5, are the functions called directly from



```
clc;
clear;
close all;

%% Add subfolders to directory path to be able to access them
%

currentFolderContents = dir(pwd);      %Returns all files and folders in the current folder
currentFolderContents (~[currentFolderContents.isdir]) = [];  %Only keep the folders
for i = 3:length(currentFolderContents)            %Start with 3 to avoid '.' and '..'
   addpath(['./' currentFolderContents(i).name])
end

%% Import data from Excel
% Check that path to file and that Excel file names is correct

path = 'C:\Users\User\Desktop\29September2016\ThesisNov2016\Matlab\';
```

Figure 5.5: Screen shot of the DST's Matlab code for *Thesis.m*.

Thesis.m, and the rest of the functions are within one of the folders. The name of the folder that a function is in depends on what function it is called from.

The user has to determine the values of a number of variables and set the values in *Thesis.m*. The first, and most important, is *path*, which represents the address of the folder in which *Thesis.m* can be found. This address tells Matlab where to find the '*.m' files, the MS Excel file with the historical call data, and where the MS Excel solution files and the '*.mat'-files are to be saved – data can be written and saved to these files. The '*.mat'-files contain Matlab formatted data that can later be loaded into the Matlab workspace to be used in a '*.m'-file.

The first process to be called is *ImpData.m*, which imports the relevant historical call data columns, mentioned in Section 5.2.1, from the MS Excel file. The user will have to make some changes to the formatting of the MS Excel file's cells if the cells containing numbers are formatted as text instead of numbers. Within *ImpData.m* empty cells are imported with the value of NaN (Not a Number) and their values are changed to 0. The output of *ImpData.m* is a table, called 'RawData', which contains only the relevant columns of all the rows of calls from the original MS Excel file. If the historical call data from the MS Excel file had already been imported it does not have to be done again, unless the content of the file has been updated. *ImpData.m* saves the imported table to *impdata.mat*.

Before *ProcessData.m* can be called the user needs to determine whether the DST is to be run for Scenario 1 or Scenario 2 and provide the start date of the week to be solved for. Scenario 1 is when all the calls are classified as P1 regardless of the actual incident priority type, and Scenario 2 is when the calls are classified as P1 or P2 depending on the incident priority type. The reasoning behind the two scenarios are described in Section 6.1. The user also needs to state if the scenario is to be run with the relaxed ambulance allocation constraint or the original ambulance allocation constraint, by setting *alloRelax* to 1 or 0, respectively. It was decided to code the DST for the two scenarios since it became evident from the historical data that the WC ECC compares the response time of each call to the 15 minute response time target for urban P1 calls regardless of priority type and area classification. All the variable values input by the user are saved to *scInfo.mat*, which can be called by the '*.m'-files that require those variables.

The user also has to choose a value for *num*, which determines the number of demand nodes that will be created, where the number of nodes is equal to $(num-2)^2$. In *ProcessData.m* the data file *impdata.m* is loaded, and the table 'RawData' is converted to an array, called 'filterarray'. Duplicate calls were assumed to indicate that more than one ambulance was required and dispatched. The duplicates were kept to indicate better accuracy of ambulance demand. The service time of each call is calculated in *ProcessData.m* by summing the registration time, dispatching time, response time, time on scene, mission time, time to hospital, and the time at the hospital. The average service time is used as an estimate for the service time per call.

Finally, *num* is then used to created 'nodes', an array which contains the demand node number, node's longitude and latitude coordinates, and the latitude and longitude coordinates of its borders. The latitude and longitude coordinates of a node correspond to the midpoint of a square, where the square's edges are the node's boundaries. The node blocks are created by determining the emergency call located furthest to the left, right, top, and bottom, when considering the world map on paper. The coordinates of these calls are then used to created a tight square, but to encompass also the calls at the edges of the tight square a buffer of 0.01 degrees is added to create a loose square. The coordinates that indicate the corners of the loose square are then used to create a linear progression of points, *num* points, in the longitude; then the latitude coordinate system and these points are used draw the edges of the node blocks. The middle of each block is used as the location of the node. Therefore, the number of nodes are determined by the number of points, *num*, chosen for the linear regression, i.e. a 52 point linear regression would create 2,500 nodes ($(52-2)^2$). It became apparent that the number of nodes used, i.e. the size of the blocks, could adversely or beneficially influence the accuracy of the demand rate forecast per node for each hour. The number of nodes were initially taken to be equal to 2,500, then 10,000, and finally 5,625 to evaluate the effect that the number of nodes has on forecasting the demand rate per node, this is done in Section 5.5 for Scenario 1 and Section 6.3 for Scenario 2. The output of *Processdata.m* is save to *processdata.mat*.

The same process is used to create the holding site nodes array, 'hnodes', in *CreateH.m*. The user again determines the number of points, *hnum*, required for the linear regression. However, *hnum* should be less than *num* in order to create larger blocks for the holding site nodes. The coordinates of each holding site node are then checked to see whether the holding site node is on land. Only the holding sites that are on land are saved in array 'VarH'. 'VarH' has a similar form to the 'nodes' array, but it contains a column that indicates the ambulance capacity for each holding site node. The holding site nodes in 'VarH' indicate the block areas where one or more holding sites can be placed, but to leave the actual choice of holding site placement to the experienced dispatcher. This allows for their own knowledge to make the final decision, since the DST is only intended to act as support for the decision-making process and not to replace it. *DistanceHH.m* is then used to calculate the Haversine distances between holding site nodes, which is required in *Relo.m*. 'VarH', and the array with the distances between the holding sites are saved to *holdingSiteNodes.mat*.

Every logged emergency call in 'filterarray' then has to be assigned to a demand node in *ProcessDemand.m*. The calls in 'filterarray' are assigned to a demand node if their incident longitude and latitude values fall within a node's borders; the minimum and maximum longitude and latitude values indicated by the sides of the node's block. This information, along with a column that indicates whether the call is P1 and another if it is P2, is saved in an array called 'Dem'. Arrays 'DemP1' and 'DemP2' are created from 'Dem'. 'DemP1' contains only the P1 calls and 'DemP2' contains only the P2 calls. *DemRate.m* is then called from within *ProcessDemand.m* with 'nodes' and 'Dem' as input. *DemRate.m* counts the demand rate per hour per demand node for Scenario 1 and creates an array, 'VarD'. Also, *DemRate.m* creates a structure, 'SSA_VarD', where each element of the structure is an array with the demand rate per hour for a single node. The structure is in the required form for *SSAmain.m*, but the output from *SSAmain.m* will be an array of the same from as 'VarD'. 'VarD' is therefore the observed demand rate per hour per node for Scenario 1. *DemRate.m* is called two more times with 'nodes' and 'DemP1' as input to create 'P1' and 'SSA_P1', and with 'nodes' and 'DemP2' as input to create 'P2' and 'SSA_P2'. The arrays 'P1' and 'P2' are later combined to create one array and used as the observed demand rate per node for Scenario 2. 'SSA_P1' and 'SSA_P2' are used separately as input for *SSAmain.m* to forecast the ambulance demand for P1 calls and P2 calls, respectively. The predicted P1 and P2 calls are then later combined in an array used as the predicted demand rate per node for Scenario 2.

SSAmain.m uses the first four months of observed ambulance demand to forecast the last two months. The reason for this is explained in Section 5.5. Before *SSAmain.m* can be called, two control parameter values need to be chosen. The process followed to determine the best control parameter values for *SSAmain.m* when forecasting for Scenario 1 is described in Section 5.5, but the user can still decide to test it for themselves. The forecasting process followed for Scenario 2 is explained in Section 6.3. The purpose of the control parameters L and N were explained in Section 3.5. The user can assign a value to L as long as it is less than or equal to half of the number of data points used as input for *SSAmain.m*, which in this case is 3,672 data points, i.e. the hourly demand rates for the first four months of data for a single node. The forecasting is done one node at a time; then the predicted demand rates for each demand node are combined in one array, called 'ForecastVarD' when forecasting for Scenario 1. The value of N can be chosen more arbitrarily, but a good starting number is 10. When forecasting for Scenario 1 the predicted demand is saved to *forecastdemandS1.mat*, and for Scenario 2 it is saved to *forecastdemandS2.mat*.

The functions described above do not have to be repeated unless changes are made to the MS Excel file containing the historical call data, or if new user input values are chosen. The rest of the functions are run in a loop for each day of the week being solved for. This is shown in Figure 5.4.

A number of variables need to be given values before *CreateNeigh.m* and *ABCcon.m* can be called. The user needs to determine the values of the following:

- the service reliability value, α;
- the value of β, the weight variable that determines the size of the impact that the relocation phase will have on the objective function value;
- the maximum cycle number, MCN, allowed for the ABC algorithm to solve for a single shift;
- the colony size, $ColSiz$, for the artificial bee colony used in the ABC algorithm;
- the number of ambulances available during the day shift, $NumAmbDay$, which is 70 for the WC ECC's real-world instance;
- the number of ambulances available during the night shift, $NumAmbNight$, which is 55 for the WC ECC's real-world instance;
- the average ambulance speed, $ambSpd$;
- the response time target for P1 calls, $rP1$; and
- the response time target for P2 calls, $rP2$.

In Section 4.2.2.2 it was stated that the extended Q-MALP model implements queuing theory and that the stochastic nature of the location-allocation and relocation problem is illustrated by defining three neighbourhoods. The neighbourhoods are created in *CreateNeigh.m*, just before $ABCcon.m$ is called for that day, and its inputs are the predicted demand array, the shift date, 'VarHn', the response time targets, 'nodes', and the average ambulance speed. The Haversine distance method is used to calculate distances between all the nodes (demand nodes and holding site nodes), and along with the average ambulance speed a drive time estimate is determined and checked against the response time targets. The neighbourhoods are saved to $Var.mat$.

$ABCcon.m$ solves the holding site location, ambulance allocation, and relocation problem for both shifts of each day of the planning week. The ABC algorithm is repeated for a shift until the number of cycles reach the MCN value, and then it moves on to the next shift. A single cycle consists of the different bee phases shown in Algorithm 1, but changes had to be made to use the ABC algorithm for the constrained problem, which is explained in Section 5.6.1. During initialisation the employed bees solutions are initialised. It starts by randomly selecting holding site nodes in which to place holding sites for the specific shift. This is then the input to $AmbAllo.m$ where for each hour of that shift ambulances are assigned to the selected holding site nodes. The holding site node locations and ambulance allocation for that hour is then the input to *AlgthesisAlt.m*, which determines the expected coverage for this solution and whether the solution is feasible. This is sent back to *AmbAllo.m* and it is processed to become *AmbAllo.m*'s output, which is sent back to *ABCcon.m*. *Relo.m* is then called, from *ABCcon.m*, to determine the best ambulance relocation possible for the solution. In Section 4.3 it was explained that the ABC algorithm aims to minimise, and therefore a negative sign had to be added to the expected coverage value. The value that has to be minimised is the sum of the negative expected coverage and the corresponding relocation cost for that shift. Whenever changes are made to the solutions or new solutions are created *AmbAllo.m*, *AlgthesisAlt.m*, and *Relo.m* have to be called. The Matlab codes for *Thesis.m*, *ABCcon.m*, *AmbAllo.m*, and *AlgthesisAlt.m* are in Appendix F. Once *ABCcon.m* has a near-optimal solution for the day and night shift of the shift date the solutions are saved to a number of MS Excel files. Let the shift date be 2016/01/01, then the MS Excel files would be: '20160101 Performance.xls'; '20160101 Relocation.xls'; '20160101 Shift1 Demand Coverage.xls'; 'NodesDefined.xls'; '20160101 Shift1.xls'; '20160101 Shift2 Demand Coverage.xls'; and '20160101 Shift2.xls'. The first file provides data on the performance of the solution, the second provides the solution for each hour of the day, and the third defines the demand nodes and holding sites nodes in terms of their node numbers and longitude and latitude locations. The rest of the files are created to provide the user with data that can be studied for further information on the relocations and which holding site covers which demand node.

5.5 The forecasting process

As was stated in Section 5.2.1, only 6 months' call data was provided for the project by the WC ECC. This data is to be used partially as input for the SSA forecasting method and the rest to internally validate the SSA method. This method cannot be externally validated, as no access was given to a second independent sample of data. The RMSE forecasting-accuracy equation, shown in Section 3.3.2, requires historical data for the time period that is being predicted to compare and determine the accuracy of the method. Therefore, a partitioning method was implemented, where the first four months' call data was used as input for the SSA method to forecast the last two months' call data and the historical data of the last two months was used to determine the method's accuracy. The implemented SSA algorithm code is briefly described in Section 5.5.1. In Section 5.5.2 the process followed to determine the control parameter values for forecasting for Scenario 1 is explained, and in Section 5.5.3 a suggested step-by-step process for future determination of control parameter values is described.

5.5.1 The coded singular spectrum analysis algorithm

The majority of the code used in *SSAmain.m* was created by Tkachev (2014) in 2014 at MIT. The code is free to use and change, as long as the license information is provided alongside it. The code makes use of SSA easy and explains the method well. Only part of the code was used, since the only required functionality was forecasting. The SSA algorithm was explained in Section 3.5 and *SSAmain.m* follows the same basic structure.

To use the *SSAmain.m* two control parameter values have to be determined. The two values are the window length, L, and the number of reconstructed components to use for forecasting, N. As mentioned in Section 3.5, the value of L has to be less than half the data points that make up the input time series, which is 3,672 data points in this case. However, the choice of L can also improve the accuracy of the forecasting. Generally, the value of L is chosen to represent an expected periodicity. Since the time series looked at is provided in terms of hours, the expected periodicity might be daily, weekly, or monthly. The value of L could therefore be 24, 168, 720, or any of their multiples.

If the value of N is taken to be the total number of reconstructed components, then the reconstructed time series would be equal to the input time series; i.e. it will include the periodicity (trend) components and all the noise components. Therefore, the value of N needs to be chosen in order to only use the components that represent the periodicities. A scree plot can be used to determine the near best value of N. Since each node has its own time series, the possible combination for L an N became cumbersome when considering a different value for each node's time series. Also, to print and inspect the scree plot for each possible L value to gain the best N value for each node's time series would become too time consuming. A few L and N combinations were therefore tested on a number of the demand nodes' time series. The RMSE forecast-accuracy metric, described in Section 3.3.2, was used to determine the accuracy of the forecasting done with the different values of L and N. The possibility of running a simple optimisation for determining the best L and N value for each time series was tested, but the results showed large unrealistic demand predictions for a number of nodes. Therefore, it was decided to rather find a single L and N value to forecast all time series.

The accuracy of the predicted ambulance demand is important, as it directly affects the accuracy and usefulness of the solutions provided by *ABCcon.m* for the week planning horizon. The forecasting accuracy and choice of L and N values are also related to the size of the demand node blocks and therefore the number of demand nodes.

The process followed to create the demand nodes was described in Section 5.4. Initially $num = 102$ was chosen in order to create 100 nodes in a row and 100 nodes in a column, i.e. 10,000 demand nodes. The value of num was later halved to be able to see quickly how well the different processes worked. This produced 50 nodes in each row and column, i.e. $num = 52$, to create 2,500 demand nodes in total. It became evident that to fully understand the impact that node size has, the midpoint between 100 nodes and 50 nodes in each row and column would also have to be tested. Therefore, num was chosen as 77 to create 5,625 demand nodes in total. Three demand node sizes were considered: 2,500, 5,625, and 10,000 demand nodes.

The best combination of number of demand nodes and the values of L and N had to be found for both scenarios. The RMSE equation, shown in (3.1), was used to calculate the root-mean-square error between the observed and predicted values for all the data points. The RMSE values for different combinations were calculated and the combinations with the lowest RMSE values were identified. Three graphs were then plotted for a number of these identified combinations to compare the predicted ambulance demand per hour per node to the historical demand per hour per node: hourly demand rate graph, scatter plot showing the demand rate per hour per node, and hourly heat maps for the first twelve hours of 2016/01/01. This process was first tested for forecasting for Scenario 1, described in Section 5.5.2, and from there the process was refined and described for future use in Section 5.5.3.

5.5.2 Forecasting: Scenario 1

In Section 3.5 it was stated that the value of L should be equal to or a multiple of the expected periodicity and needs to be less than or equal to half of the data points. The number of data points is 3,672, i.e. the number of hours in the four months for a single demand node. Vile *et al.* (2012) provided another guideline which stated that L should also be greater than a third of the data points. However, this guideline is not present in all the literature on SSA, but the main rule which states that L should be greater than two and less than or equal to half of the data points is (de Klerk, 1994; Golyandina & Korobeynikov, 2014; Golyandina *et al.*, 2001b; Hassani, 2010). Therefore, the focus was put on the main rule, but some choices where made that took the extra guideline into consideration. The testing for Scenario 1 started with L equal to 24, 168, 720 and then later the second guideline was added and tests were done with L equal to 1,248 (24 x 52), 1,344 (168 x 8), and 1,440 (720 x 2). The value of N is associated with the L value used, and for each L value the value of N was varied.

The initial tests with L equal to 24, 168, and 720 were all run on a Dell Latitude with an Intel Core i5-5300 CPU @ 2.30GHz and 8GB RAM. Figure 5.6 shows a graph depicting the forecasting hours against the L value when the forecasting was done on the Dell Latitude for 2,500, 5,626, and 10,000 demand nodes. From the figure it can be seen that the forecasting hours increased as the value of L and the number of demand nodes increased. However, due to an accident the computer was out of service for several months. The last tests with L equal to 1,248, 1,344, and 1,440 were done on a computer with an Intel Core i7-4790K CPU @ 4.00 GHz and 32 GB RAM. The computer was able to run *SSAmain.m* much faster for the smaller L values, but for the larger L values a significant increase in memory usage caused the computer to lag. The lag only got worse with the higher number of demand nodes, specifically with 10,000 demand nodes, to the extent that the computer became unresponsive for increasingly longer periods of time. This computer's specifications are better than what the WC ECC could reasonable be expected to have available, since such a computer can be quite expensive. The assumption was made that it would be unrealistic to build or buy a computer solely for the purpose of being able to forecast with the larger L values for 10,000 demand nodes, therefore only L equal to 24, 168, and 720 were tested for 10,000 demand nodes. However, forecasting was done with the larger values of L for 2,500 and 5,625 demand nodes; it is believed that this might also be too memory intensive for most computers.

The RMSE values for the L and N combinations when forecasting with 2,500, 5,625, and 10,000 demand nodes are given in Table 5.1, Table 5.2, and Table 5.3, respectively.

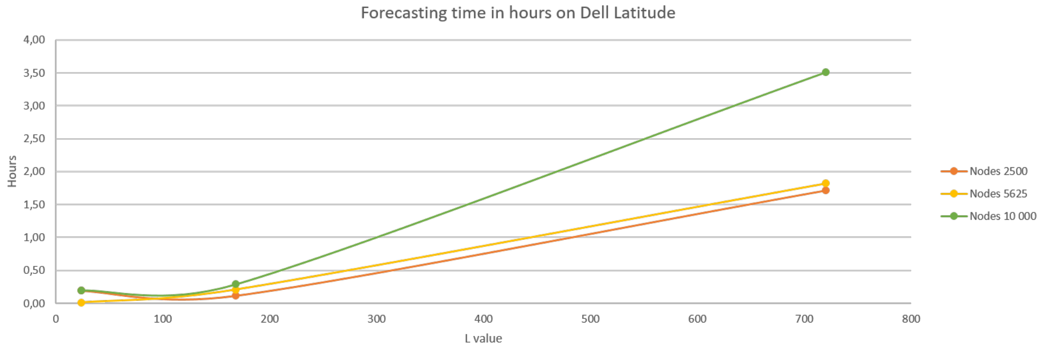


Figure 5.6: Forecasting time in hours on Dell Latitude E7450.

The tables only contain the RMSE value of the L with its best N value according to the RMSE value. L equal to 24 did not work, as it was too small and caused some of the predicted demand rates to be NaN. L equal to 168 also caused some forecasts to be NaN, and the RMSE values for 2,500, 5,625, and 10,000 demand nodes to be NaN or to be greater than 50. The only acceptable RMSE values with 2,500 demand nodes are all above one; whereas those for 5,625 and 10,000 demand nodes are below one. The reason for this might be that the demand node blocks are too large and therefore contain too much noise along with trend, to the extent that the noise overshadows the trend, or it can be that too many conflicting trends are represented by a single demand node.

Table 5.1: 2,500 demand nodes best L and N combinations.

Demand nodes	L	N	RMSE
2,500	24	18	NaN
2,500	168	12	8,727.88
2,500	720	20	1.0367
2,500	1,248	20	1.029
2,500	1,344	20	1.026
2,500	1,440	20	1.026

The fact that the RMSE values found when forecasting for 2,500 demand nodes were either NaN or greater than 50, excludes 2,500 demand nodes from further consideration. The expectation is that the best forecast accuracy will be found for either 5,625 or

Table 5.2: 5,625 demand nodes best L and N combinations.

Demand nodes	L	N	RMSE
5,625	24	20	NaN
5,625	168	8	78.70
5,625	720	8	0.817
5,625	1,248	30	0.845
5,625	1,344	20	0.823
5,625	1,440	10	0.815

Table 5.3: 10,000 demand nodes best L and N combinations.

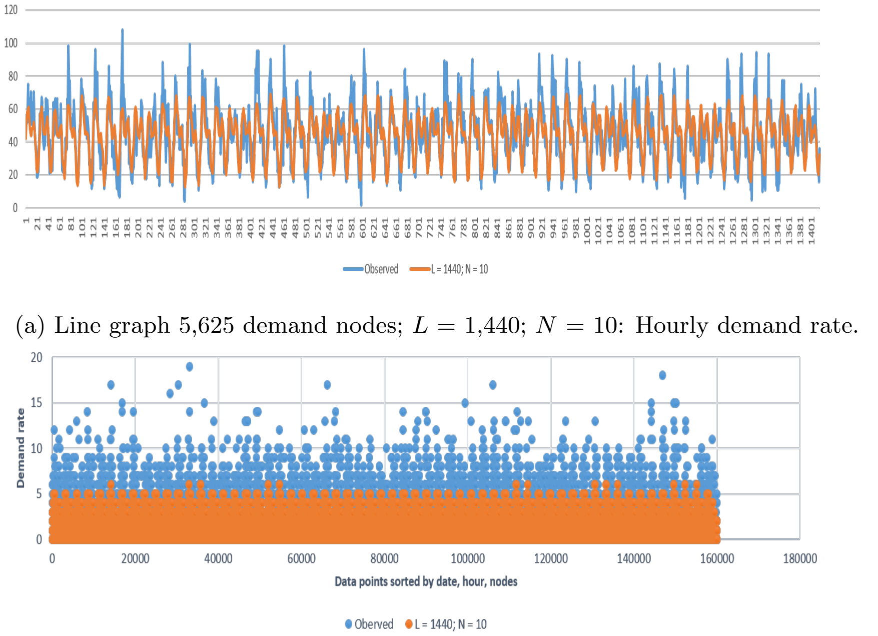
Demand nodes	L	N	RMSE
10,000	24	8	NaN
10,000	168	10	NaN
10,000	720	8	0.710

10,000 demand nodes, shown in Table 5.2 and Table 5.3, respectively. The original tables with the raw test results were reconsidered to create a table with the six best combinations of L, N, and the number of demand nodes.

Table 5.4: The six best RMSE value combinations.

Demand nodes	L	N	RMSE
10,000	720	8	0.710
10,000	720	20	0.746
10,000	720	40	0.749
5,625	1,440	10	0.815
5,625	1,440	18	0.822
10,000	720	80	0.850

Table 5.4 shows the combinations ranked from lowest to highest RMSE value, as expected the table contains no combination with 2,500 demand nodes. It is interesting to note that the L values in the table are all multiples of 720, indicating a monthly periodicity. If the RMSE value was to be the only criteria, the best combination would



(a) Line graph 5,625 demand nodes; L = 1,440; N = 10: Hourly demand rate.

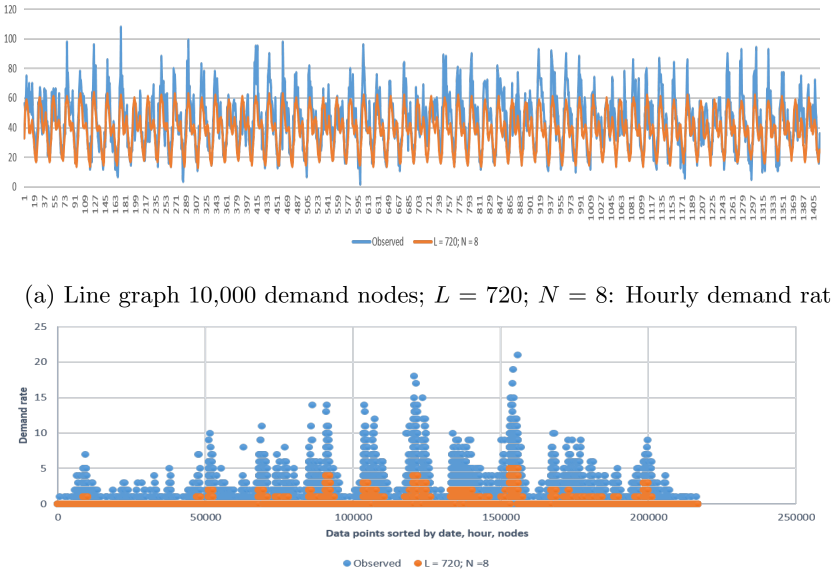
(b) Scatter graph 5,625 demand nodes; L = 1,440; N = 10: Demand rate per hour per node.

Figure 5.7: Graphs for forecasting with 5,625 demand nodes and L = 1,440; N = 10.

be 10,000 demand nodes with $L = 720$ and $N = 8$. However, it was decided that a visual criterion should also be considered, since the demand rates have both a time and location aspect. Consequently three graphs were plotted: hourly demand rate line graph, demand rate per hour per node scatter plot, and hourly heat maps for the first twelve hours of 2016/01/01. Graphs were plotted for the two combinations with the lowest RMSE values. To evaluate the impact of the RMSE as a forecast-accuracy metric for forecasting a spatio-temporal data set, graphs were also plotted for two combinations with high RMSE values. The fourth and sixth combinations, presented in Table 5.4, were chosen. The graphs were created for 10,000 demand nodes, $L = 720$, $N = 8$ and $N = 20$ and $N = 80$, along with 5,625 demand nodes $L = 1,440$, $N = 10$.

The line and scatter graphs are shown in Figures 5.7 - 5.10. The heat maps that were created for the four combinations are shown in Figures D.1 - D.12, in Appendix D. These figures are the visual representations of the predicted demand rates relative to the observed demand rates.

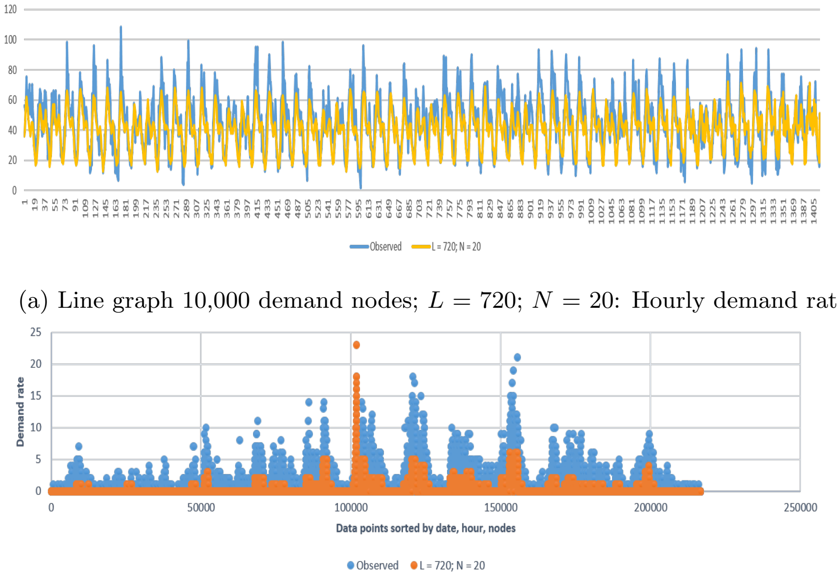
The top graph in Figure 5.7 shows that, in terms of hourly demand rate, the com-



(a) Line graph 10,000 demand nodes; $L = 720$; $N = 8$: Hourly demand rate.

(b) Scatter graph 10,000 demand nodes; $L = 720$; $N = 8$: Demand rate per hour per node.

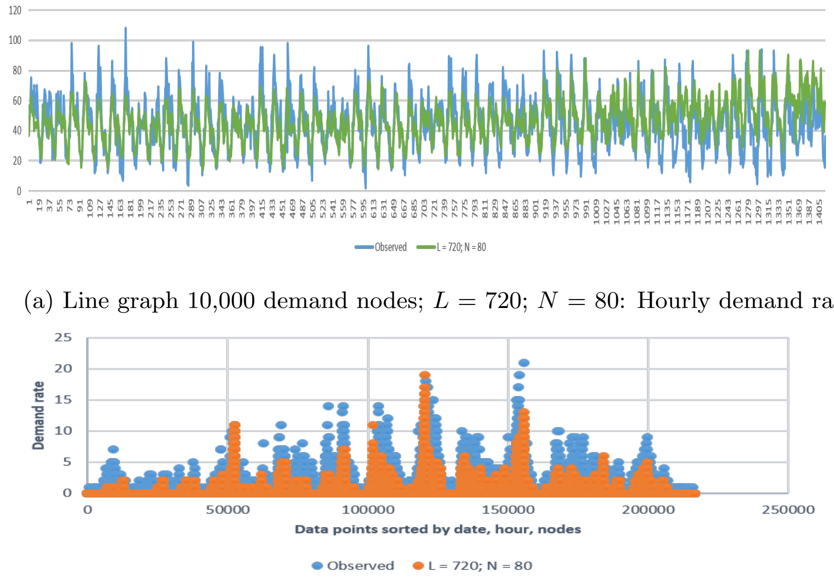
Figure 5.8: Graphs for forecasting with 10,000 demand nodes and $L = 720$; $N = 8$.



(a) Line graph 10,000 demand nodes; $L = 720$; $N = 20$: Hourly demand rate.

(b) Scatter graph 10,000 demand nodes; $L = 720$; $N = 20$: Demand rate per hour per node.

Figure 5.9: Graphs for forecasting with 10,000 demand nodes and $L = 720$, $N = 20$.



(a) Line graph 10,000 demand nodes; $L = 720$; $N = 80$: Hourly demand rate.

(b) Scatter graph 10,000 demand nodes; $L = 720$; $N = 80$: Demand rate per hour per node.

Figure 5.10: Graphs for forecasting with 10,000 demand nodes and $L = 720$, $N = 80$.

bination of 5,625 demand nodes, $L = 1440$, and $N = 10$ follows the observed demand rates for Scenario 1 well, but the bottom graph and the heat maps in Figures D.1 - D.3 show that the demand rates in terms of nodes, i.e. position, struggles to follow the observed demand rates. This is not surprising as this combination is the combination with the fourth highest RMSE value in Table 5.4.

The other three combinations all consider 10,000 demand nodes and $L = 720$. The line and scatter graphs for $N = 8$, $N = 20$, and $N = 80$ are shown in Figures 5.8, 5.9, and 5.10, respectively. The top graphs for all three figures show that the hourly demand rates follow the observed demand rates well, for all three values of N, except that at the end of Figure 5.10a the amount of noise pushes the hourly demand rate to increase too quickly. Therefore, its believed that if the graph would have been lengthened the predicted demand rate per hour would overtake the observed demand rate per hour. This excludes 10,000 demand nodes, $L = 720$, and $N = 80$ from further consideration, which is not surprising as it did have the highest RMSE value of all the combinations in Table 5.4.

The best combination is therefore 10,000 demand nodes and $L = 720$ with $N = 8$

or $N = 20$. The RMSE value for $N = 8$ is lower than for $N = 20$, but looking at Figure 5.8b it can be seen that, though the predicted demand rates follow the trend of the observed historical demand rates, it is too conservative. Figure 5.9b does contain some noise, seen just after the 100,000 data points mark, but overall it follows the trend of the observed demand rates better. The heat maps shown in Figures D.7 - D.9 also show that the positioning of the demand is similar to the observed demand. Therefore, the chosen combination for forecasting for Scenario 1 is 10,000 demand nodes, $L = 720$, i.e. monthly periodicity, and $N = 20$.

The exclusion from selection of the two combinations with the fourth and sixth highest RMSE values, in Table 5.4, due to their failure in the visual criterion shows that the RMSE is a good forecast-accuracy metric for forecasting a spatio-temporal data set. However, the chosen combination for forecasting for Scenario 1 is not the combination with the lowest RMSE value, in Table 5.4, but the combination with the second lowest due to it following the observed demand in terms of position better.

The selection of parameters for the SSA is a lengthy process, but it is necessary to ensure that the predicted demand is as accurate as can be, otherwise the DST's solution would be incapable of supporting improved emergency response service delivery. From this lengthy process it was seen that the smaller the demand node block size the higher the accuracy, but also the higher the time required to forecast. The necessity of the second L guideline (discussed at the start of Section 5.5.2) could not be proved or disproved, though better results were seen for 5,625 demand nodes with higher L values but this could not be tested for 10,000 demand nodes. Also, the forecasts for 10,000 demand nodes with the lower L values still outperformed the forecasts with for 5,625 demand nodes with the higher L values.

5.5.3 Recommended control parameter selection process

In Section 5.5.2 the process followed to determine the best combination of demand nodes and L and N values when forecasting for Scenario 1 was set out. This process has to be repeated when new observed data is added to the historical data, new time period intervals are used, or to forecast for a new scenario. Therefore, a step-by-step process for future implementation was deemed necessary.

Firstly, divide the map into demand nodes. In Section 5.5.2 it was concluded that it is beneficial to keep the demand node blocks as small as possible, since it allows for improved accuracy when forecasting demand rates in terms of time and location.

Secondly, determine what the expected periodicity is. This will depend on the time intervals used for the demand rate. If hourly time intervals are used, as is the case for this project, the possible periodicities are hourly, weekly, or monthly. The results in Section 5.5.2 showed that a good initial assumption for the real-world instance is monthly periodicity, i.e. L equal to 720. The initial choice for the expected periodicity should be substantiated by testing it against another assumed periodicity. The multiples of the expected periodicity values should be considered for L if the computing power and memory of the computer is capable of handling it, since it could not be proved or disproved that including the second L value guideline, stated in Section 5.5, leads to more accurate forecasts.

Thirdly, forecast with $L = 720$ and N equal to 10, 30, and 60. Determine the RMSE values of each and establish whether there is a trend. If, for example, it seems that a better forecast is possible with N between 30 and 60, then forecast with a value of N between those two values.

Finally, create visual representations for the three combinations with the lowest RMSE values to help in determining the best combination. Create an hourly demand line graph, a scatter plot illustrating the demand rate per hour per node, and, finally, heat maps for every hour for the first 12 hours of a day. The hourly demand line graph demonstrates visually whether the predicted demand rates follow the observed demand rates in terms of its time aspect. The scatter plot and heat maps show whether the predicted demand rates follow the location aspect of the observed demand rates. Base the selection on a combination of the RMSE value and the three graphs. This process should provide the user with a well substantiated choice of control parameters.

5.6 The solution process

The solution process comprises of a number of processes: *ABCcon.m*, *AmbAllo.m*, *AlgthesisAlt.m*, and *Relo.m*. These processes are repeatedly called to solve the model for each shift of each day of the planning horizon. The solution process firstly requires the start date of the planning horizon, then *CreateNeigh.m* is called to determine the neighbourhoods for each hour of that day, and finally the solution process is called. A for-loop allows this process to be repeated for each day of the planning horizon. This is shown in Figure 5.4 and the relationships between the processes were explained in Section 5.4. The planning horizon implemented in the DST is a week. In Section 4.2.2.3, the use of a shorter planning horizon was suggested. A shorter planning horizon allows for shorter run times and for new call data to be added to the historical call data regularly before solving for the next planning horizon. Therefore, a week planning horizon was chosen, i.e. 7 days. The planning horizon can easily be changed if desired by the user.

The location-allocation and relocation problem is a constrained problem. Therefore, either the model has to be transformed into an unconstrained problem or the ABC algorithm has to be adapted for constrained optimisation. In Section 5.6.1 it is explained that the ABC algorithm had to be adapted and how it was adapted. The implemented location-allocation phase, i.e. *AlgthesisAlt.m*, of the extended Q-MALP model is described in Section 5.6.2, and the relocation phase, i.e. *Relo.m*, is described in Section 5.6.3.

5.6.1 Constrained artificial bee colony algorithm

The ABC algorithm code in *ABCcon.m* is based on a code provided by an online academic and professional website that aims to be a resource of source codes and tutorials in the fields of Artificial Intelligence, Machine Learning, Engineering Optimisation, OR, and Control Engineering (Heris, 2015). Changes had to be implemented to use it for this project, since the coded ABC algorithm was for the basic ABC algorithm, which is used for unconstrained problems.

Constrained optimisation problems are found in structural optimisation, engineering design, economics, and allocation-location problems. The ABC algorithm was initially proposed for unconstrained optimisation problems, where it tended to outperform other meta-heuristics (Brajevic, 2010; Karaboga & Akay, 2011; Karaboga & Basturk, 2007; Karaboga *et al.*, 2014). Most optimisation algorithms were initially designed to optimise unconstrained problems, with constraint handling techniques added later in order to direct the search towards feasible regions of the search space. Koziel & Michalewicz (1999) grouped the constraint handling methods into four categories (Karaboga & Akay, 2011):

- the use of operators to change an infeasible solution into a feasible solutions in order to preserve feasibility;
- the use of a penalty term, added to the original objective function, to penalise solutions that violate constraints, thus transforming the problem from constrained to unconstrained;
- the creation and use of methods that make a clear distinction between feasible and infeasible solutions; and
- the use of hybrid methods which combine evolutionary computation techniques with deterministic procedures for numerical optimisation.

In Section 3.11.2 it was stated that either the ABC algorithm has to be adapted for constrained optimisation, or the constrained location-allocation and relocation problem had to be transformed into an unconstrained problem. The initial approach followed was to add a penalty function to the original objective function to penalise solutions that violate constraints and so transform the problem into an unconstrained problem. Thus, there was initially no need to change the ABC algorithm. A possible optimal solution for shift date 2016/01/01 was found using this method. However, changes to the ABC algorithm's control parameters did not improve or change the best solution found. It was concluded that this occurred because the addition of a penalty function forced the artificial bees into a direction with only feasible solutions, which lead to a solution population that was not diverse enough, i.e. infeasible solutions were not allowed into the solution population. Further research was deemed necessary to find a method for adapting the ABC algorithm for constrained optimisation, rather than the problem model.

Since its introduction the ABC algorithm has been adapted and modified to be used on other problems. Karaboga & Basturk (2007) modified the ABC algorithm for constrained optimisation and tested it on a number of constrained problems. Brajevic (2010) proposed improvements on the modifications implemented by Karaboga & Basturk (2007), which were also tested on several engineering constrained problems, some of which contained discrete and continuous variables (Karaboga *et al.*, 2014). A number of other modifications, which are more case specific, are: Mezura-Montes *et al.* (2010)'s novel ABC algorithm for solving constrained numerical optimisation problems;

Mezura-Montes & Velez-Koeppel (2010)'s algorithm based on the ABC algorithm to solve constrained real-parameter optimisation problems; and Stanarevic *et al.* (2010)'s modified ABC algorithm for constrained problems, which employs a memory aspect (Karaboga *et al.*, 2014).

All of the aforementioned adaptations modified the ABC algorithm and not the problem model. Therefore, it was decided to base the modifications of the basic ABC algorithm, as provided by Heris (2015), on the ABC algorithm modifications that are not problem specific, i.e. on the work done by Karaboga & Basturk (2007), Brajevic (2010), and Karaboga & Akay (2011). The modified ABC algorithm, which will be used in the DST and coded in $ABCcon.m$, will still follow the same phases as the basic ABC algorithm, seen in Algorithm 1. Similar rules will be implemented for the modified ABC algorithm for constrained optimisation as for the basic ABC algorithm: 50% of the artificial bee colony is employed bees and 50% is onlooker bees; the number of solutions is the same as the number of employed bees; and the inactive employed bees become scout bees (Karaboga & Basturk, 2007). The modified ABC algorithm for constrained problems will be explained in terms of the phases and how it differs from the basic ABC algorithm for unconstrained problems.

During the initialisation phase the solution population is randomly initialised. Each solution is a Z-dimensional vector; each dimension refers to an optimisation parameter. It would take too long to allow only initialisation with feasible solutions, and it is not always possible to produce feasible solutions randomly. Therefore, the initial solution population is not considered to be feasible. The initialisation phase assigns random values to each dimension of the solution vector, such that the value is between the lower and upper boundaries of that specific dimension (Karaboga & Akay, 2011; Karaboga & Basturk, 2007). The ABC algorithm can optimise for continuous or discrete parameters, but it is initially coded to optimise for continuous parameters. For discrete parameters the values of the dimensions are initialised as explained in Algorithm 1, but the nearest discrete value is taken as the parameter value (Brajevic, 2010). This modification was required for the project's constrained location-allocation and relocation problem model. After the initialisation phase, the employed bees, onlooker bees, and scout bees phases are repeated for a number of cycles until the MCN cycle is reached (Karaboga & Akay, 2011; Karaboga & Basturk, 2007).

During the employed bees phase, an employed bee modifies the position of the solution, i.e. food source, depending on visual information and evaluates the fitness value, i.e. nectar amount, of the new solution. In the basic ABC algorithm only one randomly chosen parameter, i.e. dimension, is modified, while the other parameters are just copied from the old solution. However, in the modified ABC algorithm a uniformly distributed random real number, R_j, between 0 and 1, is produced for each parameter j, and if R_j is less than the modification rate, MR, the parameter j is modified. If none of the parameters are modified then one parameter is chosen randomly and modified. The new solution is then compared to the old and either replaces the old solution or is thrown away; this is done by ways of a selection process. For the basic ABC algorithm a greedy selection process is usually implemented where the solution with the best objective function or fitness value is kept and the other is discarded. This process should not be used in constrained optimisation, as it will severely limit the solution space explored, since only feasible solutions would be considered (Karaboga & Akay, 2011; Karaboga & Basturk, 2007).

Karaboga & Basturk (2007), Brajevic (2010), and Karaboga & Akay (2011) suggested using Deb's (2000) rules to determine which solution to keep. Deb's (2000) rules consist of three simple heuristic rules and a probabilistic selection scheme for feasible solutions based on their fitness and infeasible solutions based on their constraint violation values. Deb's (2000) rules is a tournament selection operator, where two solutions are compared at a time with the help of the following criteria:

- any feasible solution, i.e. $violation = 0$, is preferred to any infeasible solution, i.e. $violation > 0$;
- if both solutions are feasible the solution with the better objective function value is preferred; or
- if both solutions are infeasible the solution with the smaller constraint violation is preferred.

The use of Deb's (2000) rules have been linked to a lack of diversity in the solution population, since feasible solutions are preferred to infeasible solutions. The diversity of the solution population is extremely important when designing a competitive constraint handling solution approach. Karaboga & Akay (2011) stated that solution population diversity is ensured with the ABC algorithm, since the scout bees phase allows infeasible solutions to be added to the population, and the onlooker bees phase allows for infeasible solutions to be selected probabilistically and inversely proportional to the constraint violation value. Deb's (2000) selection mechanism therefore does not interfere with the solution population diversity of the ABC algorithm.

After the employed bees have each completed the local search process, they share information on the quality, fitness, and position of their solutions with the onlooker bees through a dance. This dance is mimicked in the ABC algorithm by calculating probability values. This process is quite simple for unconstrained problems, as seen in Algorithm 1 (Brajevic, 2010; Karaboga & Basturk, 2007). In the modified ABC algorithm infeasible solutions are allowed to be in the solution population; therefore a modification is required to assign probability values for infeasible and feasible solutions. The probabilities for the constrained ABC algorithm is calculated as shown in (5.3) where $violation_i$ is the constraint violation penalty value; fit_i is the fitness value of the solution i (which is proportional to the nectar amount of that food source); and SN is the colony size (Brajevic, 2010; Karaboga & Akay, 2011; Karaboga & Basturk, 2007).

$$p_i = \begin{cases} 0.5 + \left(\frac{fit_i}{\sum_{j=1}^{SN/2} fit_j} \right) \times 0.5, & \text{if solution is feasible;} \\ \left(1 - \frac{violation_i}{\sum_{j=1}^{SN/2} violation_j} \right) \times 0.5, & \text{if solution is infeasible.} \end{cases} \tag{5.3}$$

The probability of choosing an infeasible solution is then between 0% and 50%, while the probability of choosing a feasible solution is between 50% and 100%. The chosen selection mechanism is the roulette wheel; the feasible solutions are then selected probabilistically proportional to their fitness values, and the infeasible solutions are selected probabilistically inversely proportional to their constraint violation values (Karaboga & Akay, 2011). This mimics the onlooker bees' evaluation and selection of a solution based on the information provided by the employed bees, and is the start of the onlooker bees phase. The onlooker bees also modify their selected solutions, just like the employed bees, and determine its objective function value. If it happens that the modified parameter exceeds the predetermined boundaries, it is set to an acceptable value. Generally, the parameter is made to equal the closest boundary (Brajevic, 2010; Karaboga & Akay, 2011; Karaboga & Basturk, 2007).

After the onlooker bees phase, the solutions that have not been improved for a specified number of trials ($Limit$), i.e. are not worth exploiting any more, are determined. Those solutions are then abandoned and are replaced with new solutions discovered during the scout bees phase. This is done by randomly producing a solution, like in the initialisation phase, and replacing the abandoned solution with it. Another difference in the modified ABC algorithm is that the scout bees phase is only activated at a predetermined period of cycles, and at any of its multiples. The predetermined period of cycles is called the scout production period (SPP). At each cycle that is a multiple of SPP, the solution population is checked to see whether any solution needs to be abandoned, and if there are any, then the scout bees phase will be carried out. The scout bees phase allows new, and probably infeasible solutions, to be added to the solution population, which increases the diversity of the solution population (Karaboga & Akay, 2011).

The modified ABC algorithm adds two control parameters to improve the convergence capability of the ABC algorithm for constrained optimisation problems. The parameters are the modification rate, MR, and the scout production period, SPP. Another difference between the basic and modified ABC algorithm is the use of a selection approach based on Deb's (2000) rules, instead of the normal greedy selection process. The final differences are allowing infeasible solutions into the solution population to provide diversity and assigning probability values inversely proportional to the infeasible solution's constraint violation values and so allowing them a chance to be selected during the onlooker bees phase, i.e. roulette wheel selection (Karaboga & Akay, 2011; Karaboga & Basturk, 2007).

Karaboga & Akay (2011) did not just modify the ABC algorithm but aimed to determine the range of values for the control parameters that produce the best convergence results. They tested different levels for the different control parameters and created ANON tables. These tables indicated that MR should be chosen in the range $[0.3; 0.8]$, $Limit$ in the range $[0.5 \times SN \times Z; SN \times Z]$, and SPP in the range $[0.1 \times SN \times Z; 2 \times SN \times Z]$.

The control parameters used in both the basic and modified ABC algorithm are colony size, number of employed bees, number of onlooker bees, number of scouts, and the maximum number of cycles (MCN) (Brajevic, 2010). The colony size should be in the range $[40, 100]$. The number of employed bees and onlooker bees are generally taken to be equal to 50% of the colony size respectively. The number of scout bees is dependent on the number of abandoned solutions. According to Karaboga & Akay (2011) the maximum number of cycles required is 6,000.

5.6.2 Location-allocation phase

The extended Q-MALP model was briefly described in Section 3.7.3. The real-world instance and the requisite changes to the model were explained in Chapter 4. *Algthesis.m* contains the coded algorithm of the location-allocation phase of the extended Q-MALP model, but the possible solutions are initialised and manipulated in *ABCcon.m* and *AmbAllo.m*. The holding site selection, and later manipulation of the selection, is done in *ABCcon.m*, which is to be the same for an entire shift and is sent to *AmbAllo.m* where ambulance allocation is initialised and manipulated. The decision was made to hard code the integer and binary constraints into *ABCcon.m* and *AmbAllo.m*, since they are hard constraints. If these constraints are violated the solution cannot possibly be feasible. This decision excludes the possibility of a solution that requires 1.5 ambulances to be allocated to a holding site node which is only 49% selected.

AlgthesisAlt.m contains the coded location-allocation phase of the extended and changed Q-MALP model, as explained in Section 4.2.2. It calculates the values of the queuing theory parameters and the model parameters for the specific possible holding site node locations, as provided by *ABCcon.m* for that shift, and possible ambulance allocation solution, as provided by *ABCcon.m* for an hour of that shift, and then calculates the expected coverage, if that solution were to be implemented. In the final step, the possible solution is checked to see whether any constraints are violated.

In Section 4.2.2.3 it was stated that two constraints were relaxed. The relaxation of constraint (3.24) was required to accommodate the representation of the placement of holding site locations by holding site nodes, where each node can contain more than one holding site. The relaxed constraint is shown in (4.14). Constraint (3.26) was also relaxed and became (4.15). To test the impact of the number of ambulances allocated, constraint (3.26) was relaxed. *AlgthesisAlt.m* was coded with both the original ambulance allocation constraint, (3.26), and the relaxed ambulance allocation constraint, (4.15), but only one can be activated at a time. Each scenario will be solved with the relaxed ambulance allocation constraint and then with the original ambulance allocation constraint.

5.6.3 Relocation phase

In Section 4.3.2 it was stated that the relocation phase of the extended Q-MALP model can be seen as a minimum cost flow problem, which is generally solved with the simplex method. Matlab has a built-in simplex method which is easy to use when the problem is written in the required form. In Section 4.2.2 it was explained that a dummy holding site had to be added for the minimum cost flow problem to be solved. The dummy holding site balances the transportation problem equations and ensures that the number of ambulances allocated during every hour of a shift stays the same. The dummy station was taken to represent the ambulance depot, but the dispatchers can place the ambulances located by the model to the dummy site wherever they believe they will be the most beneficial. The ambulance depot used in the model was taken as the Metro Emergency Medical Service, Western Division, in Ndabeni.

5.7 Conclusion: The concept demonstrator decision support tool

In Chapter 5 how the DST fits into the WC ECC's system and the choice of software for use in this concept demonstrator were explained. The flow of data through *Thesis.m* were described and the different processes were explained. In Chapter 6 the two scenarios for which the DST were run are described. The solutions and results for running each scenario with the relaxed and the original ambulance allocation constraint, i.e. the instances, for the scenario-specific predicted demand are provided. The results for applying the instances' solutions to the scenario-specific observed demand are provided and compared and the best instances identified. Recommendations for the future use of the DST are explained and the validation and verification process for the different processes of the DST are described.

Chapter 6

Scenario analysis

In Chapter 5 where the DST fits into the WC ECC's system, the analysis process required for the historical call data, and the programming software choice made were explained. The flow of data through the DST and the DST's forecasting and solution processes were also described. In this chapter the two scenarios and two ambulance allocation constraint implementations for which the DST was solved are described, i.e. the four instances. The two scenarios are to evaluate the need for resource deployment planning according to incident type. While the two ambulance allocation constraint implementations are to evaluate whether the WC ECC requires more ambulances. The results for applying the DST's instance-specific solutions, created for the scenario-specific predicted demand, to the scenario-specific observed demand are provided and compared. Thereafter, the best instance for each scenario for the WC ECC's real-world instance is determined.

6.1 The four instances

As stated in Section 4.1, the WC ECC has response time targets for P1 calls according to area, i.e. urban or rural. The urban P1 response time target is 15 minutes and the rural P1 response time target is 40 minutes. The WC ECC does not have a response time target for P2 calls. The historical data, however, showed that the WC ECC checks the response times of all calls against the urban P1 target, regardless of incident type or area. Also, the number of rural calls for the City of Cape Town and the Cape Winelands municipalities are so small as to be negligible. The decision was made to reclassify the rural calls as urban calls, instead of leaving out the calls. This inclusion provides a better overall idea of the ambulance demand.

The incident classification and the incident type and area specific response time targets are indicative of an ambulance system that is moving away from the "golden hour" philosophy, and yet the response times of all calls are checked against the urban P1 response time target which is in line with the philosophy. To evaluate the utilisation of the "golden hour" philosophy and the incident type and area specific response time targets in terms of ambulance efficiency, the DST was set up for two scenarios. Both scenarios look at selecting holding site nodes, allocating ambulances, and relocating ambulances in order to simultaneously maximise the expected coverage and minimise relocation cost, but the call classifications and response time targets differ. Scenario 1 considers all calls to be P1 and tests their response times against the 15 minute target. Scenario 2 considers P1 and P2 calls separately and checks the response times of P1 calls against the 15 minute target and the response times of P2 calls against a 30 minute target. Both scenarios assume that all calls are from urban areas.

Section 2.4.1 provided the response time targets implemented in other countries. The response time targets show that the countries all strive toward low response times, but that there is no consensus in terms of the targets set or whether the targets should be case and/or area specific. The WC ECC's response time target for urban P1 calls is higher than those for the highest category of urban calls in the other countries mentioned in Section 2.4.1. The concept demonstrator DST is created to help with the dispatcher's decision-making process, but it has to take into account that even if no target is set for P2 calls they still need to be reached in as short a time as possible without neglecting P1 calls. In Section 2.4.2 it was stated that incident type prioritisation is difficult and usually impossible to do accurately, which can cause P1 calls to be classified as P2 or vice versa. The DST, therefore, required a target for P2 calls to provide solutions with a better representation of the spread of ambulances required to cover calls. The P2 target has to be lower, i.e. longer, than the P1 target, but it has to take into account that classification errors are possible and, therefore, the target was chosen to be 30 minutes.

In Section 1.1.2 it was mentioned that fleet size, holding site placement, and ambulance allocation are the critical factors that ECC managers try to control in order to improve ambulance efficiency. The improved management of these factors, with the help of ambulance location models, can improve the efficiency of the overall ambulance service system. This is exactly what the concept demonstrator DST aims to do with two of the three factors, namely holding site placement and ambulance allocation.

In Section 4.2.2.3 the relaxed ambulance allocation constraint for the WC ECC's real-world instance was explained. The original ambulance allocation constraint ensures that all available ambulances are allocated to chosen holding site nodes, but this was relaxed only to ensure that no more than the available ambulances are allocated. This relaxation was done to make it possible to check whether the WC ECC requires more ambulances to achieve the desired coverage or whether they have enough. To facilitate the testing of the ambulance fleet, the ambulance allocation constraint was implemented in its relaxed form, (4.15), and in its original form, (3.26), for both scenarios. This created four instances that had to be solve for:

- Instance 1: Scenario 1 with the relaxed ambulance allocation constraint;
- Instance 2: Scenario 1 with the original ambulance allocation constraint;
- Instance 3: Scenario 2 with the relaxed ambulance allocation constraint; and
- Instance 4: Scenario 2 with the original ambulance allocation constraint.

In Section 4.2.2.3 it was also mentioned that the holding site node selection constraint, (3.24), was relaxed. This was done, and the resulting solutions for the four instances required more than 78% of the possible holding site nodes to be selected per shift, and some holding site nodes' ambulance capacity restrictions were also exceeded. It became evident that by using the available number of holding site nodes as the maximum number of holding site nodes that can be selected was not going to work, it allowed a too large solution space for which there is not enough time to fully inspect. This outcome could be mitigated by implementing a cost function which would associate a cost to the use of a holding site node or to add a user-selected variable which limits the maximum number of holding site nodes that can be selected. If the first was used then its impact on the entire model would have to be considered, but the second requires only that a user-selected variable $NumSelect$ be created. The relaxed holding site node selection constraint was still implemented, but the maximum number of holding site nodes that can be selected is now controlled by the user of the DST.

6.2 User-selected variables

In Section 5.4 it was indicated that the DST requires a number of user-selected variables to function. The user-selected variable values for the two scenarios and two implementations of the ambulance allocation constraint, i.e. the four instances, are:

- $Scenario$ is equal to 1 for Scenario 1 and 2 for Scenario 2;
- $alloRelax$ is equal to 1 if the relaxed allocation constraint, and 0 if the original allocation constraint, is to be implemented;
- $hnum = 30$;
- $hcapacity = 10$;
- $NumSelect = 15$;
- $\alpha = 0.95$;
- $\beta = 0.015$;
- $MCN = 1,000$;
- $ColSiz = 40$;
- $NumAmbDay = 70$;
- $NumAmbNight = 55$;
- $ambSpd = 60$ km/hr;
- $rP1 = 15$ minutes; and
- $rP2 = 30$ minutes, is not used when $Scenario = 1$.

There are three other user-selected variables that are not in the aforementioned list: num, L, and N. The values of these variables are scenario-specific. The required values of these variables for Scenario 1, which provided the best forecasting accuracy, were determined in Section 5.5.2 to be $num = 102$, $L = 720$, and $N = 20$. The values required for high forecasting accuracy for Scenario 2 is provided in Section 6.3.

The first two variables indicate which instance is to be implemented, the rest are either specific to the real-world instance being solved for, or are the control parameters for the implemented ABC algorithm. The variables specific to the real-world instance are: $hnum$, $hcapacity$, $NumSelect$, α, β, $NumAmbDay$, $NumAmbNight$, $ambSpd$, $rP1$, and $rP2$. The user is not allowed access to all of the ABC algorithm's control parameters, because some of the variables are influenced by the values of MCN and $ColSiz$. The DST was designed for users that are most likely not proficient operations researchers, as such, it was deemed an unnecessary risk to allow the user to change the values of variables that are dependent on other variables.

In Section 5.4 it was stated that $hnum$ should be less than num to allow for larger node blocks. Therefore, it is impacted by the chosen value of num. The decision was made to use the same $hnum$ value for Scenario 1 and Scenario 2, which means that all four instances have the same number of, and locations for, their possible holding site nodes. This allows for better comparison between the results of applying the instances' solutions to the scenario-specific observed demand. A good value for $hnum$, based on Scenario 1's $num = 102$, is 30. This created a grid with 784 possible holding site nodes, of which only 28 were deemed viable options, i.e. they are on land.

The ambulance capacity of the holding site nodes was set to ten, because a petrol station holding site, which is generally what is chosen by the WC ECC dispatchers, would on average be able to accommodate three ambulances. Therefore, if ten ambulances were allocated to a holding site node then at least three holding sites would need to be located in that node. A value had to be assigned to $NumSelect$, but if the value was chosen to be too small or too large it would negatively influence the size of the solution space. The value of $NumSelect$ was chosen to equal 15 as it is just more than half of the available holding site nodes.

South Africa does not have a service reliability target level. The service reliability value most often implemented by other countries is 95%, as seen in Section 2.4.1. It was therefore decided to run the DST with a service reliability value of 95%, i.e. $\alpha = 0.95$. The β variable determines the impact of the relocation cost on the overall objective function value. Since the main focus of the objective function is to maximise the expected coverage, the value of β was chosen to be 0.015 to keep the impact of the relocation cost on the objective function value small. This was chosen after it was determined that the magnitude of the relocation cost per shift tended to be in the hundreds.

The MCN and $ColSiz$ variables are two of the control parameters of the ABC algorithm. According to Karaboga & Akay (2011), a good value for MCN is 6,000.

Taking into account that $ABCcon.m$ is solved per shift, the actual number of cycles that would be required to solve for one day would be 2,000 if $MCN = 1,000$, which was observed to take approximately 20 hours to finish. Therefore, if MCN were to be 6,000 $ABCcon.m$ would require about 42 days to solve for the week planning horizon. It was determined that Matlab can be opened multiple times and run simultaneously which would decrease the run time to seven days, but it is CPU and memory intensive. Also, the user would have to ensure that the loaded '*.mat'-files and the user-selected variable values are the same for each opened Matlab to ensure that the instance being solved for is the same.

The ABC's artificial bee colony size is indicated by $ColSiz$. The value of $ColSiz$ was chosen to equal 40. This is at the low range of Karaboga & Akay (2011)'s indicated good range for $ColSiz$. It was chosen in that manner to not overly increase the required run time, which is already heavily influenced by the value of MCN. The other control parameter values, for those not directly determined by the user, are:

- $Limit$ is equal to $(0.6 \times ColSiz \times Z)$, with Z equal to 28, i.e. the holding site nodes available;
- MR is equal to 0.7; and
- SPP is equal to $(0.1 \times ColSiz \times Z)$.

$NumAmbDay$ and $NumAmbNight$ are the number of ambulances available for the day shift and the night shift, respectively. In Section 4.1 it was stated that the WC ECC has 70 ambulances for the day shift and 55 for the night shift. The WC ECC does not collect the data required to determine the average ambulance speed. However, it is known that the maximum speed of an ambulance is 120 km/hr and that ambulances are allowed to violate traffic rules in order to arrive at an incident in the least amount of time, but it would not be possible to drive 120 km/hr around corners and through dense traffic. The average speed was chosen to be 60 km/hr, which is merely an estimated driving speed and has to be considered as such. The last two variables are the response time targets for P1 and P2 calls, i.e. 15 minutes and 30 minutes respectively. If $Scenario$ is equal to 1 then $rP2$ is not used, but if it is equal to 2 then $rP1$ and $rP2$ are used.

6.3 Forecasting: Scenario 2

The process described in Section 5.5.3 was followed to determine the best combination of the number of demand nodes, L, and N for forecasting for Scenario 2. It was determined in Section 5.5.2 that it is beneficial to keep the demand node blocks as small as possible. Therefore, the *num* value was initially chosen to be 102 to create 10,000 demand nodes. The next step was to determine the expected periodicity and to determine whether $num = 102$ is a good choice for forecasting for Scenario 2.

The results from Section 5.5.2 showed that a monthly periodicity is a good assumption for the real-world instance's Scenario 1. Therefore, it was decided to start with an assumed monthly periodicity, i.e. $L = 720$, and the low RMSE values in Table 6.1 show that it and the $num = 102$ were good choices. Table 6.1 contains the RMSE values for forecasting P1 and P2 demand rates with 10,000 demand nodes and $L = 720$ with varying N values. A big difference between the forecasting for Scenario 1 and Scenario 2 is that for Scenario 1 only one RMSE value had to be checked, whereas for Scenario 2 there is a RMSE value for the P1 forecasting and another for the P2 forecasting.

Table 6.1: 10,000 nodes best L and N combinations for P1 and P2 calls.

	Nodes	**L**	**N**	**RMSE**		**Nodes**	**L**	**N**	**RMSE**
P1	10,000	720	5	0.445	**P2**	10,000	720	5	0.605
	10,000	720	10	0.455		10,000	720	10	0.607
	10,000	720	20	0.463		10,000	720	20	0.617
	10,000	720	30	0.466		10,000	720	30	0.633
	10,000	720	40	0.496		10,000	720	40	0.647
	10,000	720	60	1.135		10,000	720	60	0.704

If only the RMSE values were to be used to determine the best N value, then $N = 5$ would have been chosen. However, this does not take the location aspect of demand into account as much as the time aspect. Line graphs and scatter graphs were created for the P1 and P2 forecasting, as was done for Scenario 1 in Section 5.5.2, with N = 5, 10, 20, and 30. These graphs can be seen in Appendix E, along with the heat maps for P1 and P2 when $N = 30$. The line and scatter graphs showed that the best N for forecasting for Scenario 2 was 30, and this was confirmed by the heat maps for the first 12 hours of 2016/01/01, with P1 and P2 forecasting with $N = 30$.

6.4 The instance specific results

The DST's solution process, explained in Section 5.6, was used to determined the near optimal solution for the four instances based on the scenario-specific predicted demand rates. The results are shown in tables in Appendix G, in the order of the instances. The results were compiled from the MS Excel files, created for each day of the planning horizon, which are the output of *ABCcon.m*. The instances' solutions were then applied to the scenario-specific observed demand rates, as determined from the historical call data. This is done to validate and compare the results in order to determine the best instance for each scenario implementation for the WC ECC's real-world instance.

Table 6.2 contains the percentage expected coverage and relocation cost results for the entire planning week, 2016/01/01 - 2016/01/07, when each instance's solution was applied to the the scenario-specific predicted demand and the scenario-specific observed demand. The planning horizon's observed demand rates did not form part of the input data used to forecast the demand rates for the planning horizon. The results in Table 6.2 show that for the four instances' solutions the percentage expected coverage for the scenario-specific observed demand is slightly lower than for the corresponding scenario-specific predicted demand. Since, each instance's solution was created for the predicted demand this is not an unexpected outcome. The fact that the difference in expected coverage is small, as seen in the 'delta' column in Table 6.2, is an indication that the forecasting and model results can be trusted.

Table 6.2: The percentage expected coverage and relocation cost for the planning week.

	% Expected coverage for the week			**Relocation cost**
	Predicted	**Observed**	**Delta**	**Both**
Sc1 Relaxed	94.63%	92.96%	1.68%	557.27
Sc1 Original	91.39%	89.60%	1.80%	1,446.01
Sc2 Relaxed	96.62%	94.99%	1.64%	399.69
Sc2 Original	95.46%	93.97%	1.49%	1,479.45

Table 6.3 shows the average ambulance usage per hour for the day shift and the night shift for the week for each instance's solution. The ambulance usage is purely dependent on the solution and stays the same for the scenario-specific predicted and

observed demand. The average value is used, since the number of ambulances in use per hour of a shift varies.

Table 6.3: Average hourly ambulance usage for the planning week.

	Average hourly ambulance usage	
	Shift 1	**Shift 2**
Sc1 Relaxed	49.79	44.15
Sc1 Original	69.58	64.94
Sc2 Relaxed	43.10	30.07
Sc2 Original	65.94	64.39

The integer values in Table 6.4 indicate whether the instances' solutions, for the planning week, violate any constraints for any of the hours of the 14 shifts, this is shown first for the scenario-specific predicted demand and then for the scenario-specific observed demand. The binary and integer constraints were hard coded in the DST, as they are hard constraints. Therefore, the only constraints that could have been violated were constraint equations (3.18) - (3.23), (4.14), (3.26), and (3.27). The relaxed ambulance allocation constraint, (4.15), could also have been violated when Instance 1 and Instance 3 were solved, but it was not.

The values in Table 6.4 were calculated by counting the violation values for each shift of an instance, where the violation values per shift were the summed violation values per hour. The values, however, do not indicate the number of times a constraint is violated during an hour or how large the violation is. Consider the possibility that for a shift the original ambulance allocation constraint, (3.26), was violated during hour sequence 1 and 3, and the holding site node capacity constraint, (3.27), was violated for a number of nodes during hour sequence 1, and no other constraints were violated during these or any of the other hour sequence of that shift, then the violation value for that shift would be three.

Table 6.4: Constraint violations for the planning week.

	Violations	
	Predicted demand	**Observed demand**
Sc1 Relaxed	0	37
Sc1 Original	153	182
Sc2 Relaxed	0	51
Sc2 Original	142	185

6.5 Scenario 1

As stated in Section 6.1 Scenario 1 considers all calls to be P1 calls and implements the 15 minute response time target for each call. In Section 6.5.1 and 6.5.2, respectively, the results for applying Instance 1's solution and Instance 2's solution to Scenario 1's predicted demand and observed are provided. In Section 6.5.3 the instance-specific results for Scenario 1's observed demand are compared and the best instance determined.

6.5.1 Instance 1

The shift results for Instance 1's solution, i.e. Scenario 1 with the relaxed allocation constraint, based on the scenario-specific predicted demand rates are shown in Table G.1. The percentage expected coverage for each of the 14 shifts, i.e. two shifts for every day of the planning week, range between 88.24%, for 2016/01/01's night shift, and 95.97%, for 2016/01/06's night shift. The solution does not violate any constraints for any of the shifts for the scenario-specific predicted demand rates. The overall expected coverage for the week is 94.63% with a relocation cost of 557.27 and the ambulance usage per hour is 49.79 for the day shift and 44.15 for the night shift. Instance 1's solution is a feasible solution for Scenario 1's predicted demand.

Instance 1's solution was then applied to Scenario 1's observed demand rates, the results for each shift are shown in Table G.3, and the percentage expected coverage ranges from 88.03%, for 2016/01/01's night shift, to 95.51%, for 2016/01/06's night shift. The percentage expected coverage result for the planning week when Instance 1's solution is applied to the scenario-specific observed demand is 92.96% with the same relocation cost and ambulance usage as seen for the scenario-specific predicted demand. The percentage expected coverage for the week for the observed demand is not as high as for the predicted demand, but that is to be expected. The scenario-specific predicted demand rates are merely a possible future, whereas the scenario-specific observed demand rates are the actual future. This does not indicated that the solution is invalid.

Instance 1's solution when applied to the scenario-specific observed demand rates does, however, violate constraints during each shift, but an inspection showed that only one constraint was violated throughout, (3.18). Therefore, Instance 1's solution caused one or more of the scenario-specific observed demand nodes to be uncovered. The percentage of scenario-specific observed calls that are not covered by Instance 1's solution is 0.67%. In terms of the bigger picture this shows that Instance 1's solution is a valid and viable solution for the WC ECC's real-world instance. Even though constraint (3.18) is violated, Instance 1's solution is feasible for the scenario-specific observed demand.

6.5.2 Instance 2

The shift results for Instance 2, i.e. Scenario 1 with the original ambulance allocation constraint, for the scenario-specific predicted demand is shown in Table G.2. The values of the percentage expected coverage ranges from 84.15%, for 2016/01/04's night shift, to 94.29%, for 2016/01/04's day shift. It is interesting to note that the overall percentage expected coverage with the original ambulance allocation constraint is 91.39%, which is lower than for Instance 1's solution applied to the scenario-specific predicted and observed demand. The average ambulance usage per hour is 69.85 for the day shift and 64.94 for the night shift, and the relocation cost is 1446.01. The average ambulance usage per hour and relocation cost are higher for Instance 2 than for Instance 1, but this is due to the implementation of the original ambulance allocation constraint.

The solution found for Instance 2, based on the scenario-specific predicted demand, does violate a number of constraints for a number of hour sequences, as seen in Table 6.4. It was determined that 1.31% of the violations were due to one or more of the scenario-specific predicted demand nodes being uncovered; 28.10% were due to fewer ambulances being allocated than were available; 65.36% were due to more ambulances being allocated than were available; and 5.23% were due to one or more holding site node capacities being exceeded. The number of scenario-specific predicted calls which cannot be considered covered by the solution are two, which is 0.03% of the calls. The violations that are due to more ambulances being allocated than were available and holding site node capacity being exceeded, makes Instance 2's solution infeasible for Scenario 1's predicted demand.

Though it was determined that Instance 2's solution for Scenario 1's predicted demand was infeasible, the solution was still applied to Scenario 1's observed demand to determine if it provides higher expected coverage percentages than Instance 1's solution applied to Scenario 1's observed demand. If it does, it could indicate that the holding site node capacities need to be increased and that more ambulances could be required.

The shift results for Instance 2's solution applied to the scenario-specific observed demand are shown in Table G.4. A greater number of shifts have percentage expected coverage values below 90%, which was not the case when Instance 1's solution was applied to the scenario-specific observed demand. The percentage expected coverage ranges from 82.15%, for 2016/01/04's night shift, to 93.88%, for 2016/01/04's day shift. The overall expected coverage for the week is 89.6%, which is also lower than when Instance 1's solution was applied to Scenario 1's observed demand. The average ambulance usage per hour and relocation cost are the same as when Instance 2's solution was implemented for Scenario 1's predicted demand.

As expected Instance 2's solution did cause a number of constraints to be violated when it was applied to the scenario-specific observed demand. After an inspection it was determined that 17.03% of the violations were due to one or more scenario-specific demand nodes being uncovered; 23.63% were due to fewer ambulances being allocated than were available; 54.59% were due to more ambulances being allocated than were available; and 4.40% were due to holding site node capacity being exceeded. The actual percentage of calls that cannot be considered to be covered by the implemented solution is 0.64%. The high percentage of ambulance allocation constraint violations are attributed to the fact that the solution process does not have the luxury of unlimited running time. The violations that are due to more ambulances being allocated than were available and holding site node capacity being exceeded, makes Instance 2's solution infeasible for Scenario 1's observed demand and the overall expected coverage is not higher than when Instance 1's solution was applied to Scenario 1's observed demand rates.

6.5.3 Conclusion: Scenario 1

Comparing Instance 1 and Instance 2's results when their solutions were applied to the observed demand, shows that Instance 1's solution produces a higher expected coverage, lower relocation cost, and the solution is feasible. Instance 2's results do not prove that the WC ECC's fleet size or the holding site node capacity needs to be increased. Rather, it can be seen that with improved resource usage the WC ECC would be able to meet expected coverage targets with a smaller fleet when implementing the "golden hour" philosophy.

6.6 Scenario 2

As stated in Section 6.1 Scenario 2 considers calls according to their priority classification. The response time target for P1 calls is 15 minutes and the response time target for P2 calls is 30 minutes. In Section 6.6.1 and 6.6.2, respectively, the results for applying Instance 3's solution and Instance 4's solution to Scenario 2's predicted demand and observed are provided. In Section 6.6.3 the instance-specific results for Scenario 2's observed demand are compared and the best instance determined.

6.6.1 Instance 3

The shift results for Instance 3's solution, i.e. Scenario 2 with the relaxed ambulance allocation constraint, based on the Scenario 2's predicted demand rates are shown in Table G.1. The percentage expected coverage exceeds the service reliability value for all shifts. The lowest percentage of expected coverage is 96.25%, for 2016/01/01's night shift. Table 6.4 shows that Instance 3's solution does not violate any constraints for the scenario-specific predicted demand. The week's expected coverage is 96.62% with a relocation cost of 399.69 and an average ambulance usage per hour of 43.10 for the day shift and 30.07 for the night shift. These results are better than what was found with Instance 1's solution for Scenario 1's predicted demand. Intuitively, this outcome was to be expected, because the response time target for a part of the overall number of calls (P2 calls) is lower, i.e. longer, in Scenario 2 than in Scenario 1, where all calls have the same 15 minute response time target. Instance 3's solution for Scenario 2's predicted demand is feasible and viable.

Instance 3's solution was then applied to the Scenario 2's observed demand. Table G.7 contains the results for the 14 shifts, and the percentage expected coverage ranges from 93.94%, for 2016/01/01's day shift, to 96.15%, for 2016/01/07's day shift. The coverage is not as high as when Instance 3's solution was applied to the scenario-specific predicted demand, this is as expected.

Instance 3's solution did violate constraint (3.18) when it was applied to Scenario 2's observed demand. This indicates that one or more scenario-specific observed demand nodes are uncovered. The percentage of calls which are not covered by the implemented solution is 1.01%, where 2.69% is P1 calls and 0.06% is P2 calls. The expected coverage for the week with the implemented solution is 94.99%. This is an acceptable percentage with a low accompanied relocation cost. Also, this instance requires fewer ambulances than Instance 1. The results for Instance 3's solution applied to the scenario-specific observed demand proves that, as long as a response target of 30 minutes for P2 calls is acceptable to the WC ECC, this is a valid and viable, i.e. feasible, solution for the WC ECC's real-world instance. On average, the approach achieves the target response times more effectively (although the response time target for P2 calls is of course longer than in Scenario 1), whilst also requiring fewer resources (ambulances). The only drawback is that the percentage of calls left uncovered is higher for Instance 3's solution applied to Scenario 2's observed demand, than for Instance 1's solution applied to Scenario 1's observed demand. However, the probability of reaching the calls that are covered are higher for Instance 3's solution.

6.6.2 Instance 4

The shift results for Instance 4, i.e. Scenario 2 with the original allocation constraint, for Scenario 2's predicted demand are shown in Table G.6. None of the shifts have percentage expected coverage values below 93.30%, as seen for 2016/01/05's day shift. The week's expected coverage is 95.46% which is lower than when Instance 3 was implemented for the scenario-specific predicted demand. This was also seen between Instance 1 and Instance 2, where Instance 1's expected coverage for the planning horizon was higher than Instance 2's when implemented for the scenario-specific predicted demand. The average ambulance usage per hour is 65.94 for the day shift and 64.39 for the night shift, and the relocation cost is 1,479.45. Again, as with the relationship between

Instance 1 and Instance 2, the ambulance usage and relocation cost are higher than for Instance 3's solution.

Instance 4's solution for Scenario 2's predicted demand does violate a number of constraints for a number of hour sequences, as seen in Table 6.4. After an inspection, it was determined that 0.7% of the violations were due to one or more scenario-specific demand nodes being uncovered; 39.44% were due to fewer ambulances being allocated than were available; 55.63% were due to more ambulances being allocated than were available; and 4.23% were due to one or more holding site node capacities being exceeded. One P1 scenario-specific predicted call cannot be considered covered by Instance 4's solution, which is 0.04% of all the P1 predicted calls. The violations that are due to more ambulances being allocated than were available and holding site node capacity being exceeded makes Instance 4's solution infeasible for Scenario 2's predicted demand.

Instance 4's solution for Scenario 2's predicted demand was also applied to the scenario-specific observed demand rates, even though the solution was found to be infeasible for the predicted demand rates. Similar results to what was found when Instance 2's infeasible solution was applied to Scenario 1's observed demand, were expected. Table G.8 contains the results for the 14 shifts. The range of percentage expected coverage values is lower than what was found when Instance 3's solution was applied to Scenario 2's observed demand rates. The percentage expected coverage ranges from 91.51%, for 2016/01/05's day shift, to 95.68%, for 2016/01/06's night shift. The overall expected coverage for the week is 93.97%, which is also lower than what was found when Instance 3's solution was applied to the scenario-specific observed demand rates.

Table 6.4 shows that Instance 4's solution violates constraints when it is applied to Scenario 2's observed demand rates: 23.78% were due to one or more scenario-specific observed demand nodes being uncovered; 30.27% were due to fewer ambulances being allocated than were available; 42.70% were due to more ambulances being allocated than were available; and 3.24% were due to holding site node capacity being exceeded. The percentage of scenario-specific observed calls which were not covered by the Instance 4's solution is 0.91% calls, where 2.47% is P1 calls and 0.02% is P2 calls. The violations that are due to more ambulances being allocated than were available and holding site node capacity being exceeded, makes Instance 4's solution infeasible for Scenario 2's observed demand rates and the overall expected coverage is not higher than when Instance 3's solution was applied to Scenario 2's observed demand.

6.6.3 Conclusion: Scenario 2

Comparing Instance 3 and Instance 4's results when the solutions were applied to Scenario 2's observed demand, showed the same relationship as was found between Instance 1 and Instance 2. Instance 3's solution resulted in higher expected coverage, lower relocation cost, and the solution is feasible for Scenario 2's observed demand. As expected the results for Instance 4's solution applied to Scenario 2's observed demand did not prove a required increase in fleet size or holding site node capacity. It is interesting to note that the percentage of calls left uncovered are lower for Instance 4, but this might just be due to the fact that for some hours more ambulances were used than were available. It is believed that the ABC algorithm is not given enough time to run through the large solution space to find viable results with the original ambulance allocation constraint, but that an increase in run time will not provide an Instance 4 solution that would provide a higher expected coverage for the observed demand rates than the resulting Instance 3 solution. The results rather show that the WC ECC should be able to meet expected coverage targets with a smaller fleet when calls are handled according to their incident type, i.e. when the "golden hour" philosophy is not implemented.

6.7 Comparison of results

Section 6.5.1 - 6.5.2 and Section 6.6.1 - 6.6.2 provided the results when the instances were solved for the scenario-specific predicted demand rates and the results when the instances' solutions were applied to the scenario-specific observed demand rates. The expected coverage results for the same instance's solution was lower when applied to the observed demand rates than when applied to the predicted demand rates. This was not unexpected, since each instance's solution was created for the scenario-specific predicted demand. The relocation costs for the instances' solutions when applied to the scenario-specific predicted and observed demand are identical, since the relocations along with the holding site node selection and ambulance allocations are specific to the applied solution.

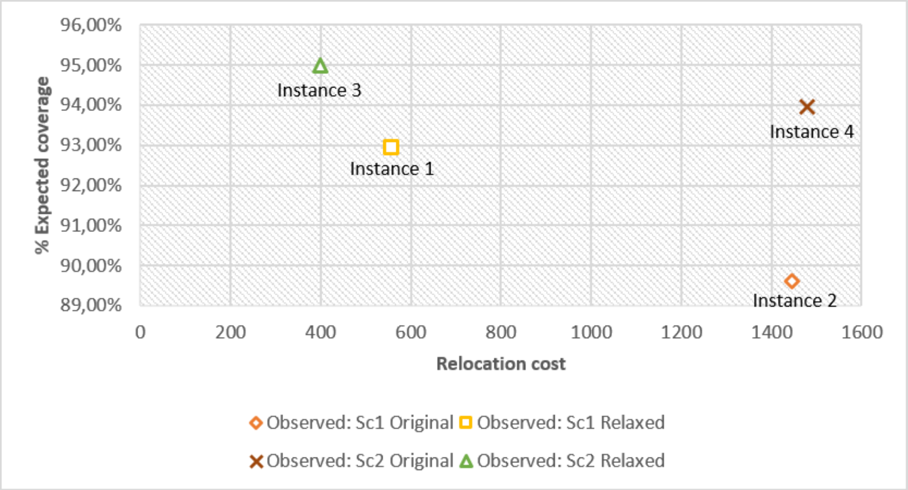


Figure 6.1: Comparison graph: Instance-specific results.

Figure 6.1 shows the percentage expected coverage versus the relocation cost of the results for the observed demand rates for the two scenarios and the two ambulance allocation implementations. The best, and the only feasible, instance for Scenario 1's observed demand was determined to be Instance 1, and the best, and only feasible, instance for Scenario 2's observed demand was determined to be Instance 3. It is also seen in Figure 6.1 that these two feasible solutions provided the two best results for the expected coverage and relocation cost for the planning week.

Instance 3's solution implemented for Scenario 2's observed demand provided the highest expected coverage (94.99%), and lowest relocation cost (399.69), for the planning week, shown in Table 6.2. However, the percentage calls left uncovered is 1.01%, where 2.69% are P1 and 0.06% are P2 calls.

Instance 1's solution implemented for Scenario 1's observed demand provided the second highest expected coverage (92.96%), and the second lowest relocation cost (557.27). The percentage calls left uncovered is 0.67%, with all calls classified as P1.

In terms of the number of calls left uncovered Instance 1 is the better option. However, the higher expected coverage found with Instance 3 means that the number of Scenario 2 observed calls covered have a higher probability of actually being reached within the specific response time standards. The final choice between Instance 1 and Instance 3 for the WC ECC's real-world instance when looking at the percentage of calls left uncovered and the probability of the covered calls actually being reached within the response time target is purely a medical one and cannot be determined in this project.

The winning instance for both scenarios is the one that implemented the relaxed ambulance allocation constraint. Therefore, it is concluded that for both Scenario 1, when the "golden hour" philosophy is implemented, and Scenario 2, when calls are handled according to their incident type, the WC ECC should be able to improve their expected coverage with the current, or even smaller, fleet size if holding site placement, ambulance allocation, and relocation decisions are made in anticipation of possible future demand.

It could be that if the MCN value was to be increased that the results for the original ambulance allocation constraint, i.e. Instance 2 and Instance 4, would be better, but the choice of the MCN value is limited by the realistic run time available for solving for the week planning horizon. In an ideal environment, the model would be allowed to run for an exceedingly longer time in order to explore a larger part of the solution space. This is not a feasible option for such large time specific problems, since the time required would be greater than that which is available for the decision to be made, i.e. when a recommended solution is finally found it would not be useful any more as these problems are time specific.

The actual percentage calls that were covered during the planning week, 2016/01/01 - 2016/01/07, was determined from the historical data provided by the WC ECC, for the real-world instance. The calculations were done in terms of the two scenarios to allow for comparison with each scenario's best instance. The actual percentage calls that were covered are shown in Table 6.5, for Scenatio 1, 33% percent of calls were covered and for Scenario 2, 38%. The highest expected coverage found for Scenario 1's observed demand with the DST was with Instance 1, i.e. the relaxed ambulance allocation constraint, and the value was 92.96%. The highest expected coverage found for Scenario 2's observed demand was also with the relaxed ambulance allocation constraint, i.e. Instance 3, and the value was overall the best with 94.99%. The difference between the actual percentage coverage and the DST's best instance's solution's expected percentage coverage is significant, exceeding 150% for both scenarios. However, it is important to note that it is invalid to compare these values like-for-like as a significant number of real-world factors, including the specific road conditions at the time of each call, the responsiveness of both the ECC operator handling the call and the ambulance team involved, and the communication connection between the ECC call operator and the ambulance team, influence the real-world response rate and could not be modelled in the DST. However, even when these factors are taken into account, the discrepancy between the historical and predicted performance presented in Table 6.5 is sufficient to convincingly demonstrate the potential of the DST to assist the WC ECC in further improving their response time and coverage.

Table 6.5: Actual historical performance contrasted with predicted DST-assisted performance for the week 2016/01/01 - 2016/01/07.

	Percentage calls covered[1]	
	Without DST-assistance[2]	**With DST-assistance**[3]
Scenario 1	32.56%	92.96%
Scenario 2	37.56%	94.99%

6.8 Recommendations for future use

In Section 6.7 it was concluded that Instance 1 is the best for Scenario 1 and Instance 3 is the best for Scenario 2. The final choice between these two instance for the WC ECC's real-world instance is purely a medical one and cannot be determined in this project. However, it was concluded that the WC ECC should be able to improve their expected coverage with the current, or even smaller, fleet size if holding site placement, ambulance allocation, and relocation are done in anticipation of possible future demand. Therefore, if the DST is to be tested at the WC ECC the relaxed ambulance allocation constraint should be implemented for both scenarios. The instance-specific results should be compared and medical input should be gathered to conclude which is the best scenario for the WC ECC's real-world instance.

If possible the DST's coding should be improved to the point that the MCN value of the ABC algorithm can be increased without a large increase in the run time. This would allow for a greater exploration of the search space, which is required for Instance

[1] It is invalid to compare these values on a like-for-like basis, refer to the accompanying text.

[2] Based on historical performance data obtained from the WC ECC for the week 2016/01/01 - 2016/01/07.

[3] Based on the predicted performance of the DST for Instance 1 and Instance 3 respectively, on the observed demand.

2 and Instance 4. The expectation is that it should be possible to find a solution for Instance 1, i.e. Scenario 1 with the original ambulance allocation constraint, and Instance 4, i.e. Scenario 2 with the original ambulance allocation constraint, which is feasible. However, the results are not expected to be better than those found for Instance 1 and Instance 3, respectively. The results would provide the WC ECC with a better idea of what is possible with their current fleet.

Finally, if the forecasting process has to be repeated for any new historical data or scenario then the control parameter selection process, set out in Section 5.5.3, should be followed.

6.9 Validation and verification

Validation and verification is not a step that is done only once at a certain point during the creation of a model or simulation; it is an activity that needs to be done continuously throughout the creation and testing processes, and even after the model or simulation is finished (Balci, 1997). Validation deals with building the right model and verification with building the model correctly (Balci, 1997; Robinson, 1997).

Model validation evaluates whether the model, behaves with satisfactory accuracy according to the model's objectives within its application environment (Balci, 1997; Robinson, 1997). This means testing the model's results for accuracy and comparing the results to what was expected. Model verification is the process of substantiating that the conceptual model has been transformed into a computer model with sufficient accuracy (Balci, 1997; Robinson, 1997). Sufficient accuracy refers to the fact that no model is 100% accurate. Also, this accuracy refers to the model's purpose; therefore the objectives of a model need to be known before it can be validated and verified (Robinson, 1997).

The DST's forecasting and solution processes were validated and verified during its creation and after it was finished. The historical call data also had to be validated, to ensure that the data is sufficiently accurate for its purpose as input for the forecasting method, SSA. This validation was done in Section 5.2.2, where it was determined that the data does not contain outliers. Whenever new data is added to the DST's historical data, it will have to be validated again.

The validation of the forecasting method started in Section 3.2, where the decision concerning the forecasting method was based on research. The SSA method was chosen for its ease of use, robustness, and the fact that it has been used to forecast ambulance demand before (Gillard & Knight, 2014; Vile *et al.*, 2012) and was deemed to outperform methods currently used. To improve the likelihood of implementing the SSA algorithm correctly, i.e. verification process, it was decided to use a tested SSA Matlab code created by Tkachev (2014). Also, the predicted demand was compared to the actual demand using a forecast-accuracy metric, RMSE. The SSA's user-selected values – window length and number of components to use to reconstruct the time series – per scenario were selected after a number of combinations were tested; the combination with the highest accuracy, for each scenario, in terms of time and location was finally selected. This was done in Section 5.5.2 for Scenario 1 and in Section 6.3 for Scenario 2. The validation of the accuracy of the SSA method for the project could only be done internally, because no access was provided to a second independent data set.

The solution process of the DST consists of the problem model, i.e. the extended Q-MALP, and the solution methods, i.e. the ABC algorithm and simplex method. The choice of problem model was based on research done on similar problems (Section 3.7) and on the requirements of the WC ECC's real-world instance (Chapter 4), i.e. the objectives.

The model formulation was validated also by comparing the extended QMALP model, as implemented by Andrade & Cunha (2015), to the original Q-MALP (Marianov & ReVelle, 1996) and a problem-specific implemented Q-MALP (Ghani, 2012). This was done in Section 4.2.2, therefore it was not assumed that Andrade & Cunha (2015) adapted the model without any mistakes and the model's formulation was validated. The functionality of the formulated model was validated during scenario analysis.

The validation of the solution method for the location-allocation phase of the objective function of the model started in Section 3.9 with research on meta-heuristics. The ABC algorithm was chosen based on its ease of use and the fact that it had been used to solve the model for a similar problem. The solution method's functionality was validated during scenario analysis. A Matlab code for the basic ABC algorithm created by Heris (2015), was used and modified for use on constrained optimisation.

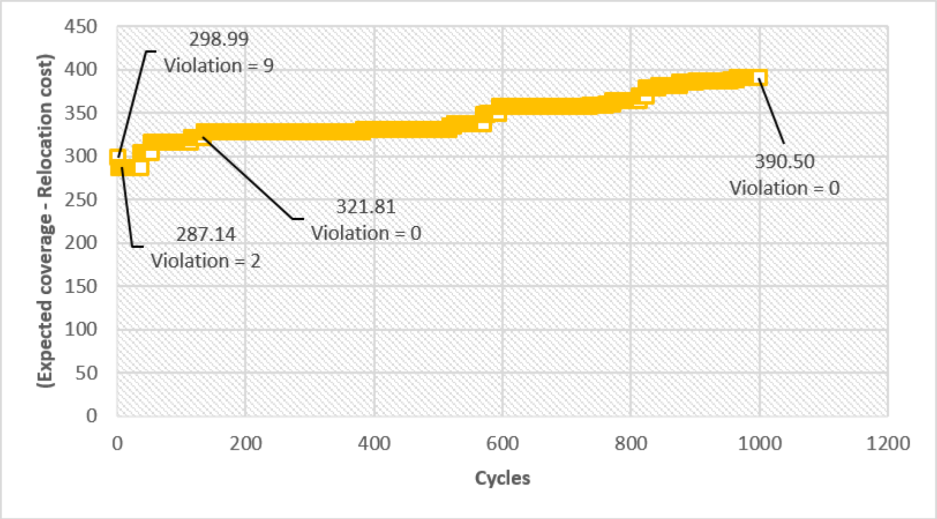


Figure 6.2: Convergence graph: Instance 3; 2016/01/01's night shift.

The changes were based on work done by Karaboga & Basturk (2007), Brajevic (2010), and Karaboga & Akay (2011).

The results in Sections 6.5.1 - 6.6.3 form part of scenario analysis. This was used to determine whether the modified ABC algorithm converged. If it does converge, it shows that the solution method behaves with satisfactory accuracy (validation) and that it was transformed into the coded Matlab correctly (verification). Figure 6.2 shows the convergence graph for the modified ABC algorithm, when Instance 3 was solved for 2016/01/01's night shift's Scenario 2 predicted demand rates. Though the modified ABC algorithm coded in the DST aimed to minimise the objective function value, the negative sign was omitted for the graph. Therefore, since the plot moves towards the maximum it can be stated that the coded ABC algorithm does converge.

The relocation phase of the objective function of the model required its own solution method. The relocation problem was defined and described as a minimum cost flow problem in Section 4.3.2. It was also stated that the method most often used to solve this type of problem is the simplex method. Matlab has a built-in dual-simplex method, which only requires the problem to be written in a specified form. The build of the dual-simplex method was therefore verified by Matlab. The results of Matlab's dual-simplex method were deemed accurate after the model and solution methods were run for small problems and the answers were checked by hand. This allowed for a small scale validation of the model and both solution methods.

The combination of the problem model and solution methods were tested in this chapter. The results for the four instances could only be internally validated and verified, as no access was provided to a second independent data set. The solution process was, therefore, tested by determining a solution for the planning horizon based on the predicted demand rates, for each of the four instances, and evaluating the results. The instances' solutions' results for the scenario-specific predicted demand were validated by applying the solutions to the scenario-specific observed demand rates and comparing the results. The scenario-specific observed demand rates of the planning horizon was not part of the input for forecasting the scenario-specific demand rates for the planning horizon. The results were validated by considering the accuracy of the solution, while remembering that the results could not be compared like-for-like. This was done for the four instances in Sections 6.5 to 6.6.3.

If this project were to be taken further, the next step for verification and validation would be to test the model as-is at the WC ECC. This process will allow for validation and verification of the DST with its usefulness in reality, where this project could only consider its usefulness in terms of static historical data. This will also provide information on which to base improvements to the model and its programming, and the creation of a user-interface. The process is expected to take a long time, it will most probably require weeks of observation, tests, and interview in order to gain the required information. Thereafter, the improvements, reprogramming, and programming of the user-interface will require more time.

6.10 Conclusion: Scenario analysis

In Chapter 6 the use of the DST for four instances were considered, recommendations for future use provided, and the validation and verification process followed during the creation and testing of the DST provided. In Chapter 7 the project summary is provided and the most significant research findings are highlighted. The value of the DST for the EMS field and its contributions to the WC ECC in particular, will also be explained. Finally, opportunities for further study will be discussed.

Chapter 7

Conclusion

In Chapter 6 the two scenarios and the two implementations of the ambulance allocation constraint, which form the four instances for which the DST was run, were explained. The resulting solutions were then applied to the scenario-specific observed demand rates and compared. The best instance for each scenario for the WC ECC's real-world instance was determined and recommendations were made for future use of the concept demonstrator DST. Finally, the validation and verification process followed during and after the creation and testing of the DST were explained. In this chapter the project summary and the research findings are provided. The contributions of the research and opportunities for further work are also described.

7.1 Project summary

The project was conducted in association with the WC ECC for the purpose of developing a concept demonstrator DST to assist the dispatchers' decision-making process by providing near-optimal holding site node selection per shift, ambulance allocation, and relocation per hour of that shift, for each day of a week, for the City of Cape Town and the Cape Winelands municipalities based on predicted ambulance demand rates. The DST is not created to replace the dispatchers, but provide them with information on which to base their decisions. Although the concept demonstrator DST was created specifically for this real-world instance, it can be applied to other real-world instances with or without minimal modifications. The design type of the project was determined to be statistical modelling and computer simulation, which focusses on the development and validation of accurate models for real-world circumstances.

Research was conducted regarding the necessity of ambulance efficiency, the role OR plays in EMS management, DSTs in EMS, ambulance location problem models, and solution methods. Decisions were made based on this research. The SSA method was chosen to forecast the ambulance demand, the extended Q-MALP model was chosen and adapted, and the ABC algorithm was coded to determine the near-optimal solution for the constrained problem. The real-world instance on which the DST was tested is six months' historical call data, 1 August 2015 at 7 A.M. until 29 February 2016 at 06.59 A.M., from the City of Cape Town and the Cape Winelands municipalities, provided by the WC ECC. The DST was programmed in Matlab, where every '*.m'-file contains the code for a different process.

The completed concept demonstrator DST was run for four instances, i.e. one of two scenarios with one of two ambulance allocation constraint implementations, and the input data was the predicted ambulance demand rates. The predicted ambulance demand for the planning week was predicted with the SSA method, but the historical demand for the planning week was not part of input for the SSA method. The resulting solutions for the predicted demand were implemented for the scenario-specific observed demand. The expected coverage and relocation cost results for the four instances' solutions implemented on the scenario-specific observed demand were compared and the best instance for each scenario was identified.

7.2 Research findings

It was determined that the accuracy of the predicted demand with SSA does depend on the size of the demand nodes and the value of L, the expected periodicity. The conclusion is that the demand nodes should be chosen to be as small as possible, so long as the computer can accommodate the increased memory usage required. The smaller the demand nodes the larger the number of demand nodes for which predictions have to be made and the more computer memory is required. The number of demand nodes found to work best for Scenario 1 and Scenario 2 was 10,000 demand nodes. It was also determined that the best expected periodicity assumption for the two scenarios was monthly, but that the value for N has to be determined through trial-and-error.

The best and the only feasible instance solutions for the observed demand were found with Instance 1 for Scenario 1 and Instance 3 for Scenario 2. Instance 1 and Instance 3 have the relaxed ambulance allocation constraint implemented. This led to the conclusion that the WC ECC should be able to increase their expected coverage with the current fleet, or even a smaller fleet, provided that the fleet, holding sites chosen, and relocations are managed based on predicted ambulance demand rates. The DST cannot replace the dispatchers, but can provide knowledge on which to base their decisions.

For the entire planning week, Instance 3 provided the highest percentage expected coverage, 94.99%, and the lowest relocation cost, 399.69, but left 1.01% of the calls uncovered, where 2.69% are P1 and 0.06% are P2 calls. Instance 1 provided the second highest percentage expected coverage, 92.96%, and the second lowest relocation cost, 557.27, and its percentage calls left uncovered is lower than that of Instance 3, at 0.67%, with all calls classified as P1. The higher expected coverage found with Instance 3 means that the number of Scenario 2 observed calls covered have a higher probability of actually being reached within the specific response time standards. It was determined that the choice to handle holding site placement, ambulance allocation, and relocation according to Scenario 1 or Scenario 2 is primarily a medical one and cannot be determined in this project.

The data presented in Table 6.5 indicates that the actual percentage of calls covered for the two scenarios, as calculated based on the historical call data, is markedly lower than the expected percentage coverage determined to be likely for any of the instances with the DST (refer to Table 6.5). Though it is invalid to compare these performance statistics on a like-for-like basis, the difference is sufficiently large to conclude that the DST holds significant potential to assist the WC ECC in further improving their emergency responsiveness. Further development of this concept demonstrator into a DST to be implemented at the WC ECC is therefore recommended.

7.3 Research contributions

It is possible to implement an emergency-specific response time standard in first world countries with accurate historical data, forecasting methods, and call-takers trained to the point of medical diagnosticians. This is not yet a possibility in most of South

Africa's provinces. However, since the implementation of CareMonX in 2014, the WC ECC has greatly improved the quality of their data. The use of the CareMonX system along with a fully equipped version of the concept demonstrator DST emanating from this project would bring the WC ECC closer to meeting the highest international standards. In summary, the use of the DST can help dispatchers to make decisions which should improve the expected ambulance coverage and the efficiency of the ambulance service.

This project demonstrates that providing dispatchers at ECCs in South Africa with decision-support on the allocation of ambulances to holding sites based on predicted demand, has significant potential to improve emergency responsiveness. The concept demonstrator DST conceptualised and developed in this research provides a basis for the development of a DST that can be implemented at ECCs in South Africa. However, it must be made known that the DST cannot and should not replace the dispatchers.

The step-by-step process for selecting the best combination of the number of demand nodes, L, and N for forecasting spatio-temporal data with SSA, is another contribution. It provides a starting point from which to expand the research in implementing the SSA method for forecasting ambulance demand. The method's popularity is due to its ease of use, but the control parameter selection is still a difficult and tedious process.

7.4 Opportunities for further work

The concept demonstrator DST was created and used to solve for the WC ECC's real-world instance when all calls are considered to be from urban areas. As it stands, the Matlab code that comprises the DST is not user-friendly, and it requires approximately 20 hours to solve for a shift day. An interesting opportunity for further work would be to take the concept demonstrator DST, implement it at the WC ECC, and evaluate whether there is a sufficiently large improvement in performance when its outcome is used to assist dispatcher decision-making to warrant the investment that would be required to develop this concept demonstrator into a DST that can be implemented at ECCs.

It was not possible to test the DST for calls from rural areas, due to a shortage of rural calls in the historical call data provided by the WC ECC. Call data from a district with a higher percentage of rural calls was requested in order to test the model in a rural setting. The data did contain more rural calls in a larger area than the original historical data, but it was still deemed too few calls to test the accuracy of the DST in a rural setting. This project can be expanded on by determining whether the DST's accuracy and usefulness holds true when rural call data forms part of the real-world instance's data.

Another opportunity, which should be explored along side the previous one, is to code the DST to run in parallel, which would decrease the run time. Matlab does have a package for parallel computing, but it was not part of the student license bought and used for this project. The DST can also be coded in another programming language which might be able to run the processes faster. The DST also does not have a user interface, but requires the user to actually make changes in the code, which could cause problems. If any further work is done a user interface should be added.

7.5 Conclusion

In Chapter 7 a summary of the project was described, and the research findings were provided. Thereafter, the contributions of this research were explained, and the possible opportunities for furthering this research were described.

References

Aartun, E.N. & Leknes, H. (2014). *Strategic ambulance location: optimization with multiple performance measures*. Master's, Norwegian University of Science and Technology.

Alexandrov, T. (2009). A Method of Trend Extraction Using Singular Spectrum Analysis. 1–22.

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

Alter, S. (1977). A Taxonomy of Decision Support Systems. *Sloan Management Review*, **19**, 39–56.

Andersson, T. & Värbrand, P. (2006). Decision support tools for ambulance dispatch and relocation. *Journal of the Operational Research Society*, **58**, 195–201.

Andrade, L.A.C.G. & Cunha, C.B. (2015). An ABC heuristic for optimizing moveable ambulance station location and vehicle repositioning for the city of São Paulo. *International Transactions in Operational Research*, **22**, 473–501.

Arnott, D. & Pervan, G. (2008). Eight key issues for the decision support systems discipline. *Decision Support Systems*, **44**, 657–672.

Başar, A., Çatay, B. & Ünlüyurt, T. (2012). A taxonomy for emergency service station location problem. *Optimization Letters*, **6**, 1147–1160.

Balci, O. (1997). Verification, Validaiton and Accreditation of simulation models. *Proceedings of the 1997 Winter Simulation Conference*, 135–141.

Bansal, J.C., Sharma, H. & Jadon, S.S. (2013). Artificial bee colony algorithm: a survey. *International Journal of Advanced Intelligence Paradigms*, **5**, 123.

Basu, S., Sharma, M. & Ghosh, P.S. (2015). Metaheuristic applications on discrete facility location problems: a survey. *Opsearch*, **52**, 530–561.

Bélanger, V., Ruiz, A. & Soriano, P. (2015). Recent advances in emergency medical services management. *International Research Centre on Enterprise Networks, Logistics and Transportation*, **28**, 38.

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

Beraldi, P., Bruni, M. & Conforti, D. (2004). Designing robust emergency medical service via stochastic programming. *European Journal of Operational Research*, **158**, 183–193.

Bertsimas, D. & Tsitsiklis, J. (1993). Simulated annealing.pdf. *Statistical Science*, **8**, 11–15.

Bianchi, G. & Church, R.L. (1988). A hybrid fleet model for emergency medical service system design. *Social Science & Medicine*, **26**, 163–171.

Blackwell, T.H. & Kaufman, J.S. (2002). Response time effectiveness: Comparison of response time and survival in an urban emergency medical services system. *Academic Emergency Medicine*, **9**, 288–295.

Blum, C. & Roli, A. (2003). Metaheuristics in combinatorial optimization. *ACM Computing Surveys*, **35**, 268–308.

Blum, C., Aguilera, M.J.B., Roli, A. & Sampels, M. (2008a). *Hybrid Metaheuristics: An Emerging Approach to Optimization*. Springer Publishing Company, Incorporated, 1st edn.

Blum, C., Roli, A. & Sampels, M. (2008b). *Hybrid Metaheuristics: An Emerging Approach to Optimization*. Studies in Computational Intelligence, Springer.

Brajevic, I. (2010). Improved Artificial Bee Colony Algorithm for Constrained Problems. In *Proceedings of the 11th WSEAS International Conference on Nural Networks and 11th WSEAS International Conference on Evolutionary Computing and 11th WSEAS International Conference on Fuzzy Systems*, 185–190, World Scientific and Engineering Academy and Society (WSEAS), Wisconsin.

Brajevic, I., Tuba, M. & Subotic, M. (2011). Performance of the improved artificial bee colony algorithm on standard engineering constrained problems. *International Journal of Mathematics and Computers in Simulation*, **5**, 135–143.

Briceño, H., Rocco, C.M. & Zio, E. (2013). Singular Spectrum Analysis for Forecasting of Electric Load Demand. In *4th IEEE Conference on Prognostics and System Health Management (PHM)*, vol. 33, 919–924.

Broomhead, D., Jones, R., King, G. & Pike, E. (1987). *Singular System Analysis with Application to Dynamical Systems*. CRC Press, Bristol.

Broomhead, D.S. & King, G.P. (1986). Extracting qualitative dynamics from experimental data. *Physica D: Nonlinear Phenomena*, **20**, 217–236.

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

Busetti, F. (1983). Simulated annealing overview. 1–10.

Cantwell, K., Morgans, A., Smith, K., Livingston, M. & Dietze, P. (2015). Temporal trends in cardiovascular demand in EMS: Weekday versus weekend differences. *Chronobiology International*, **32**, 731–738.

Cardoen, B., Demeulemeester, E. & Beliën, J. (2010). Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, **201**, 921–932.

CerebPalsy.org (2015). Hypoxic-Ischemic Encephalopathy, or HIE, also known as Intrapartum Asphyxia.

Channouf, N., L'Ecuyer, P., Ingolfsson, A. & Avramidis, A.N. (2007). The application of forecasting techniques to modeling emergency medical system calls in Calgary, Alberta. *Health Care Management Science*, **10**, 25–45.

Chanta, S., Mayorga, M.E. & McLay, L.a. (2014). The minimum p-envy location problem with requirement on minimum survival rate. *Computers & Industrial Engineering*, **74**, 228–239.

Church, R. & ReVelle, C. (1974). The maximal covering location problem. *Papers of the Regional Science Association*, **32**, 101–118.

Cunningham, W.H. (1976). A network simplex method. *Mathematical Programming*, **11**, 105–116.

Dahl, S. & Derigs, U. (2011). Cooperative planning in express carrier networks - An empirical study on the effectiveness of a real-time Decision Support System. *Decision Support Systems*, **51**, 620–626.

Dan Dan, E. & Ijeoma, O.A. (2013). Statistical Analysis/ Methods of Detecting Outliers in a Univariate Data in a Regression Analysis Model. *International Journal of Education and Research*, **1**.

Daskin, M.S. (1983). A Maximum Expected Covering Location Model: Formulation, Properties and Heuristic Solution. *Transportation Science*, **17**, 48–70.

Daskin, M.S. & Stern, E.H. (1981). A Hierarchical Objective Set Covering Model for Emergency Medical Service Vehicle Deployment. *Transportation Science*, **15**, 137–152.

Daskin, M.S., Hogan, K. & ReVelle, C. (1988). Integration of Multiple, Excess, Backup, and Expected Covering Models. *Environment and Planning B: Planning and Design*, **15**, 15–35.

de Klerk, J. (1994). Automated outlier detection in Singular Spectrum Analysis. 2–7.

De Klerk, J. (2015). Time series outlier detection using the trajectory matrix in singular spectrum analysis with outlier maps and ROBPCA. *South African Statistics Journal*, **49**, 61–76.

De Vries, S., Wallis, L.A. & Maritz, D. (2011). A retrospective evaluation of the impact of a dedicated obstetric and neonatal transport service on transport times within an urban setting. *International Journal of Emergency Medicine*, **4**, 28.

DEB, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, **186**, 311–338.

DENSHAM, P. (1991). Spatial Decision Support Systems. In D. Maguire, M. Goodchild, D. Rhind & P. Longley, eds., *Geographical information systems: Principles and applications*, vol. 1, chap. 26, 403–412, Wiley (1999), 2nd edn.

DOH: PROVINCE OF KWAZULU-NATAL (2001). Health Districts.

EATON, D.J., U, H.M.L.S., LANTIGUA, R.R. & MORGAN, J. (1986). Determining Ambulance Deployment in Santo Domingo, Dominican Republic. *Journal of the Operational Research Society*, **37**, 113–126.

EDMONDS, J. & KARP, R.M. (1972). Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. *Journal of the ACM*, **19**, 248–264.

EI-MOWAFI, D.M. (2008). Bleeding in Late Pregnancy (Antepartum Bleeding).

ERKUT, E., INGOLFSSON, A. & ERDOAN, G. (2008). Ambulance location for maximum survival. *Naval Research Logistics*, **55**, 42–58.

FARAHANI, R.Z., ASGARI, N., HEIDARI, N., HOSSEININIA, M. & GOH, M. (2012). Covering problems in facility location: A review. *Computers & Industrial Engineering*, **62**, 368–407.

FITZSIMMONS, J.A. (1973). A Methodology for Emergency Ambulance Deployment. *Management Science*, **19**, 627–636.

GENDREAU, M., LAPORTE, G. & SEMET, F. (1997). Solving an ambulance location model by tabu search. *Location Science*, **5**, 75–88.

GENDREAU, M., LAPORTE, G. & SEMET, F. (2001). A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel Computing*, **27**, 1641–1653.

GENDREAU, M., LAPORTE, G. & SEMET, F. (2006). The maximal expected coverage relocation problem for emergency vehicles. *Journal of the Operational Research Society*, **57**, 22–28.

Ghani, N.A. (2012). Multi-server Queuing Maximum Availability Location Problem with Stochastic Travel Times. In *Proceedings of the World Congress on Engineering*, vol. I, 1–7.

Gillard, J. & Knight, V. (2014). Using Singular Spectrum Analysis to obtain staffing level requirements in emergency units. *Journal of the Operational Research Society*, **65**, 735–746.

Glover, F. & Greenberg, H.J. (1989). New approaches for heuristic search: linkage with artificial intelligence. *European Journal of Operational Research*, **39**, 119–130.

Goldberg, A.V. (1997). An Efficient Implementation of a Scaling Minimum-Cost Flow Algorithm. *Journal of algorithms*, **22**, 1–29.

Goldberg, J., Dietrich, R., Ming Chen, J., Mitwasi, M., Valenzuela, T. & Criss, E. (1990). Validating and applying a model for locating emergency medical vehicles in Tucson, AZ. *European Journal of Operational Research*, **49**, 308–324.

Golyandina, N. & Korobeynikov, A. (2014). Basic Singular Spectrum Analysis and forecasting with R. *Computational Statistics & Data Analysis*, **71**, 934–954.

Golyandina, N., Nekrutkin, V. & Zhigljavsky, A. (2001a). *Analysis of time series structure: SSA and related techniques*, vol. 90 of *Monographs on statistics and applied probability*. Chapman and Hall, Boca Raton, Florida.

Golyandina, N., Nekrutkin, V. & Zhigljavsky, A. (2001b). Analysis of Time Series Structure: SSA and Related Techniques.

Griffiths, J.D. (2005). A queueing model of activities in an intensive care unit. *IMA Journal of Management Mathematics*, **17**, 277–288.

Groulx, R. (2007). How To Deal With Outliers Using Quartile Analysis and Service Level.

Grubbs, F.E. (1969). Procedures for Detecting Outlying Observations in Samples. *Technometrics*, **11**, 1–21.

Hassani, H. (2010). A brief introduction to singular spectrum analysis. 1–11.

HENDERSON, D., JACOBSON, S.H. & JOHNSON, A.W. (2003). The theory and practice of simulated annealing. In *Handbook of metaheuristics*, chap. 10, 287–319.

HERIS, S.M.K. (2015). Artificial Bee Colony in MATLAB.

HILLIER, F.S. & LIEBERMAN, G.J. (2010). Solving Linear Programming Problems: The Simplex Method. In *Introduction to Operations Research*, chap. 4, 89–160, McGraw-Hill, New York, ninth edn.

HOFFMAN, E. (1976). Mortality and morbidity following road accidents. *Annals of the Royal College of Surgeons of England*, **58**, 233–240.

HOGAN, K. & REVELLE, C. (1986). Concepts and Applications of Backup Coverage. *Management Science*, **32**, 1434–1444.

JAGTENBERG, C., BHULAI, S. & VAN DER MEI, R. (2015). An efficient heuristic for real-time ambulance redeployment. *Operations Research for Health Care*, **4**, 27–35.

JARVIS, J. (1985). Approximating the Equilibrium Behavior of Multi-Server Loss Systems. *Management Science*, **31**, 235–239.

KARABOGA, D. (2005). An idea based on honey bee swarm for numerical optimization. Tech. rep., Computer Engineering Department, Engineering Faculty, Erciyes University, Turkey.

KARABOGA, D. & AKAY, B. (2011). A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems. *Applied Soft Computing*, **11**, 3021–3031.

KARABOGA, D. & BASTURK, B. (2007). Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. In *Foundations of Fuzzy Logic and Soft Computing*, 789–798, Springer Berlin Heidelberg, Berlin, Heidelberg.

KARABOGA, D., GORKEMLI, B., OZTURK, C. & KARABOGA, N. (2014). A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review*, **42**, 21–57.

KERGOSIEN, Y., LENTÉ, C., PITON, D. & BILLAUT, J.C. (2011). A tabu search heuristic for the dynamic transportation of patients between care units. *European Journal of Operational Research*, **214**, 442–452.

Kirkpatrick, S., Gelatt, C.D. & Vecchi, M.P. (1983). Optimization by simulated annealing. *Science (New York, N.Y.)*, **220**, 671–680.

Kitahara, T. & Matsui, T. (2012). On the Number of Solutions Generated by Dantzig ' s Simplex Method for LP with Bounded Variables*. *Pacific journal of optimization*, **8**, 447 –455.

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

Koziel, S. & Michalewicz, Z. (1999). Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evolutionary Computation*, **7**, 19–44.

Lee, S. (2012). The role of centrality in ambulance dispatching. *Decision Support Systems*, **54**, 282–291.

Lei, T.L., Church, R.L. & Lei, Z. (2015). A unified approach for location-allocation analysis: integrating GIS, distributed computing and spatial optimization. *International Journal of Geographical Information Science*, 1–20.

Lerner, E.B. & Moscati, R.M. (2001). The Golden Hour: Scientific Fact or Medical "Urban Legend"? *Academic Emergency Medicine*, **8**, 758–760.

Li, X. & Yang, G. (2016). Artificial bee colony algorithm with memory. *Applied Soft Computing*, **41**, 362–372.

Li, X., Zhao, Z., Zhu, X. & Wyatt, T. (2011). Covering models and optimization techniques for emergency response facility location and planning: a review. *Mathematical Methods of Operations Research*, **74**, 281–310.

Lim, C.S., Mamat, R. & Braunl, T. (2011). Impact of Ambulance Dispatch Policies on Performance of Emergency Medical Services. *IEEE Transactions on Intelligent Transportation Systems*, **12**, 624–632.

Liu, Y., Roshandeh, A.M., Li, Z., Kepaptsoglou, K., Patel, H. & Lu, X. (2014). Heuristic Approach for Optimizing Emergency Medical Services in Road Safety within Large Urban Networks. *Journal of Transportation Engineering*, **140**, 04014043–9.

Luo, Q., Su, Q., Le, J. & Lu, L. (2013). A new model for planning emergency facilities in Shanghai. In *2013 10th International Conference on Service Systems and Service Management*, 224–227, IEEE.

Marianov, V. & Revelle, C. (1994). The Queuing Probabilistic Location Set Covering Problem and Some Extensions. *Socio-Economic Planning Sciences*, **28**, 167–178.

Marianov, V. & ReVelle, C. (1996). The queueing maximal availability location problem: a model for the siting ofemergency vehicles. *European Journal of Operations Research*, **93**, 110–120.

Mason, A.J. (2013). Simulation and Real-Time Optimised Relocation for Improving Ambulance Operations. In *Handbook of Healthcare Operations Management: Methods and Applications*, chap. 11, 289–317, Springer New York.

Matteson, D.S., McLean, M.W., Woodard, D.B. & Henderson, S.G. (2011). Forecasting Emergency Medical Service Call Arrival Rates. *The Annals of Applied Statistics*, **5**, 1379–1406.

Maxwell, M.S., Restrepo, M., Henderson, S.G. & Topaloglu, H. (2010). Approximate Dynamic Programming for Ambulance Redeployment. *INFORMS Journal on Computing*, **22**, 266–281.

Melanie, M. (1999). *An Introduction to Genetic Algorithms*. The MIT Press.

Mezura-Montes, E. & Velez-Koeppel, R.E. (2010). Elitist Artificial Bee Colony for constrained real-parameter optimization. In *2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010*, 1–8.

Mezura-Montes, E., Damín-Araoz, M. & Cetina-Domínguez, O. (2010). Smart flight and dynamic tolerances in the artificial bee colony for constrained optimization. In *2010 IEEE Congress on Evolutionary Computation, CEC*, 1–8.

Moeini, M., Jemai, Z. & Sahin, E. (2014). Location and relocation problems in the context of the emergency medical service systems: a case study. *Central European Journal of Operations Research*, 1–18.

Mohammadi, M., Dashti khotbesara, Z. & Mirzazadeh, A. (2014). MCLP and SQM models for the emergency vehicle districting and location problem. *Decision Science Letters*, **3**, 479–490.

Mouton, J. (2013). *How to succeed in your Master's & Doctoral Studies*. Van Schaik Publishers.

Muritiba, A.E.F. (2010). *Algorithms and Models For Combinatorial Optimization Problems*. Ph.D. thesis, Studiorum University of Bologna.

Nair, R. & Miller-Hooks, E. (2009). Evaluation of Relocation Strategies for Emergency Medical Service Vehicles. *Transportation Research Record: Journal of the Transportation Research Board*, **2137**, 63–73.

Naoum-Sawaya, J. & Elhedhli, S. (2013). A stochastic optimization model for real-time ambulance redeployment. *Computers & Operations Research*, **40**, 1972–1978.

Nguyen, N.H.T. (2015). *Quantitative Analysis of Ambulance Location-allocation and Ambulance State Prediction*. Ph.D. thesis, Linkoping University.

Nordqvist, C. (2012). What is preeclampsia? What causes preeclampsia?

Oxford University Press (2016). English Oxford Living Dictionaries.

Pattinson, R. (2013). *Saving Babies 2010-2011 : Eighth report on perinatal care in South Africa*. Tshepesa Press, Pretoria.

Pervan, G. & Arnott, D. (2005). A critical analysis of Decision Support Systems research. *Journal of Information Technology*, **20**, 67–87.

Poulton, M. & Roussos, G. (2013). Towards Smarter Metropolitan Emergency Response. In *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2576–2580, IEEE.

Rajagopalan, H.K. & Saydam, C. (2009). A minimum expected response model: Formulation, heuristic solution, and application. *Socio-Economic Planning Sciences*, **43**, 253–262.

RAJAGOPALAN, H.K., SAYDAM, C. & XIAO, J. (2008). A multiperiod set covering location model for dynamic redeployment of ambulances. *Computers and Operations Research*, **35**, 814–826.

REPEDE, J.F. & BERNARDO, J.J. (1994). Developing and validating a decision support system for locating emergency medical vehicles in Louisville, Kentucky. *European Journal of Operational Research*, **75**, 567–581.

REVELLE, C. & HOGAN, K. (1988). A Reliability-Constrained Siting Model with Local Estimates of Busy Fractions. *Environment and Planning B: Planning and Design*, **15**, 143–152.

REVELLE, C. & MARIANOV, V. (1991). A probabilistic FLEET model with individual vehicle reliability requirements. *European Journal of Operational Research*, **53**, 93–105.

ROBINSON, S. (1997). Simulation model verification and validation. In *Proceedings of the 29th conference on Winter simulation - WSC '97*, 53–59, ACM Press, New York, New York, USA.

RÖNQVIST, M. (2012). OR challenges and experiences from solving industrial applications. *International Transactions in Operational Research*, **19**, 227–251.

SCHILLING, D., ELZINGA, D.J., COHON, J., CHURCH, R. & SCHILLING, D. (1979). The Team / Fleet Models for Simultaneous Facility and Equipment Siting. *Transportation Science*, **13**, 163–175.

SCHMID, V. (2012). Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming. *European Journal of Operational Research*, **219**, 611–621.

SCHMID, V. & DOERNER, K.F. (2010). Ambulance location and relocation problems with time-dependent travel times. *European Journal of Operational Research*, **207**, 1293–1303.

SCHOON, M.G. (2013). Impact of inter-facility transport on maternal mortality in the Free State province. *South African Medical Journal*, **103**, 534–537.

SEO, S. & GARY M. MARSH, P.D. (2006). *A review and comparison of methods for detecting outliersin univariate data sets*. Master, University of Pittsburgh.

SETZLER, H., SAYDAM, C. & PARK, S. (2009). EMS call volume predictions: A comparative study. *Computers & Operations Research*, **36**, 1843–1851.

SIMPSON, N.C. & HANCOCK, P.G. (2009). Fifty Years of Operational Research and Emergency Response. *Journal of the Operational Research Society*, **60**, S126–S139.

SMITH, H.K., HARPER, P.R., POTTS, C.N. & THYLE, A. (2009). Planning sustainable community health schemes in rural areas of developing countries. *European Journal of Operational Research*, **193**, 768–777.

SOLAK, M.K. (2009). Detection of multiple outliers in univariate data sets.

SOUTH AFRICAN GOVERNMENT (2013). NSDA: A long and healthy life for all South Africans. Tech. rep., Department of Health.

SOUTH AFRICAN GOVERNMENT (2015). National Development Plan (2030) Executive Summary.

STANAREVIC, N., TUBA, M. & BACANIN, N. (2010). Enhanced Artificial Bee Colony Algorithm Performance. In *Proceedings of the 14thWSEAS international conference on computers: part of the 14thWSEAS CSCC multiconference-volume II, World Scientific and Engineering Academy and Society (WSEAS)*, 440–445.

STATISTICS SOUTH AFRICA (2013). *Millennium Development Goals: Country Report 2013*. Statistics South Africa, 2013, Pretoria.

STATISTICS SOUTH AFRICA (2015). *Millennium Development Goals: Country Report 2015*. Statistics South Africa, 2015, Pretoria.

STÖPPLER, M.C. (2014). Eclampsia.

STORBECK, J.E. (1982). Slack, natural slack, and location covering. *Socio-Economic Planning Sciences*, **16**, 99–105.

SU, Q., LUO, Q. & HUANG, S.H. (2015). Cost-effective analyses for emergency medical services deployment: A case study in Shanghai. *International Journal of Production Economics*, **163**, 112–123.

Technopedia.inc (2017). Simplex Method.

Thaddeus, S. & Maine, D. (1994). Too far to walk: Maternal mortality in context. *Social Science and Medicine*, **38**, 1091–1110.

Thakore, S., McGugan, E.A. & Morrison, W. (2002). Emergency ambulance dispatch: is there a case for triage? *Journal of the Royal Society of Medicine*, **95**, 126–129.

The National Institute of Neurological Disorders and Stroke rt-PA Stroke Study Group (1995). Tissue plasminogen activator for acute ischemic stroke. **333**, 1581–1587.

Tkachev, A. (2014). SSA for MATLAB master.

Toregas, C., Swain, R., ReVelle, C. & Bergman, L. (1971). The location of emergency service facilities. *Operations Research*, **19**, 1363–1373.

Trunkey, D.D. (1983). Trauma. Accidental and intentional injuries account for more years of life lost in the U.S. than cancer and heart disease. Among the prescribed remedies are improved preventive efforts, speedier surgery and further research. *Scientific American*, **249**, 28–35.

United Nations (2000). Millenium Summit (6-8 September 2000).

van den Berg, P., Kommer, G. & Zuzáková, B. (2015). Linear formulation for the Maximum Expected Coverage Location Model with fractional coverage. *Operations Research for Health Care*.

Van den Berg, P.L. & Aardal, K. (2015). Time-dependent MEXCLP with start-up and relocation cost. *European Journal of Operational Research*, **242**, 383–389.

Vile, J.L., Gillard, J.W., Harper, P.R. & Knight, V.A. (2012). Predicting ambulance demand using singular spectrum analysis. *The Journal of the Operational Research Society*, **63**, 1556–1565.

World Health Organization (2013). Health statistics and health information systems: Maternal mortality ratio (per 100 000 live births).

World Health Organization (2014). Media centre Preterm birth.

Yancey, A.H. & Mould-Millman, N.K. (2015). The implementation status of obstetric EMS systems in selected South African health districts with recommendations for future development. Tech. rep., South Africans and Americans in Partnership to fight HIV/AIDS.

Zhang, Z.H. & Jiang, H. (2014). A robust counterpart approach to the bi-objective emergency medical service design problem. *Applied Mathematical Modelling*, **38**, 1033–1040.

Zhou, Z. & Matteson, D.S. (2015). Predicting Ambulance Demand: a Spatio-Temporal Kernel Approach. 2297–2303.

Appendix A

The inter-facility transport studies

As mentioned in Section 1.1, the real-world problem was brought in focus by two studies concerning IFT in South Africa, with a specific focus on maternal emergencies.

The initial focus of this research project was to create a concept demonstrator Decision Support Tool (DST) for maternal ambulance dispatching. However, the focus of the project changed when the WC ECC decided to be associated with it. Their needs moved the project toward improving holding site placement and ambulance allocation.

The two IFT studies still provide necessary background information for this project. The focus of Study 1 is proving that dedicated Maternal IFT EMS units are needed to decrease MMR. Study 2 examines implementing the procedural changes of Study 1 for all types of maternal transport in other provinces. It also places a strong emphasis on the placement of dedicated EMS units in order to determine the best placement for shorter response times.

A.1 Study 1: Impact of inter-facility transport on maternal mortality in the Free State province (Schoon, 2013)

The Schoon (2013) study was undertaken to explore the potential impact of dedicated obstetric IFT on maternal mortality. It took place after the identification of IFT as a problem in the maternity services in the Free State. This province ranks among the highest in terms of its MMR in South Africa. Consequently, the Free State DoH decided to procure an additional 48 EMS units, of which 18 are dedicated to maternal care (Schoon, 2013). The Schoon (2013) study also focusses on transport as a contributing factor because of the De Vries *et al.* (2011) article, which documents that access to emergency obstetrics care and decreased transport times to emergency obstetric facilities can reduce the MMR.

The De Vries *et al.* (2011) article deals with a retrospective study undertaken between 2006 and 2008. This study revealed that EMS units and their crews that are specifically dedicated to maternal and neonatal responses have a significant and beneficial influence on dispatch, response, and mission times. This study was conducted by METRO EMS for the Cape Town area (De Vries *et al.*, 2011).

It was the findings in the De Vries *et al.* (2011) article, along with the identification of IFT as a problem in their maternity service, that led to the changes in the Free State. Along with the addition of 48 EMS units, these changes included a realignment of the Free State EMS sector, with primary calls being serviced separately from Maternal IFT (MIFT). The implementation of dedicated EMS units for maternal and neonatal responses allowed for a 43% improvement of the Free State's MMR in 2012, which decreased from 259 to 147 (Schoon, 2013).

This intervention provided more vehicles with new staff who received an average of 6 weeks of basic obstetric first aid training. Therefore, no improvement was made to the skill level of MIFT staff, but the new staff did help to avoid depleting facilities of their skilled midwives during the transfer of patients (Schoon, 2013).

Friction and misunderstandings between management, dispatchers, EMS personnel, and midwives were inevitable, and extensive discussions had to be held. These discussions were to ensure that the dispatchers understood the need for the obstetric emergency units to be available for MIFTs, as generally they still believed that a patient outside a healthcare facility should have a higher transportation priority than an obstetric patient who needs to be transported between facilities (Schoon, 2013). The findings of this study show that there is a correlation between improvements in MMR and improved IFT.

A.2 Study 2: The implementation status of obstetric EMS systems in selected South African health districts with recommendations for future development (Yancey & Mould-Millman, 2015)

The Yancey & Mould-Millman (2015) study also looked at the De Vries *et al.* (2011) article, which considered the effects that dedicated obstetric EMS units can have on dispatch, response, and mission times. It demonstrated that the effects are extremely beneficial, according to Yancey & Mould-Millman (2015).

Since the De Vries *et al.* (2011) article did not consider the impact of the EMS units on patient outcomes, the Yancey & Mould-Millman (2015) study was undertaken to ascertain its effects on the MMR. The study investigates the present state of maternal and neonatal IFT in two provinces, namely KwaZulu-Natal and Mpumalanga. The results of the investigation in these two provinces were analysed and recommendations for improvements were made in order for the provinces to surpass the Free State's results as shown in the Schoon (2013) study.

From the Schoon (2013) study it was observed that the MMR decrease for the Free State could only be sustained with the addition of dedicated EMS MIFT units. The results of the Schoon (2013) study proposed that government should try to replicate and improve on the study's results in all provinces. The Yancey & Mould-Millman (2015) study shows that further investigations were undertaken in determining the feasibility of implementing the recommendations in other provinces.

The Yancey & Mould-Millman (2015) study followed a predefined step-by-step method. Firstly, a structured, scripted plan for on-site data and information gathering in KwaZulu-Natal and Mpumalanga was designed. Secondly, presentations on the challenges faced in maternal and neonatal EMS care were made by the provincial health districts, and these presentations were audited. Thirdly, data that was collected was analysed, with a focus on maternal emergency status. Finally, the analysed data was used to provide generic recommendations that can be used as a framework upon which to build improvements.

A.2 Study 2: The implementation status of obstetric EMS systems in selected South African health districts with recommendations for future development (Yancey & Mould-Millman, 2015)

A.2.1 Mpumalanga province

The Mpumalanga province consists of three districts with 1 to 1.5 million inhabitants each. The EMS programme for this province is part of the provincial DoH. This structure is viewed as beneficial for the purpose of implementing changes similar to those implemented in the Free State, as it could facilitate the recruitment of EMS as a public health resource and service, as well as potentially facilitate conducting a population health survey through emergency call data and patient evaluation information. According to the data, the highest mortality rate and most urgent type of obstetric emergencies in Mpumalanga are related to conditions such as hypertension, pre-eclampsia, and eclampsia [2] (Yancey & Mould-Millman, 2015).

Based on data collected for the Mpumalanga province in 2013, 20% of the 17,880 emergency calls were obstetric emergency requests. This is the second highest number of calls recorded for a medical emergency type (Yancey & Mould-Millman, 2015).

The current operational and clinical components of the EMS IFT are inadequate. None of the districts make use of scripted protocols and no categorisation guidelines exist for the prioritisation of calls. All ambulances are dispatched from the three districts' ECCs. All the calls, including obstetric calls, are prioritised within a general pool (Yancey & Mould-Millman, 2015). This places unnecessary pressure on the understaffed dispatch centres.

A trial-phase re-arrangement of ambulances has been undertaken by the province, but all the calls are still handled by each district's ECC. The re-arrangement consists of four dedicated obstetric units that are stationed at different locations every three months (Yancey & Mould-Millman, 2015). This arrangement will establish the most time-efficient geographic postings to achieve the shortest response time to obstetric emergencies. It was already determined that the existing four dedicated units are not enough, and twelve more have been ordered (Yancey & Mould-Millman, 2015).

However, this is not the only limiting factor to improving Mpumalanga's MMR, since it is estimated that Mpumalanga requires about 83 paramedics but currently employs only 6. The EMS personnel also do not have scripted critical care protocols for

[2] Pre-eclampsia: the sudden, sharp rise in blood pressure, swelling (generally fluid retention in the face, hands and feet), and excess protein in the urine (Nordqvist, 2012); Eclampsia: a pregnant woman, previously diagnosed with pre-eclampsia, develops seizures, or falls into a coma (Stöppler, 2014).

caring for patients in-transit or for referral to institutions (Yancey & Mould-Millman, 2015).

A.2.2 KwaZulu-Natal province

KwaZulu-Natal consists of eleven health districts (DoH: Province of KwaZulu-Natal, 2001). Each district's problems might vary in some way, but only three of the districts were included in the Yancey & Mould-Millman (2015) study. The general problems identified for the three are quite similar.

The most urgent type of obstetric emergency in KwaZulu-Natal stems from haemorrhage (Yancey & Mould-Millman, 2015), which is an extremely time-sensitive problem. Therefore, the availability and activation of rapid response and short transport times are extremely important to improve KwaZulu-Natal's MMR.

The district ECCs in KwaZulu-Natal do make use of an organised, detailed protocol. This protocol also differentiates between two types of calls; primary and IFT. However, no scripted protocol questionnaire or instructions exists for use by dispatchers. The dispatchers also only receive basic life support level medical training (Yancey & Mould-Millman, 2015).

The same trial-phase re-arrangement of EMS units that was implemented in Mpumalanga was implemented in KwaZulu-Natal's Uthungulu District. The conclusion again was that four dedicated obstetric EMS units are not enough (Yancey & Mould-Millman, 2015).

A.2.3 Recommendations

The Yancey & Mould-Millman (2015) study makes a number of generic recommendations concerning the procedures in the ECC, the obstetric response and transport services, the EMS destination facilities for maternal emergencies, and the EMS quality improvement and educational initiatives in emergency obstetric care. These recommendations are intended to form a framework of improvement that can be employed in any province. All of the recommendations made in the Yancey & Mould-Millman (2015) study are important, but only those that are relevant to this project's research will be mentioned.

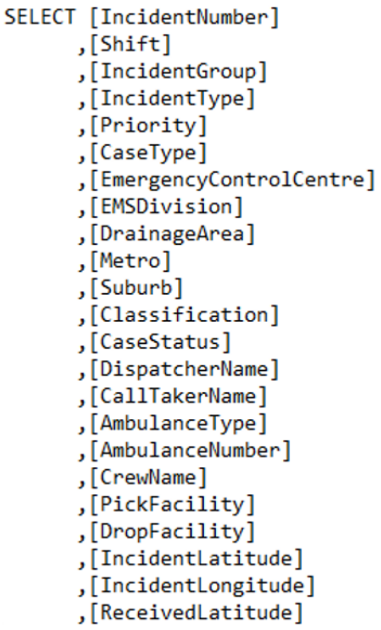
An interesting recommendation was to move the call processing to a provincial level, creating a provincial ECC. The dispatching phase would then still be handled at the district ECC, as it is assumed that the personnel there will be the most familiar with the local geography, traffic, and road patterns. The implementation of this recommendation will also provide a centralised database at provincial level, which will serve public health surveillance. The data can then be analysed to make decisions concerning the allocation of EMS response resource funding, geographic redistribution, and deployment of first response and transport resources (Yancey & Mould-Millman, 2015).

The dispatch process for the dedicated obstetric EMS units, for MIFT, should be controlled by a separate operational division at the district's ECC. This operational division should gather information through standardised protocols, dispatch the ambulances, and record the dispatch, response, and return time.

Appendix B

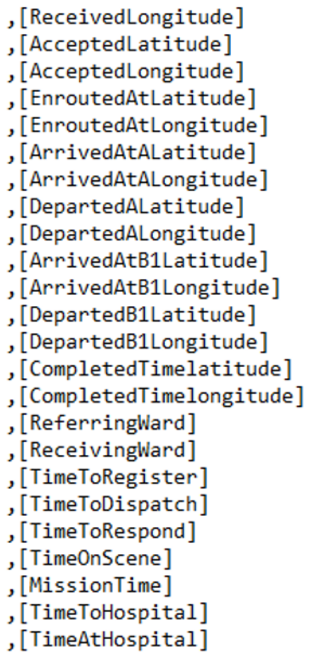
Data collection query

This appendix contains the SQL query that was used by the WC ECC data analyst to get the 6 months' call data from their system. The code is shown in Figures B.1 - B.3.



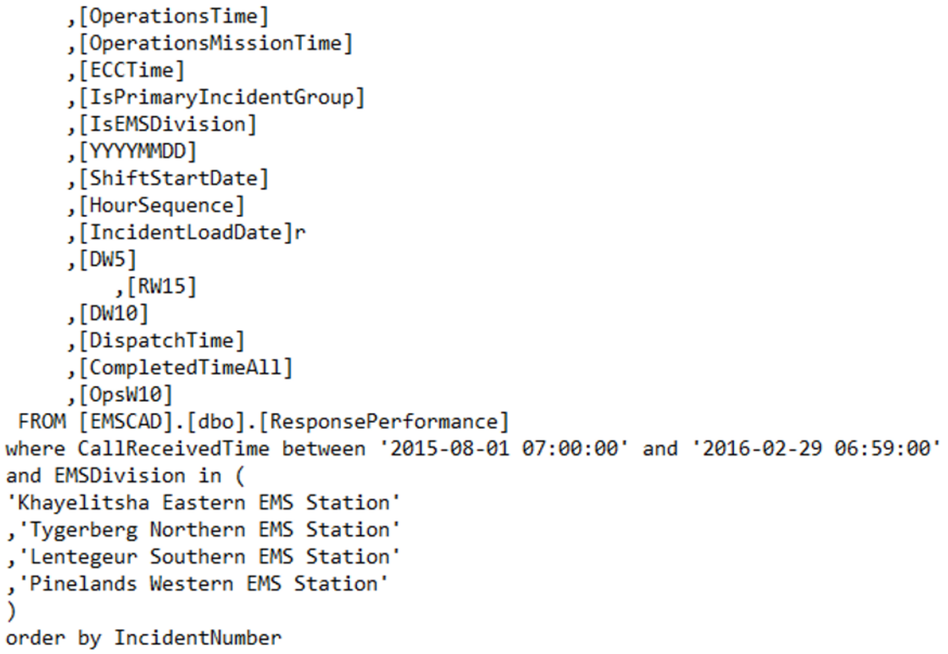
```
SELECT [IncidentNumber]
      ,[Shift]
      ,[IncidentGroup]
      ,[IncidentType]
      ,[Priority]
      ,[CaseType]
      ,[EmergencyControlCentre]
      ,[EMSDivision]
      ,[DrainageArea]
      ,[Metro]
      ,[Suburb]
      ,[Classification]
      ,[CaseStatus]
      ,[DispatcherName]
      ,[CallTakerName]
      ,[AmbulanceType]
      ,[AmbulanceNumber]
      ,[CrewName]
      ,[PickFacility]
      ,[DropFacility]
      ,[IncidentLatitude]
      ,[IncidentLongitude]
      ,[ReceivedLatitude]
```

Figure B.1: Data query part 1.



```
     ,[ReceivedLongitude]
     ,[AcceptedLatitude]
     ,[AcceptedLongitude]
     ,[EnroutedAtLatitude]
     ,[EnroutedAtLongitude]
     ,[ArrivedAtALatitude]
     ,[ArrivedAtALongitude]
     ,[DepartedALatitude]
     ,[DepartedALongitude]
     ,[ArrivedAtB1Latitude]
     ,[ArrivedAtB1Longitude]
     ,[DepartedB1Latitude]
     ,[DepartedB1Longitude]
     ,[CompletedTimelatitude]
     ,[CompletedTimelongitude]
     ,[ReferringWard]
     ,[ReceivingWard]
     ,[TimeToRegister]
     ,[TimeToDispatch]
     ,[TimeToRespond]
     ,[TimeOnScene]
     ,[MissionTime]
     ,[TimeToHospital]
     ,[TimeAtHospital]
```

Figure B.2: Data query part 2.



```
     ,[OperationsTime]
     ,[OperationsMissionTime]
     ,[ECCTime]
     ,[IsPrimaryIncidentGroup]
     ,[IsEMSDivision]
     ,[YYYYMMDD]
     ,[ShiftStartDate]
     ,[HourSequence]
     ,[IncidentLoadDate]r
     ,[DW5]
         ,[RW15]
     ,[DW10]
     ,[DispatchTime]
     ,[CompletedTimeAll]
     ,[OpsW10]
 FROM [EMSCAD].[dbo].[ResponsePerformance]
where CallReceivedTime between '2015-08-01 07:00:00' and '2016-02-29 06:59:00'
and EMSDivision in (
'Khayelitsha Eastern EMS Station'
,'Tygerberg Northern EMS Station'
,'Lentegeur Southern EMS Station'
,'Pinelands Western EMS Station'
)
order by IncidentNumber
```

Figure B.3: Data query part 3.

Appendix C

Dispatch policies

In this appendix Tables C.1, C.2, and C.3 show the advantages and disadvantages of the dispatch policies mentioned in Section 2.4.2, along with some that were not mentioned.

The most prevalent dispatch policy in ECC is closest dispatch, since the objective is generally to minimise the response time (Lim *et al.*, 2011; Poulton & Roussos, 2013). The closest dispatch policy does not always result in the best choice according to Andersson & Värbrand (2006) and Schmid (2012).

Consider the existence of two ambulances, A and B, that both have equally large areas of responsibility. However, ambulance A's area has a higher call frequency. If another call from ambulance A's area where to be received the mean response time would be lower if B were allowed to respond to some of the calls for which A is the closer ambulance. The reason for this is that the probability of another call coming from ambulance A's area is higher than for ambulance B's area, meaning that ambulance A would then be available to respond to the other call (Andersson & Värbrand, 2006).

Table C.1: Dispatch Policy 1.

DISPATCH	**ADVANTAGES**	**DISADVANTAGES**
First-in first-out	Simplest call sorting method. Requires minimum resource allocation in ECC	Lacking in optimisation for higher priority. calls.
Priority	Most commonly used call sorting method. Optimisation for higher priority calls.	Can affect response time of lower priority calls. Requires call resorting when a new call is added in the waiting queue.
Closest	Most commonly used ambulance assignment method, where the best idle unit is dispatched with the fastest response time. Can be exploited to optimise higher priority calls by using closest (tiered) dispatch after call prioritising.	Requires mobile data terminal or standard radio communication to be installed in the ambulances if the ambulances are placed at strategic location sites.
Non-closest (coverage combined with probability or preparedness)	Ambulances are reserved for expected future calls and can thus maximise EMS coverage.	May cause legal action for not dispatching the closest available ambulance. Requires more complicated probability or preparedness evaluation (Relative to closest dispatch). Requires mobile data terminal or standard radio communication to be installed inn the ambulances if the ambulances are placed at strategic location sites.

Table C.2: Dispatch Policy 2.

DISPATCH	**ADVANTAGES**	**DISADVANTAGES**
Reroute enabled	Optimisation for higher priority calls.	Higher ambulance travelling cost due to rerouting. Can affect response time of lower priority calls. Disturbance to ambulance crew. Requires mobile data terminal or standard radio communication to be installed in the ambulances. Requires extra resources to implement add-on dispatch.
Priority update enabled	Increases the efficiency of ambulance utilisation as the status of a call can be upgraded, downgraded or cancelled.	Disturbance to ambulance crew. Requires mobile data terminal or standard radio communication to be installed in the ambulances. Requires extra resources to implement add-on dispatch.
Pre-arrival instructions	Provides an aid to the victim prior to ambulance arrival and can thus reduce mortality rate.	Requires extra resources to implement add-on dispatch.

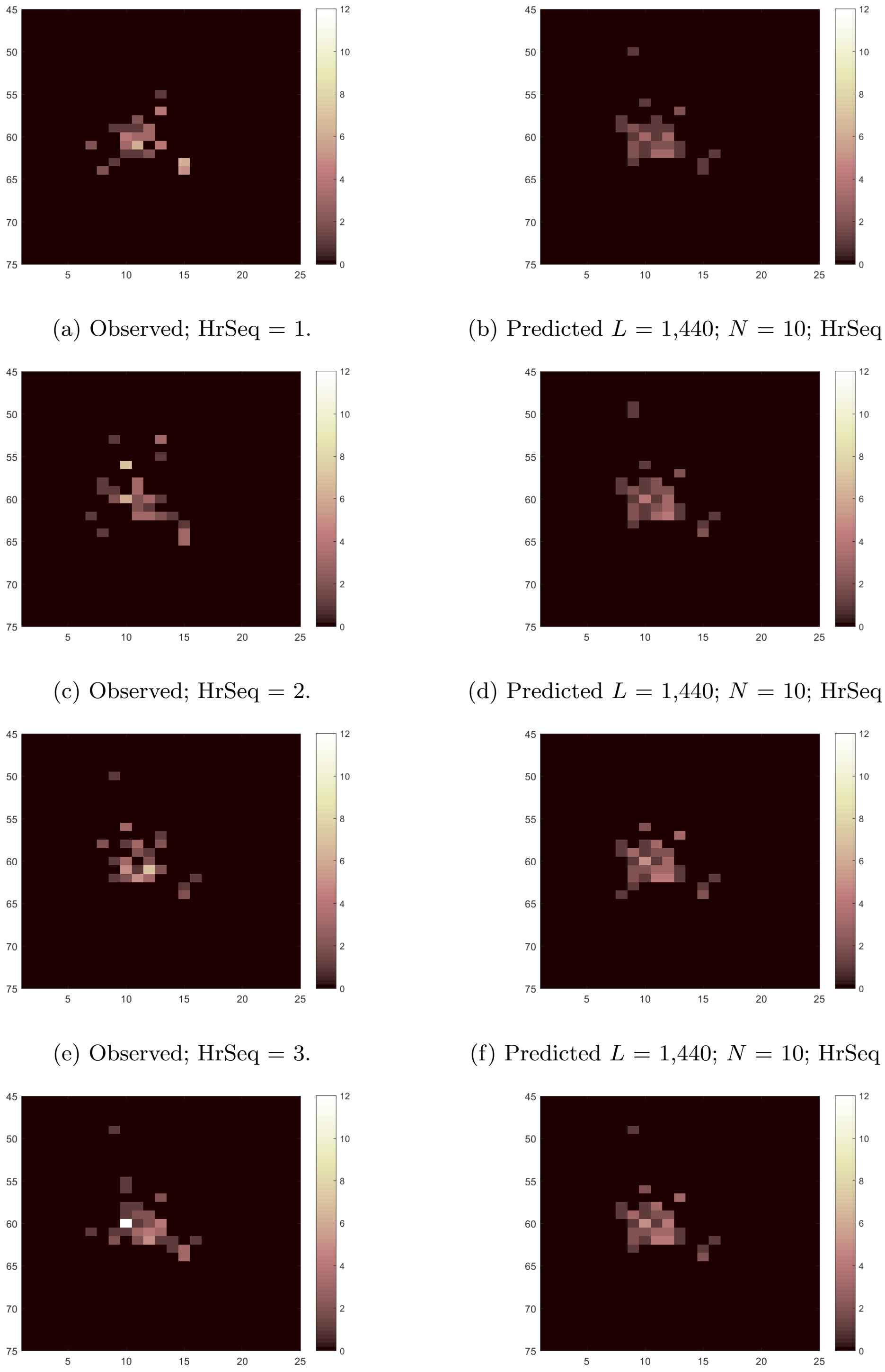
Table C.3: Dispatch Policy 3.

DISPATCH	ADVANTAGES	DISADVANTAGES
Pseudo priority	The lower priority calls with long waiting time can be shortened.	Can affect response time of higher priority calls. Requires extra resources to implement add-on dispatch.
GIS support	Increases the efficiency of ambulance utilisation through instant status and location updates.	Requires extra resources to implement add-on dispatch.
Free ambulance exploitation	Optimisation for higher priority calls.	Can affect response time of lower priority calls. Disturbance to ambulance crew. Requires mobile data terminal or standard radio communication to be installed in the ambulances. Requires extra resources to implement add-on dispatch.

Appendix D

Heat maps: Scenario 1

This appendix contains the figures depicting the heat maps for 20160101's first twelve hours for the four combinations of demand nodes, L, and N value discussed in Section 5.5 for Scenario 1. The 5,625 demand nodes, $L = 1440$, and $N = 10$ combination's heat maps are shown in Figures D.1 - D.3. The 10,000 demand nodes, $L = 720$, and $N = 8$ combination's heat maps are shown in Figures D.4 - D.6. The 10,000 demand nodes, $L = 720$, and $N = 20$ combination's heat maps are shown in Figures D.7 - D.9. The 10,000 demand nodes, $L = 720$, and $N = 80$ combination's heat maps are shown in Figures D.10 - D.12.



(a) Observed; HrSeq = 1.

(b) Predicted L = 1,440; N = 10; HrSeq = 1.

(c) Observed; HrSeq = 2.

(d) Predicted L = 1,440; N = 10; HrSeq = 2.

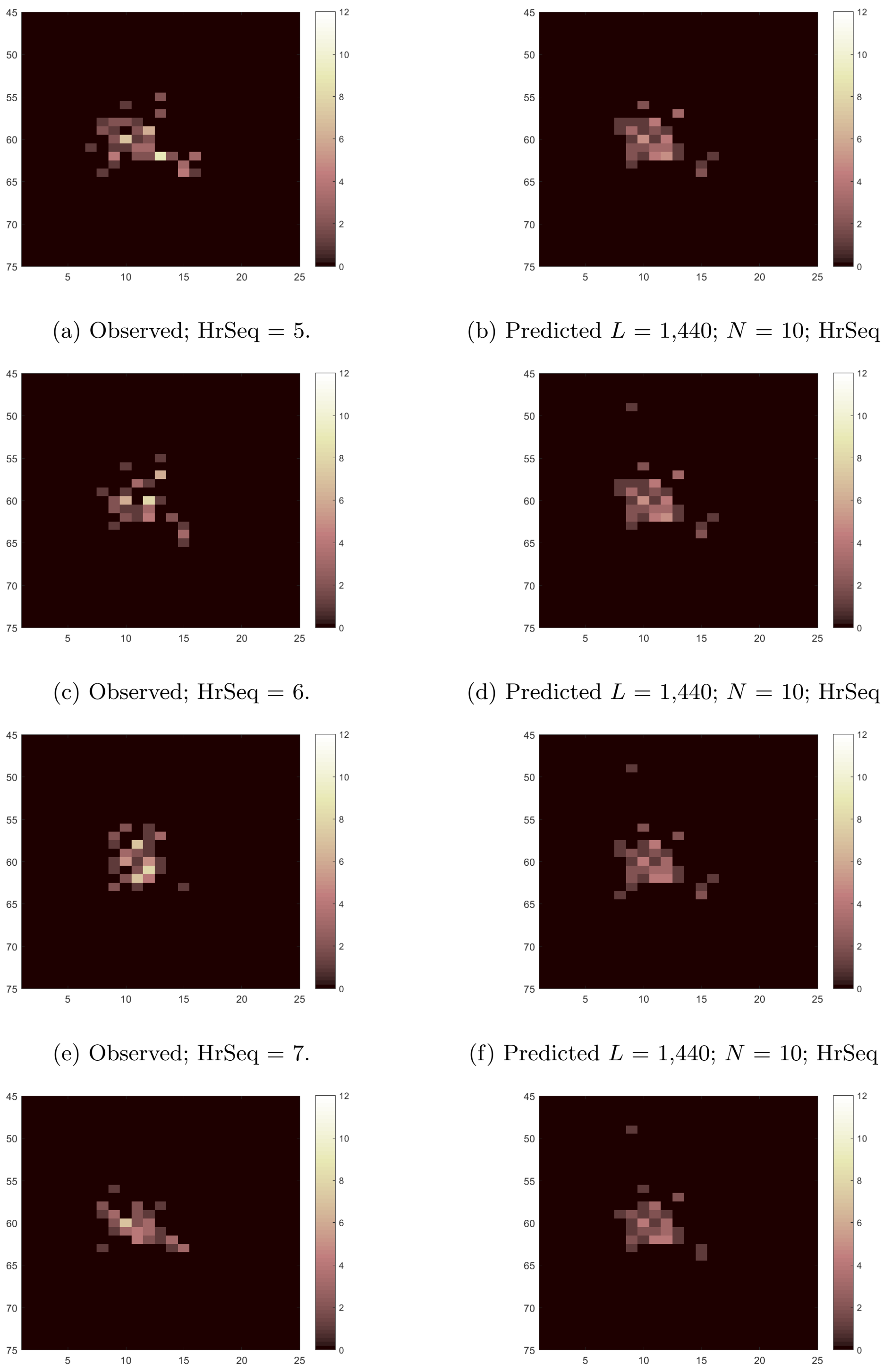
(e) Observed; HrSeq = 3.

(f) Predicted L = 1,440; N = 10; HrSeq = 3.

(g) Observed; HrSeq = 4.

(h) Predicted L = 1,440; N = 10; HrSeq = 4.

Figure D.1: Heat maps HrSeq 1 - 4: Nodes 5,625 Observed VS Predicted L = 1,440; N = 10.



(a) Observed; HrSeq = 5.

(b) Predicted L = 1,440; N = 10; HrSeq = 5.

(c) Observed; HrSeq = 6.

(d) Predicted L = 1,440; N = 10; HrSeq = 6.

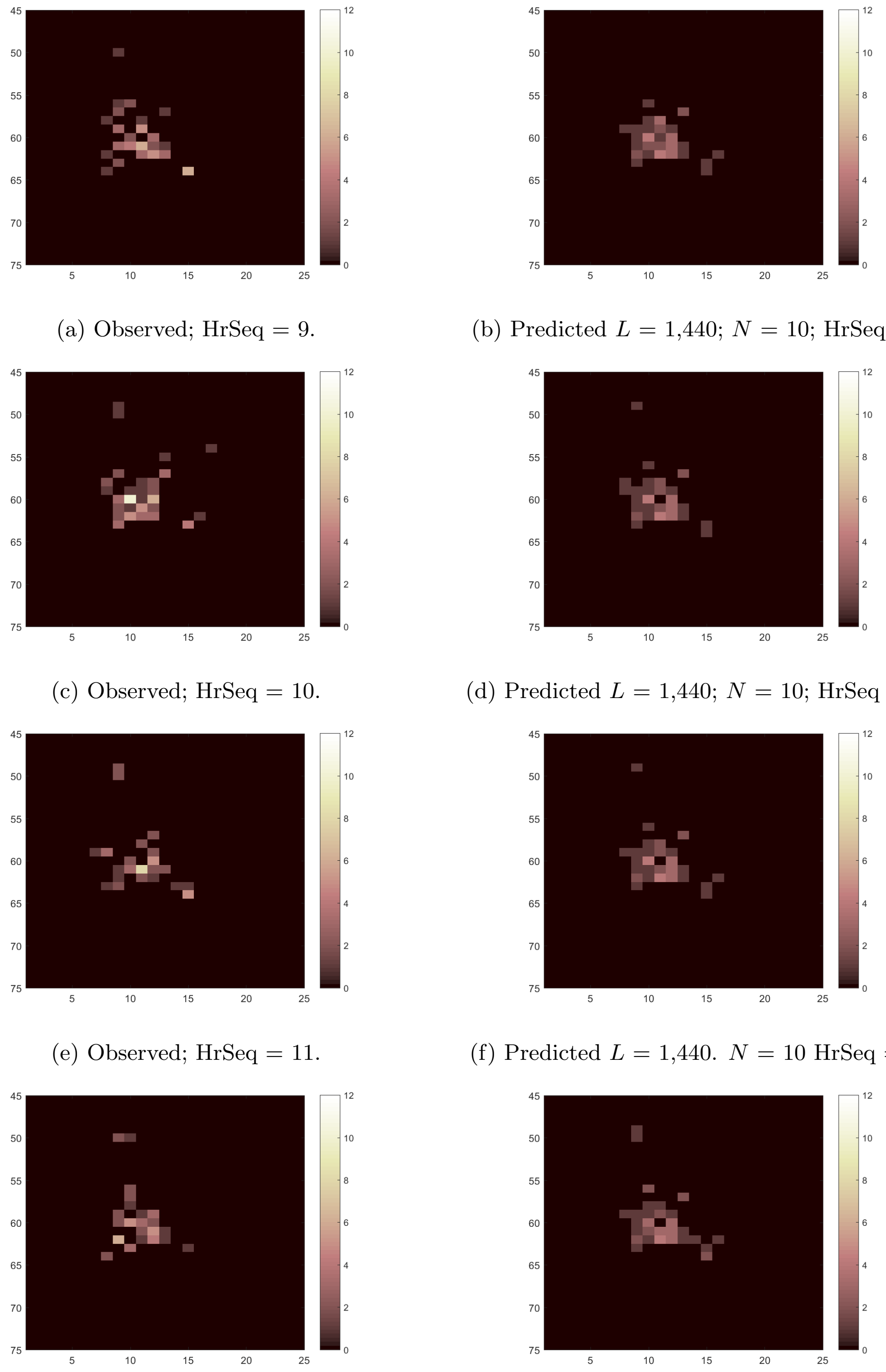
(e) Observed; HrSeq = 7.

(f) Predicted L = 1,440; N = 10; HrSeq = 7.

(g) Observed; HrSeq = 8.

(h) Predicted L = 1,440; N = 10; HrSeq = 8.

Figure D.2: Heat maps HrSeq 5 - 8: Nodes 5,625 Observed VS Predicted L = 1,440; N = 10.



(a) Observed; HrSeq = 9.

(b) Predicted L = 1,440; N = 10; HrSeq = 9.

(c) Observed; HrSeq = 10.

(d) Predicted L = 1,440; N = 10; HrSeq = 10.

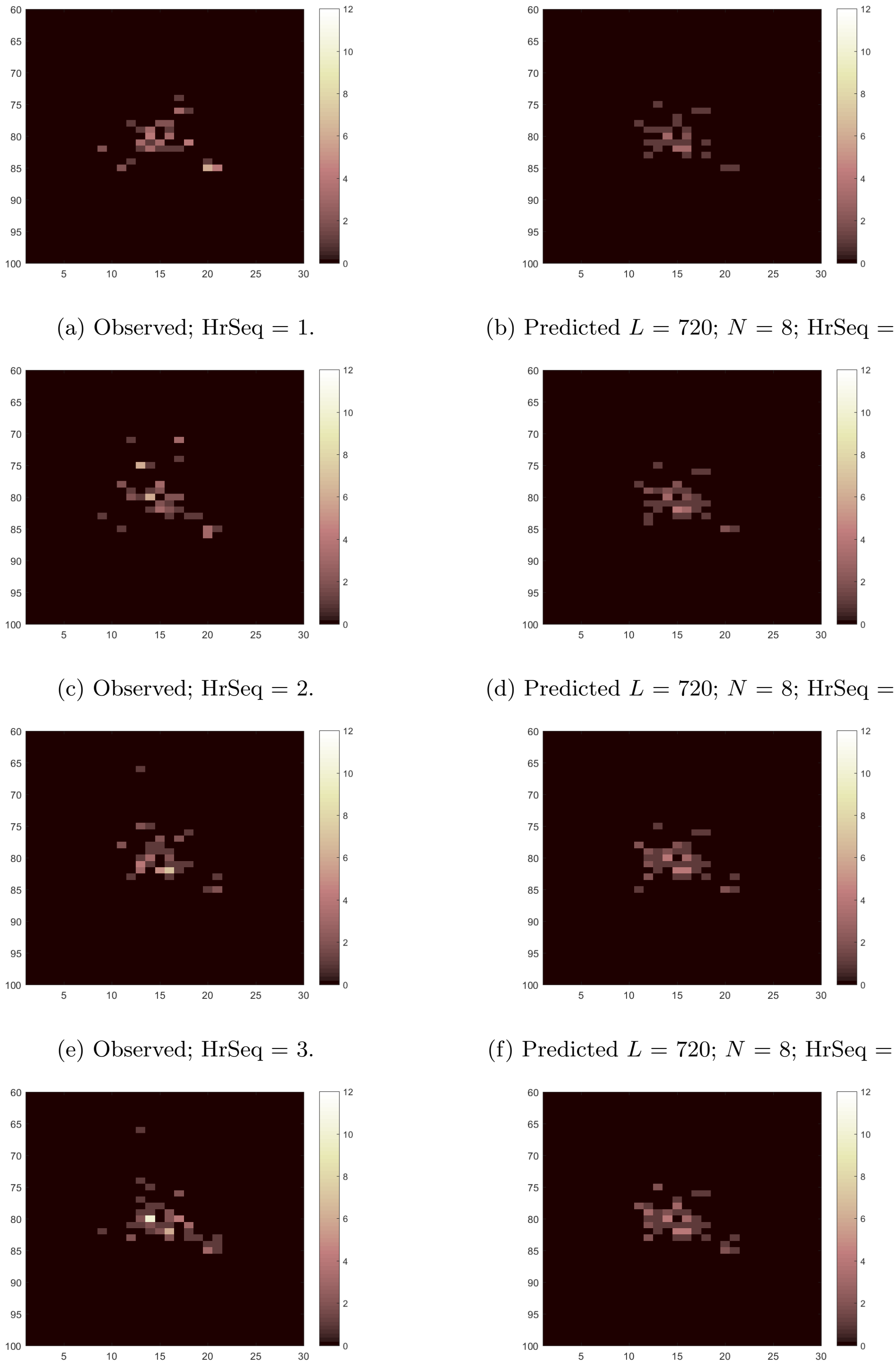
(e) Observed; HrSeq = 11.

(f) Predicted L = 1,440. N = 10 HrSeq = 11.

(g) Observed; HrSeq = 12.

(h) Predicted L = 1,440; N = 10; HrSeq = 12.

Figure D.3: Heat maps HrSeq 9 - 12: Nodes 5,625 Observed VS Predicted L = 1,440; N = 10.



(a) Observed; HrSeq = 1.

(b) Predicted L = 720; N = 8; HrSeq = 1.

(c) Observed; HrSeq = 2.

(d) Predicted L = 720; N = 8; HrSeq = 2.

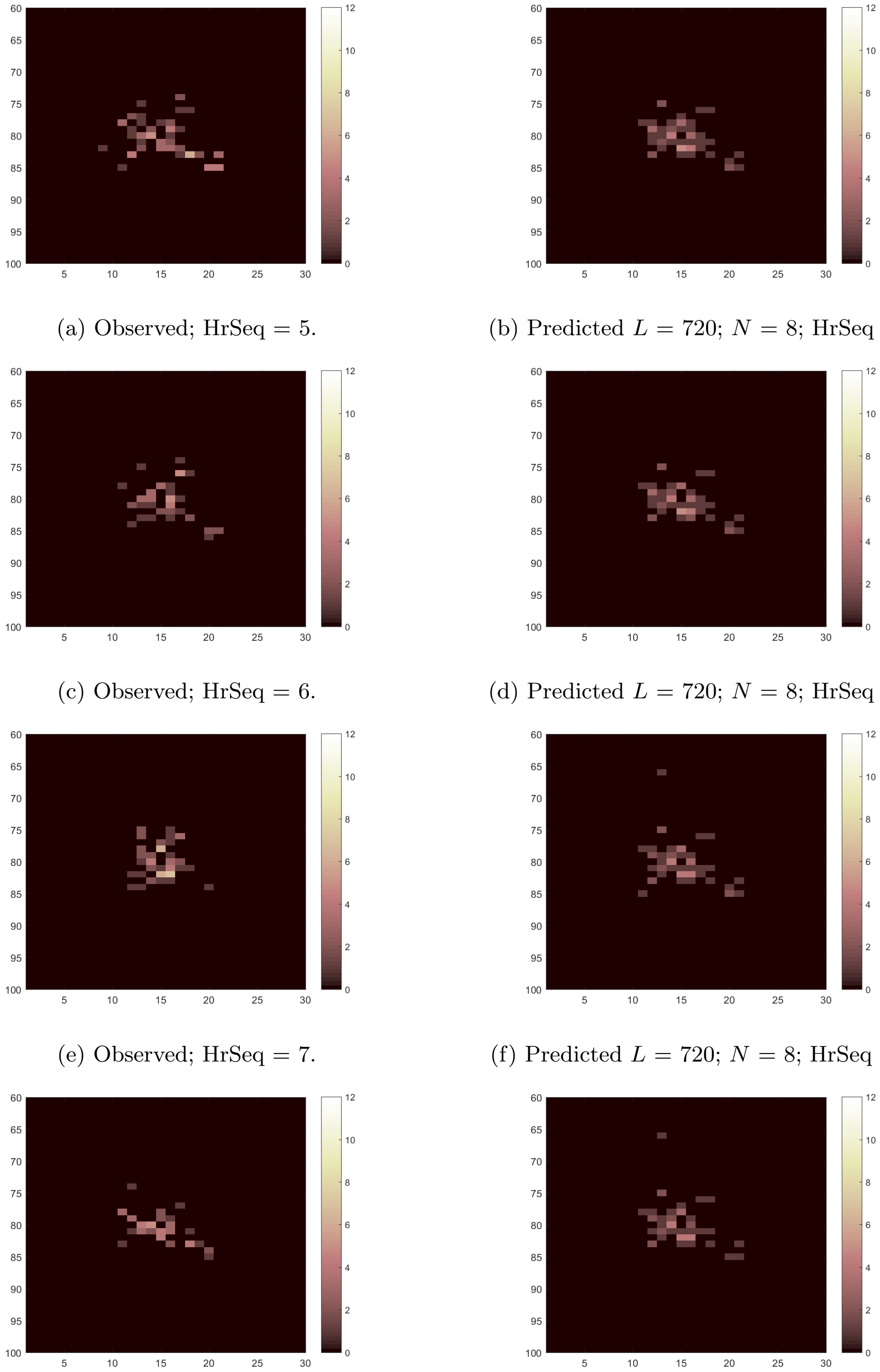
(e) Observed; HrSeq = 3.

(f) Predicted L = 720; N = 8; HrSeq = 3.

(g) Observed; HrSeq = 4.

(h) Predicted L = 720; N = 8; HrSeq = 4.

Figure D.4: Heat maps HrSeq 1 - 4: Nodes 10,000 Observed VS Predicted L = 720; N = 8.



(a) Observed; HrSeq = 5.

(b) Predicted L = 720; N = 8; HrSeq = 5.

(c) Observed; HrSeq = 6.

(d) Predicted L = 720; N = 8; HrSeq = 6.

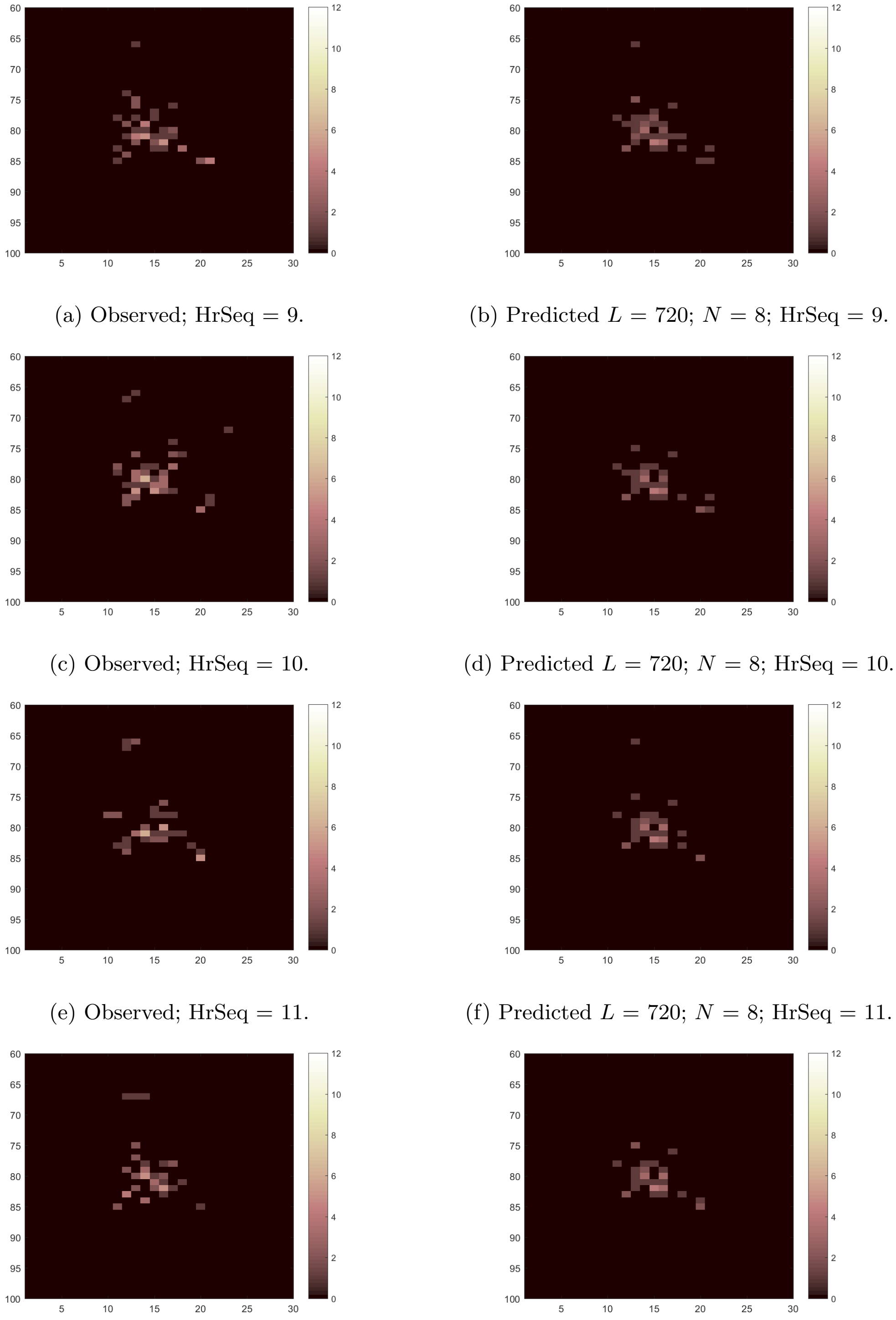
(e) Observed; HrSeq = 7.

(f) Predicted L = 720; N = 8; HrSeq = 7.

(g) Observed; HrSeq = 8.

(h) Predicted L = 720; N = 8; HrSeq = 8.

Figure D.5: Heat maps HrSeq 5 - 8: Nodes 10,000 Observed VS Predicted L = 720; N = 8.



(a) Observed; HrSeq = 9.

(b) Predicted $L = 720$; $N = 8$; HrSeq = 9.

(c) Observed; HrSeq = 10.

(d) Predicted $L = 720$; $N = 8$; HrSeq = 10.

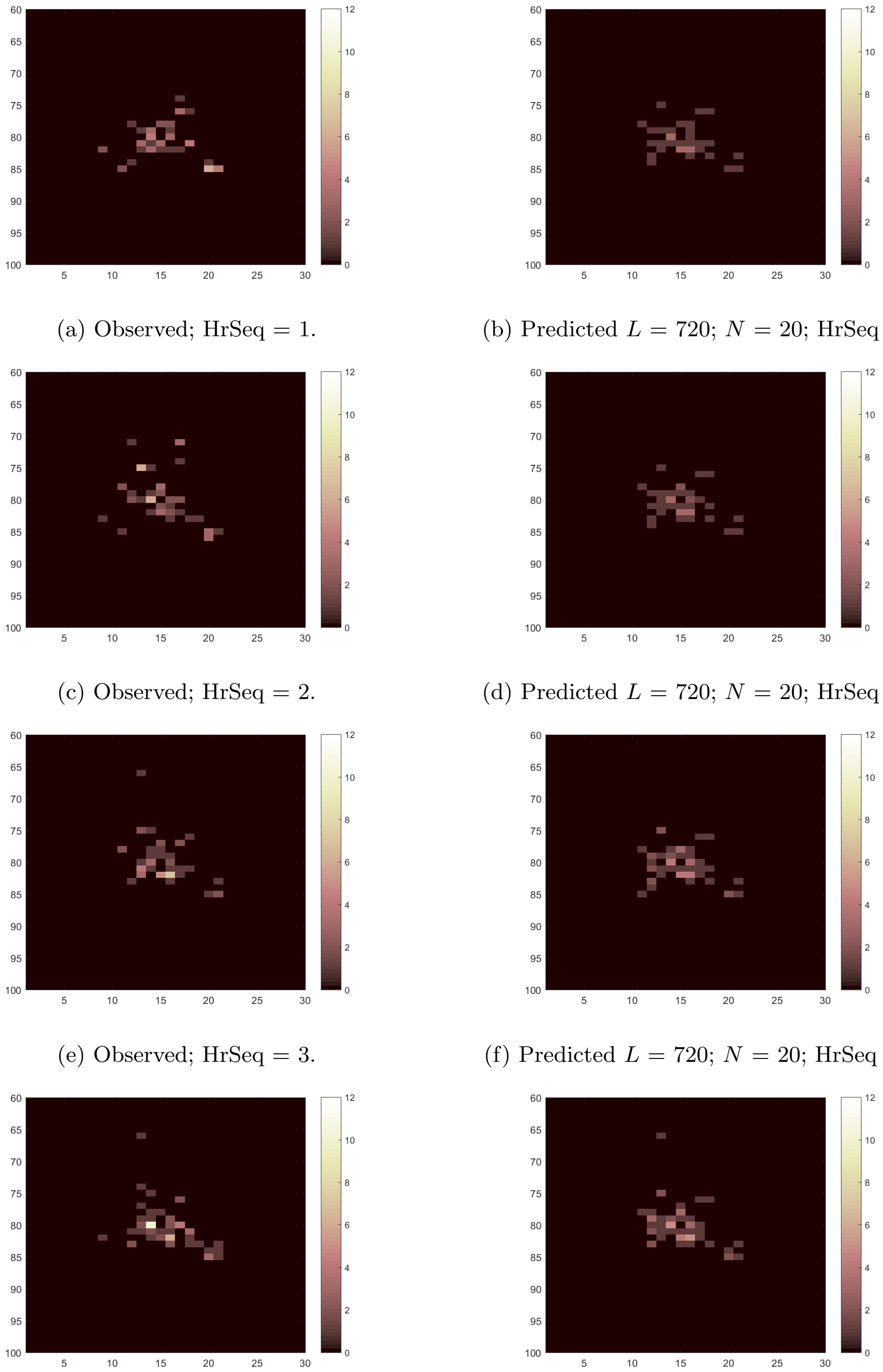
(e) Observed; HrSeq = 11.

(f) Predicted $L = 720$; $N = 8$; HrSeq = 11.

(g) Observed; HrSeq = 12.

(h) Predicted $L = 720$; $N = 8$; HrSeq = 12.

Figure D.6: Heat maps HrSeq 9 - 12: Nodes 10,000 Observed VS Predicted $L = 720$; $N = 8$.



(a) Observed; HrSeq = 1.

(b) Predicted L = 720; N = 20; HrSeq = 1.

(c) Observed; HrSeq = 2.

(d) Predicted L = 720; N = 20; HrSeq = 2.

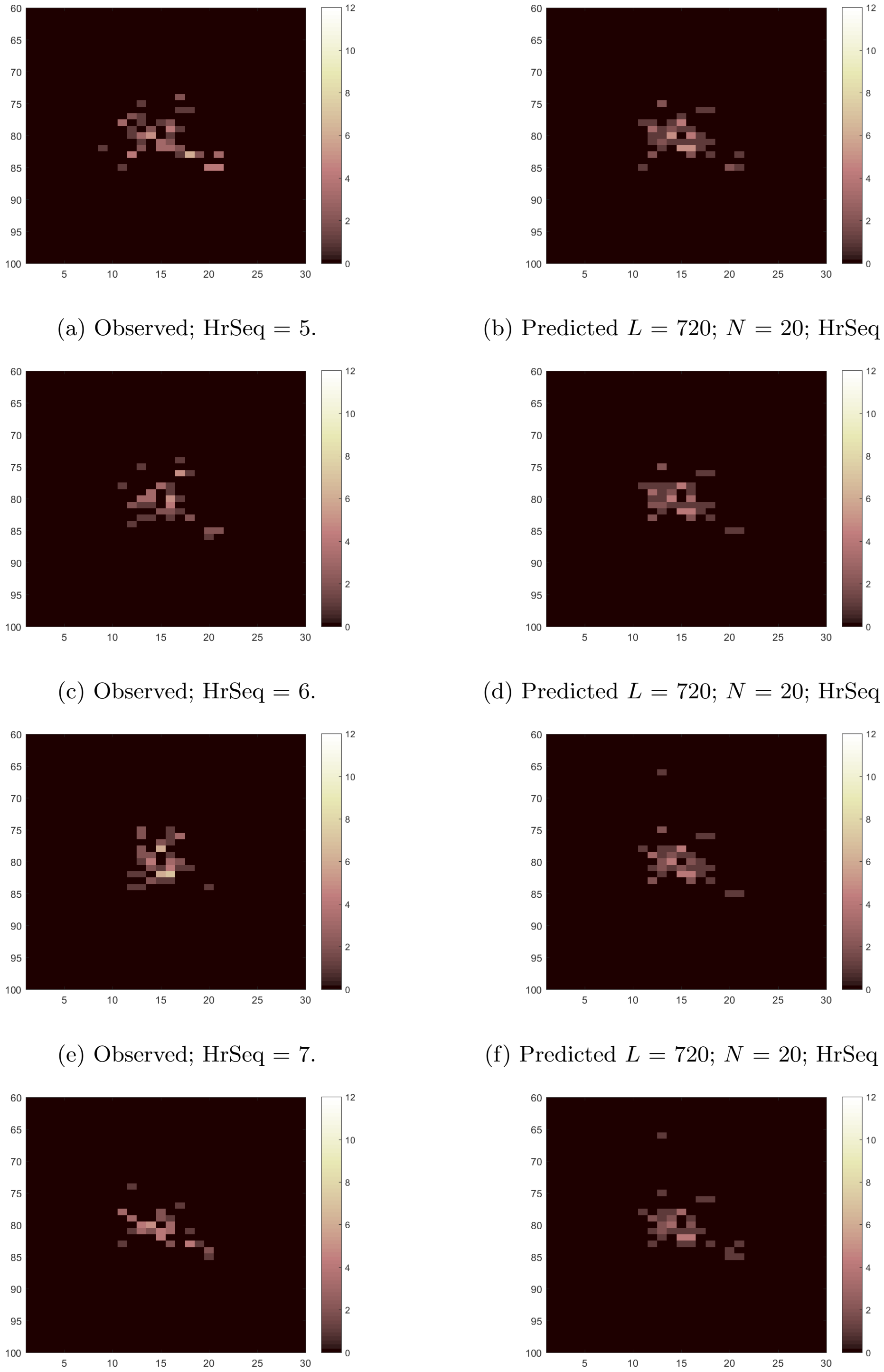
(e) Observed; HrSeq = 3.

(f) Predicted L = 720; N = 20; HrSeq = 3.

(g) Observed; HrSeq = 4.

(h) Predicted L = 720; N = 20; HrSeq = 4.

Figure D.7: Heat maps HrSeq 1 - 4: Nodes 10,000 Observed VS Predicted L = 720; N = 20.



(a) Observed; HrSeq = 5.

(b) Predicted L = 720; N = 20; HrSeq = 5.

(c) Observed; HrSeq = 6.

(d) Predicted L = 720; N = 20; HrSeq = 6.

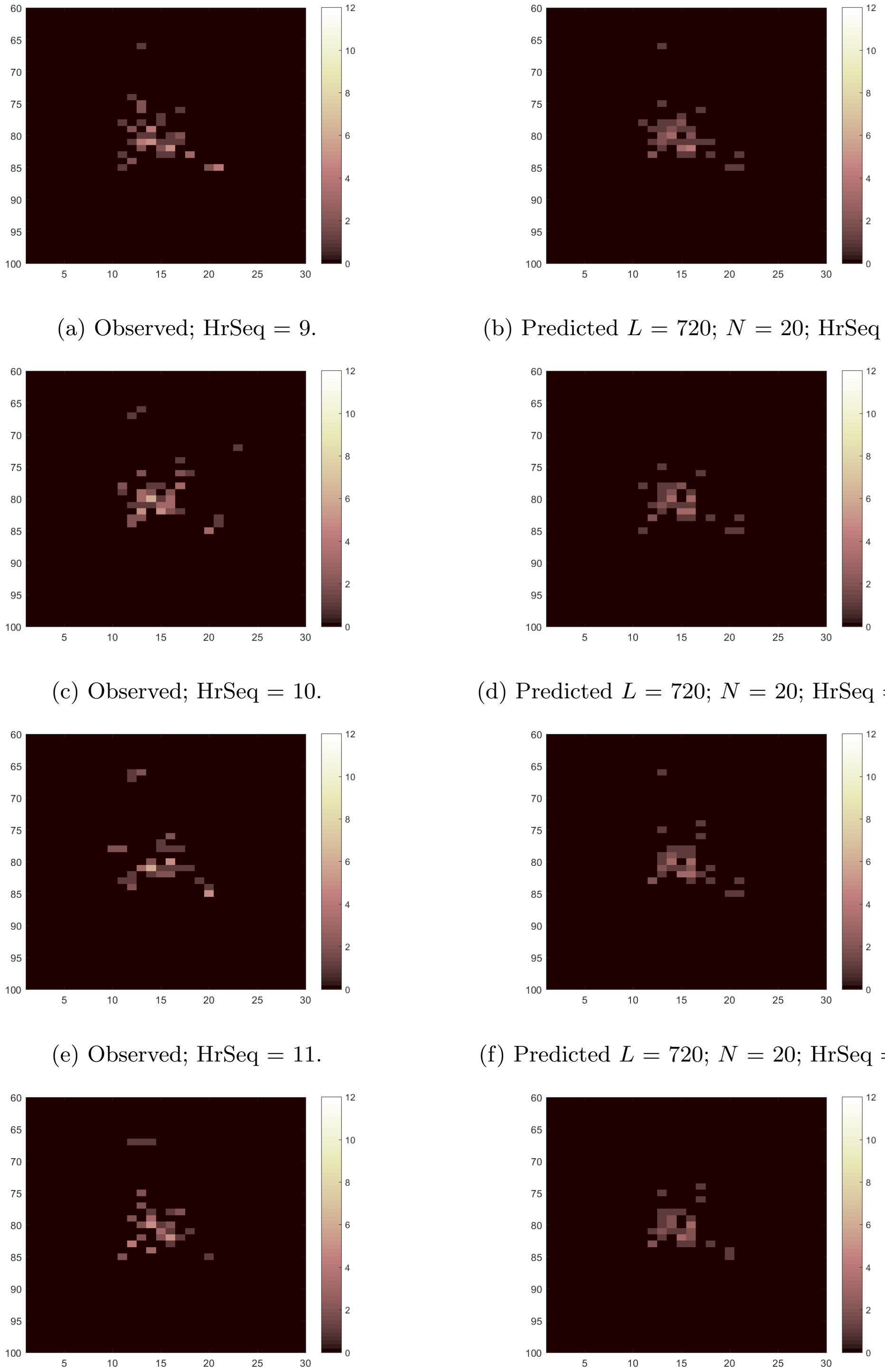
(e) Observed; HrSeq = 7.

(f) Predicted L = 720; N = 20; HrSeq = 7.

(g) Observed; HrSeq = 8.

(h) Predicted L = 720; N = 20; HrSeq = 8.

Figure D.8: Heat maps HrSeq 5 - 8: Nodes 10,000 Observed VS Predicted L = 720; N = 20.



(a) Observed; HrSeq = 9.

(b) Predicted L = 720; N = 20; HrSeq = 9.

(c) Observed; HrSeq = 10.

(d) Predicted L = 720; N = 20; HrSeq = 10.

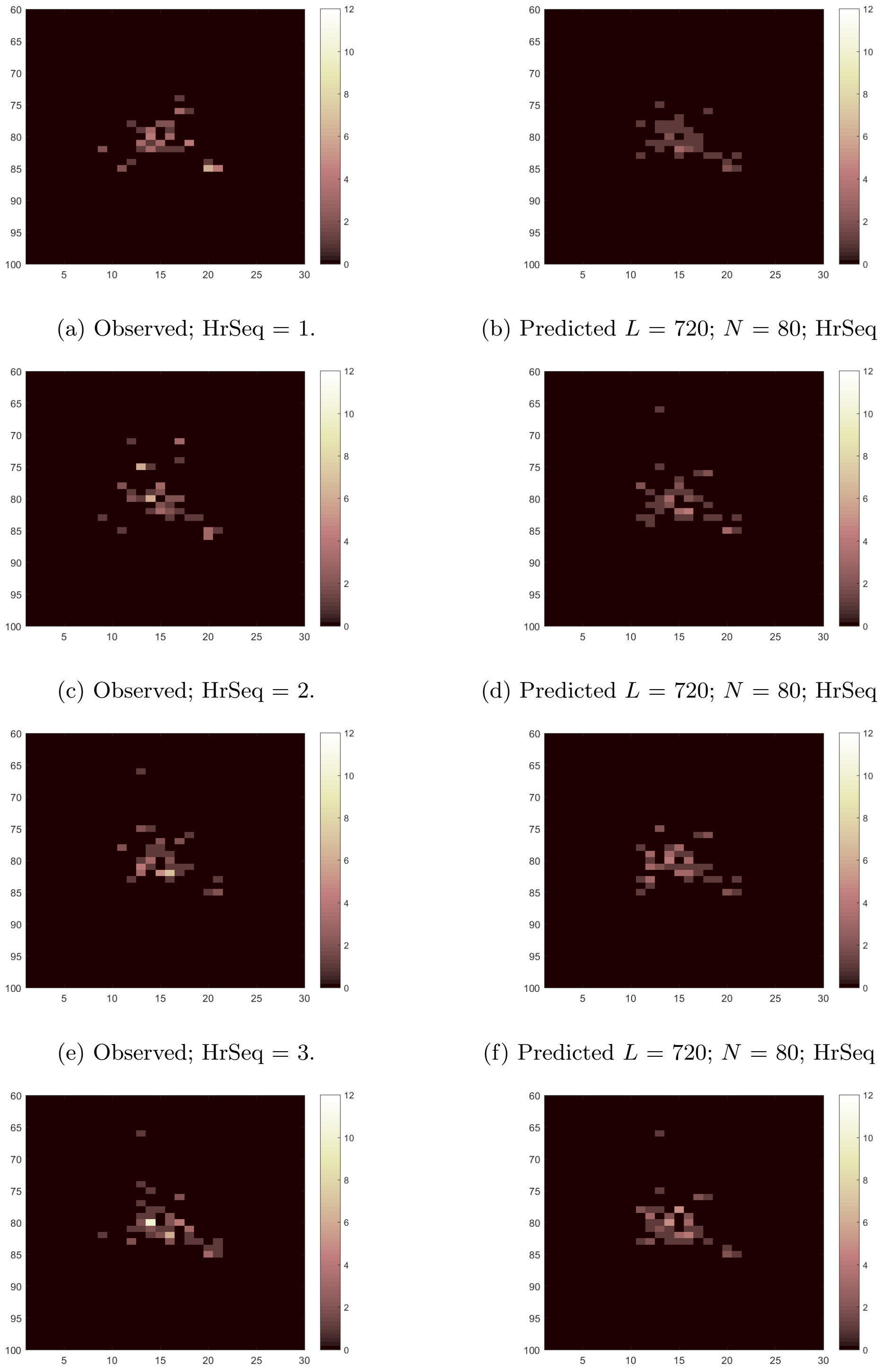
(e) Observed; HrSeq = 11.

(f) Predicted L = 720; N = 20; HrSeq = 11.

(g) Observed; HrSeq = 12.

(h) Predicted L = 720; N = 20; HrSeq = 12.

Figure D.9: Heat maps HrSeq 9 - 12: Nodes 10,000 Observed VS Predicted L = 720; N = 20.



(a) Observed; HrSeq = 1.

(b) Predicted L = 720; N = 80; HrSeq = 1.

(c) Observed; HrSeq = 2.

(d) Predicted L = 720; N = 80; HrSeq = 2.

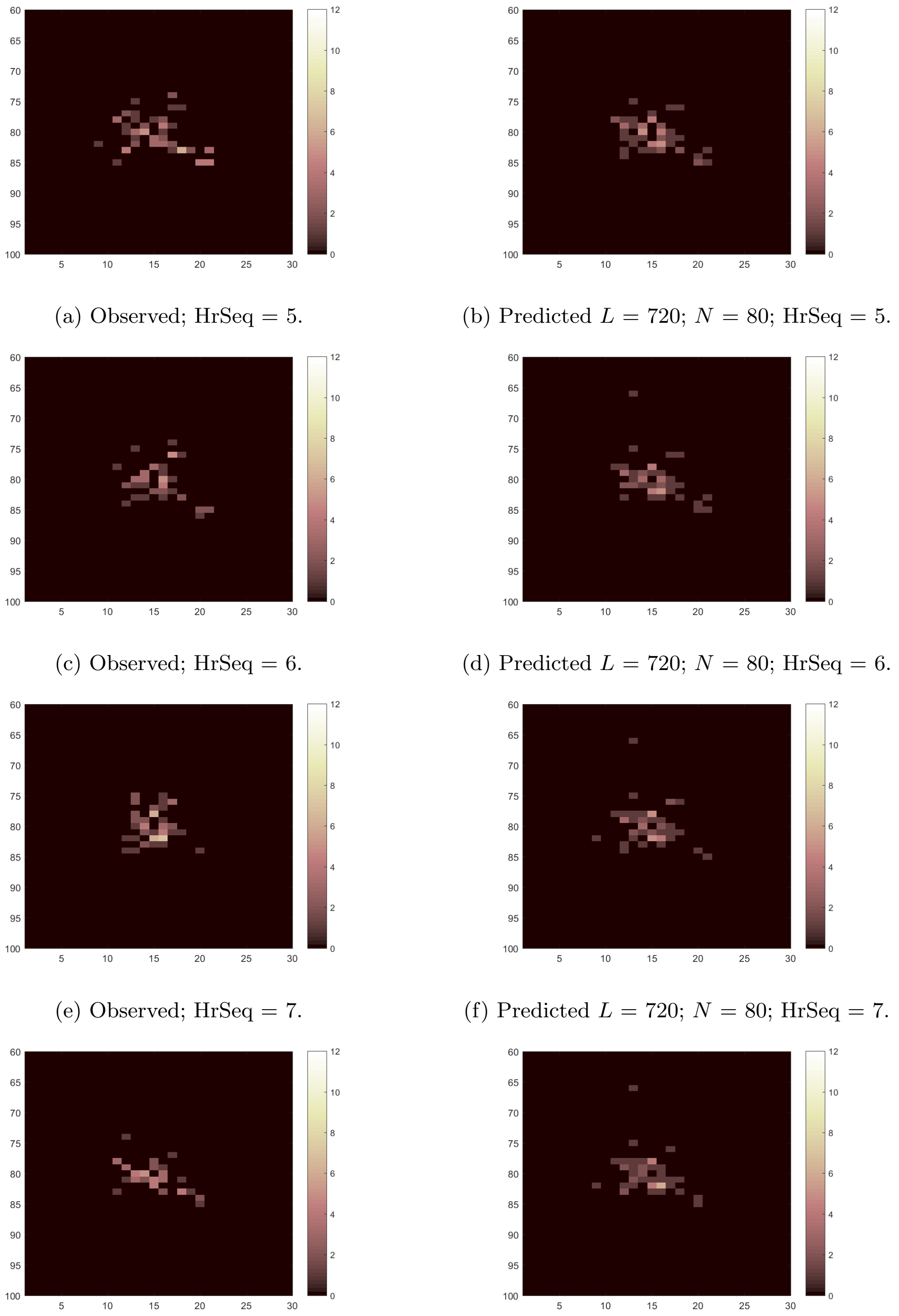
(e) Observed; HrSeq = 3.

(f) Predicted L = 720; N = 80; HrSeq = 3.

(g) Observed; HrSeq = 4.

(h) Predicted L = 720; N = 80; HrSeq = 4.

Figure D.10: Heat maps HrSeq 1 - 4: Nodes 10,000 Observed VS Predicted L = 720; N = 80.



(a) Observed; HrSeq = 5.

(b) Predicted L = 720; N = 80; HrSeq = 5.

(c) Observed; HrSeq = 6.

(d) Predicted L = 720; N = 80; HrSeq = 6.

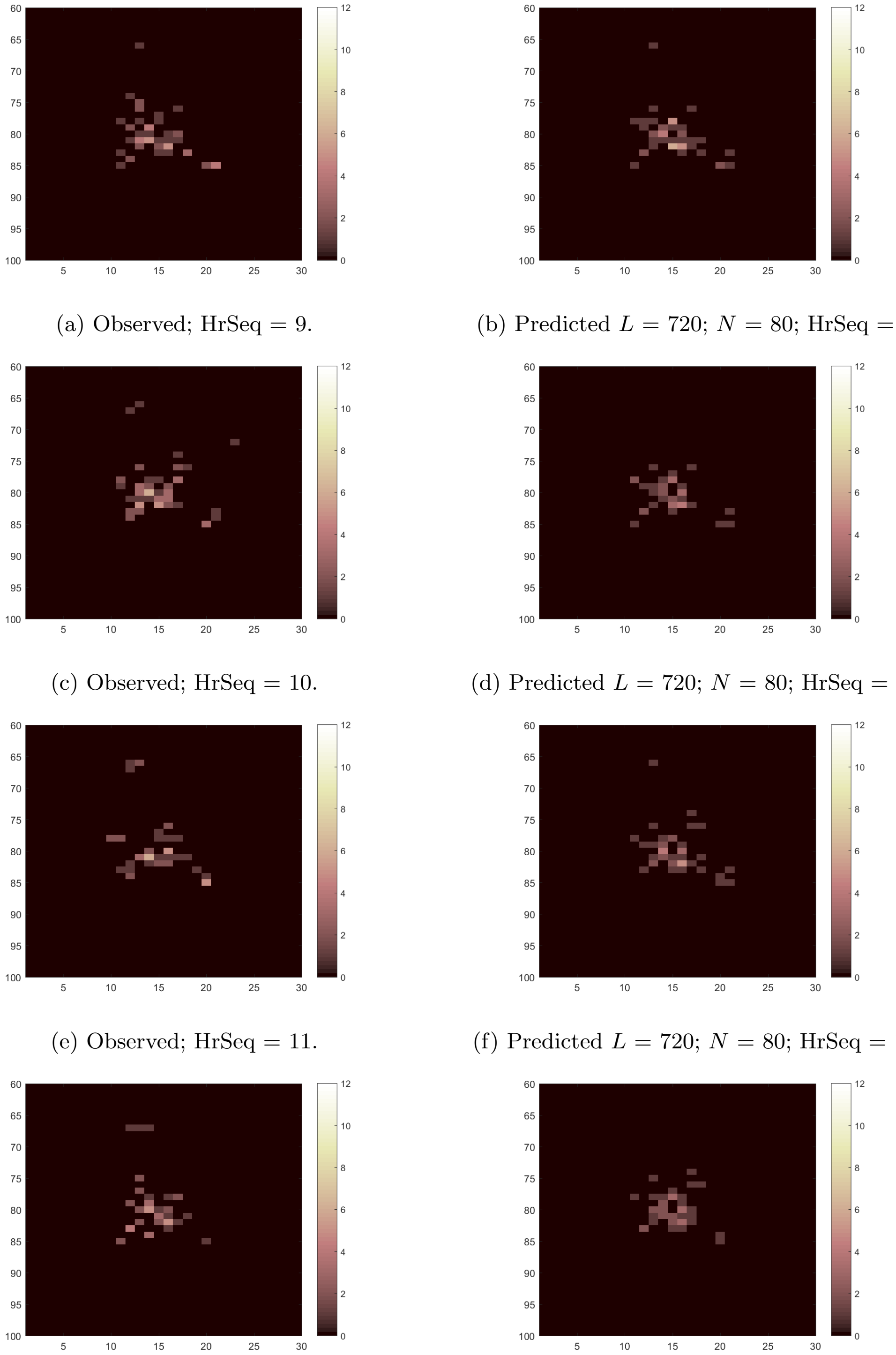
(e) Observed; HrSeq = 7.

(f) Predicted L = 720; N = 80; HrSeq = 7.

(g) Observed; HrSeq = 8.

(h) Predicted L = 720; N = 80; HrSeq = 8.

Figure D.11: Heat maps HrSeq 5 - 8: Nodes 10,000 Observed VS Predicted L = 720; N = 80.



(a) Observed; HrSeq = 9.

(b) Predicted L = 720; N = 80; HrSeq = 9.

(c) Observed; HrSeq = 10.

(d) Predicted L = 720; N = 80; HrSeq = 10.

(e) Observed; HrSeq = 11.

(f) Predicted L = 720; N = 80; HrSeq = 11.

(g) Observed; HrSeq = 12.

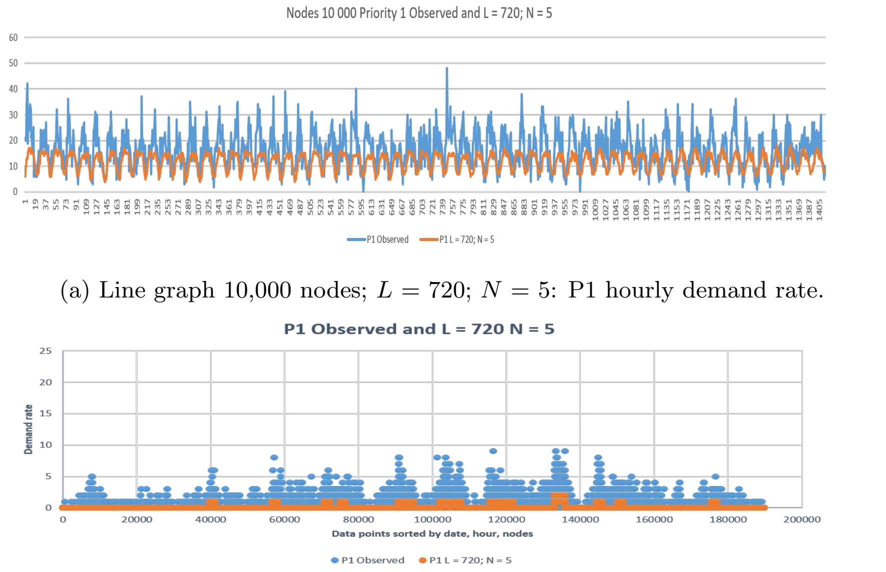
(h) Predicted L = 720; N = 80; HrSeq = 12.

Figure D.12: Heat maps HrSeq 9 - 12: Nodes 10,000 Observed VS Predicted L = 720; N = 80.

Appendix E

Forecasting graphs: Scenario 2

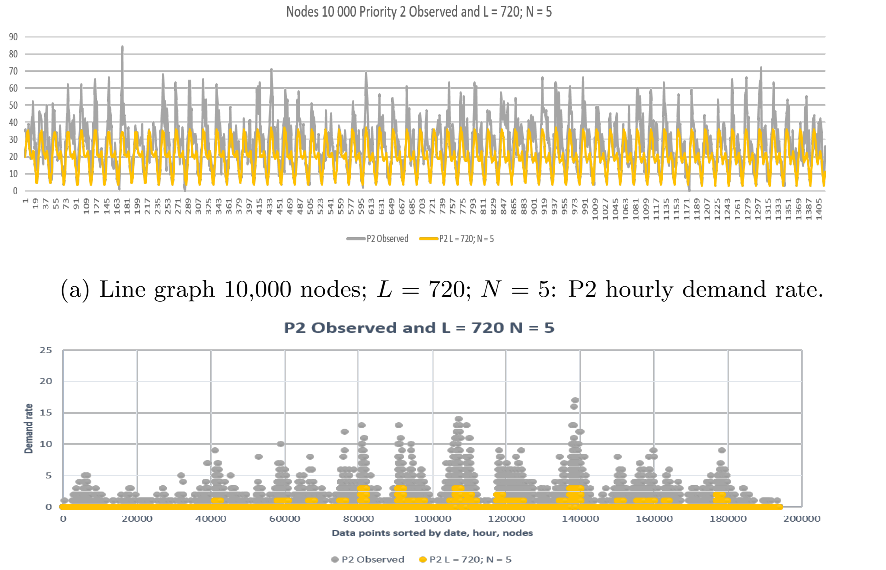
In Section 6.3 the reason for only testing with 10,000 demand nodes and an expected monthly periodicity, i.e. $L = 720$, for Scenario 2 forecasting was described. Table 6.1 contains the RMSE values for P1 and P2 forecasting with $N = 5, 10, 20, 30, 40$, and 60. The best RMSE values, i.e. the lowest, were seen with $N = 5$, but visual representations were still needed in order to make a final decision. The decision was made to create line and scatter graphs for only the four N values which produced the best RMSE values, i.e. $N = 5, 10, 20$, and 30. The line and scatter graphs for forecasting for P1 and P2 calls with 10,000 demand nodes and an expected monthly periodicity, i.e. $L = 720$ for scenario 2 are shown in Figures E.1 - E.8 for $N = 5, 10, 20$, and 30. From these figures it is evident that the forecasting for P1 and P2 follows the observed demand rates for P1 and P2 best with $N = 30$ in terms of time and location. The researcher decided to only create heat maps for $N = 30$ to substantiate the findings from the line and scatter graphs. The heat maps for P1 can be seen in Figures E.9 - E.11. The heat maps for P2 can be seen in Figures E.12 - E.14.



(a) Line graph 10,000 nodes; $L = 720$; $N = 5$: P1 hourly demand rate.

(b) Scatter graph 10,000 nodes; $L = 720$; $N = 5$: P1 demand rate per hour per node.

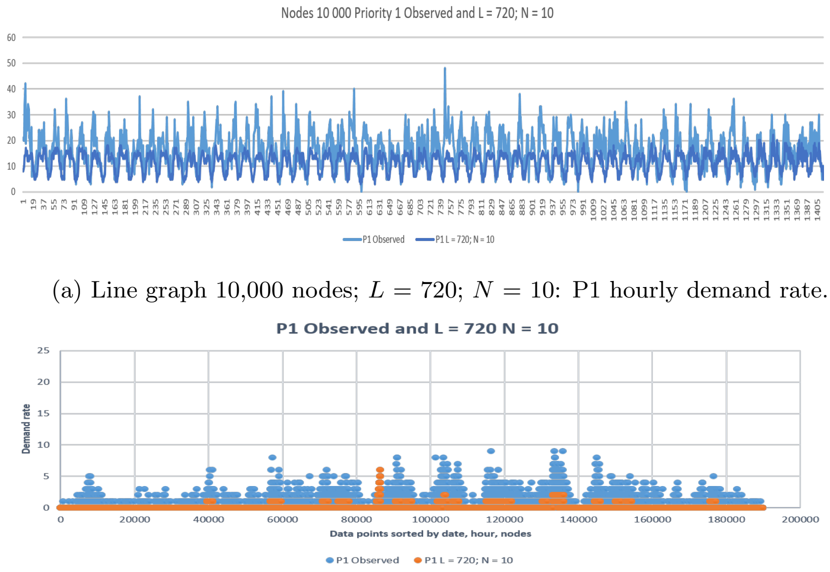
Figure E.1: Graphs for forecasting P1 with 10,000 nodes and $L = 720$; $N = 5$.



(a) Line graph 10,000 nodes; $L = 720$; $N = 5$: P2 hourly demand rate.

(b) Scatter graph 10,000 nodes; $L = 720$; $N = 5$: P2 demand rate per hour per node.

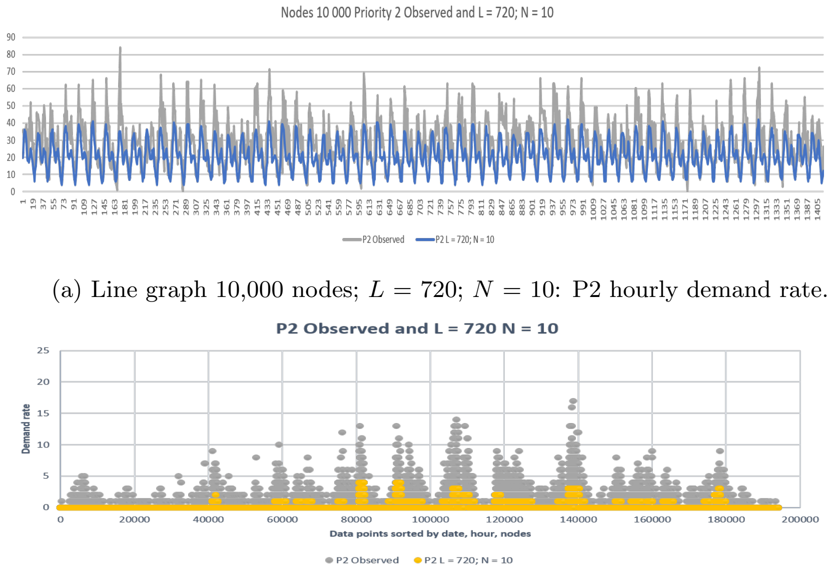
Figure E.2: Graphs for forecasting P2 with 10,000 nodes and $L = 720$; $N = 5$.



(a) Line graph 10,000 nodes; $L = 720$; $N = 10$: P1 hourly demand rate.

(b) Scatter graph 10,000 nodes; $L = 720$; $N = 10$: P1 demand rate per hour per node.

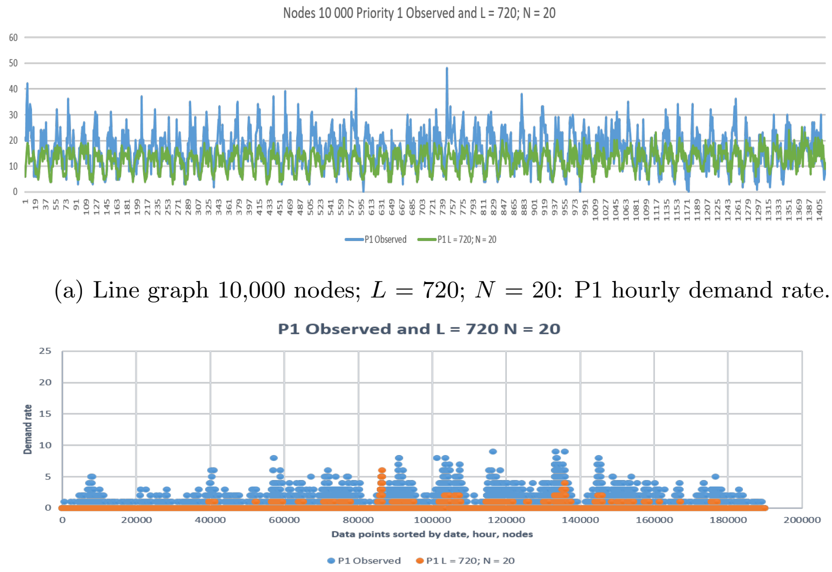
Figure E.3: Graphs for forecasting P1 with 10,000 nodes and $L = 720$; $N = 10$.



(a) Line graph 10,000 nodes; $L = 720$; $N = 10$: P2 hourly demand rate.

(b) Scatter graph 10,000 nodes; $L = 720$; $N = 10$: P2 demand rate per hour per node.

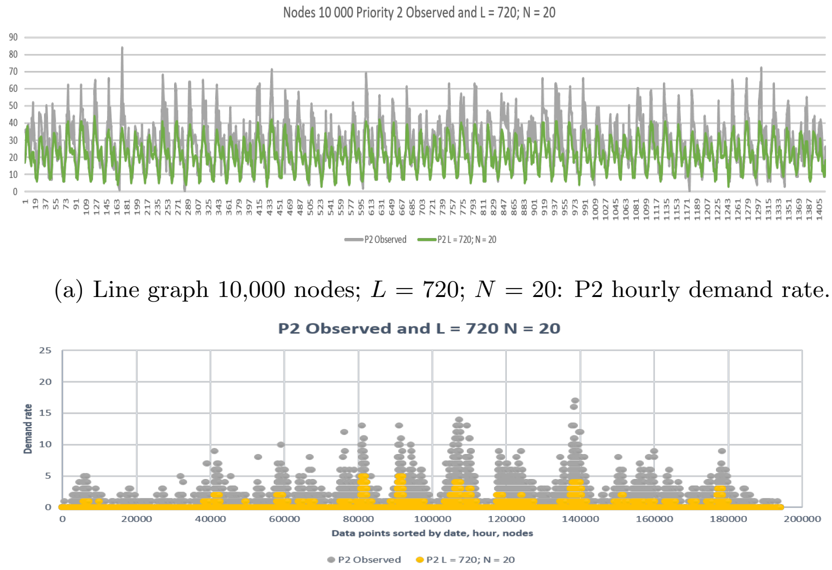
Figure E.4: Graphs for forecasting P2 with 10,000 nodes and $L = 720$; $N = 10$.



(a) Line graph 10,000 nodes; L = 720; N = 20: P1 hourly demand rate.

(b) Scatter graph 10,000 nodes; L = 720; N = 20: P1 demand rate per hour per node.

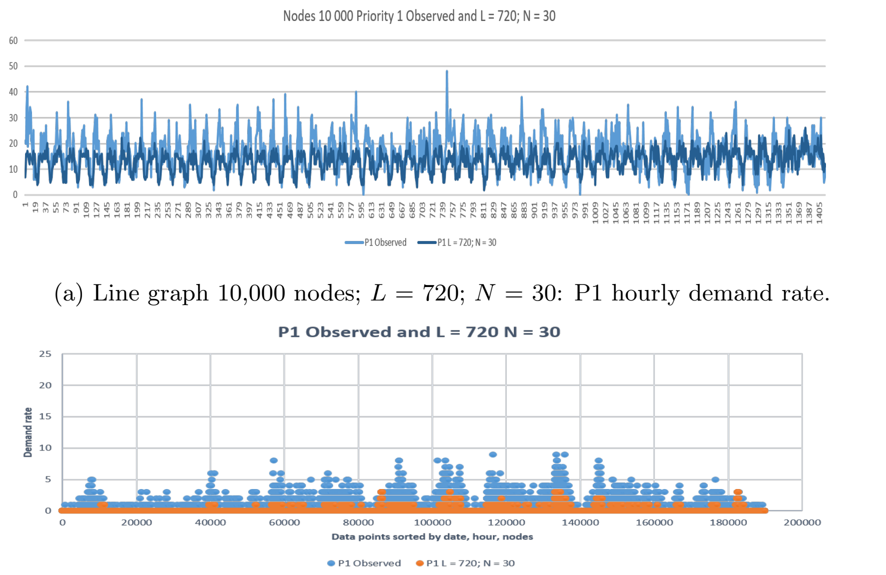
Figure E.5: Graphs for forecasting P1 with 10,000 nodes and L = 720; N = 20.



(a) Line graph 10,000 nodes; L = 720; N = 20: P2 hourly demand rate.

(b) Scatter graph 10,000 nodes; L = 720; N = 20: P2 demand rate per hour per node.

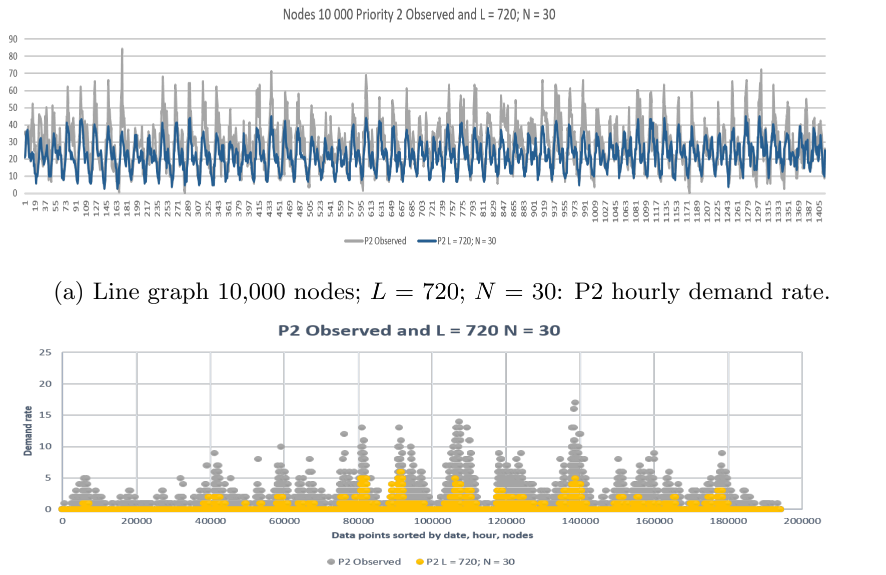
Figure E.6: Graphs for forecasting P2 with 10,000 nodes and L = 720; N = 20.



(a) Line graph 10,000 nodes; $L = 720$; $N = 30$: P1 hourly demand rate.

(b) Scatter graph 10,000 nodes; $L = 720$; $N = 30$: P1 demand rate per hour per node.

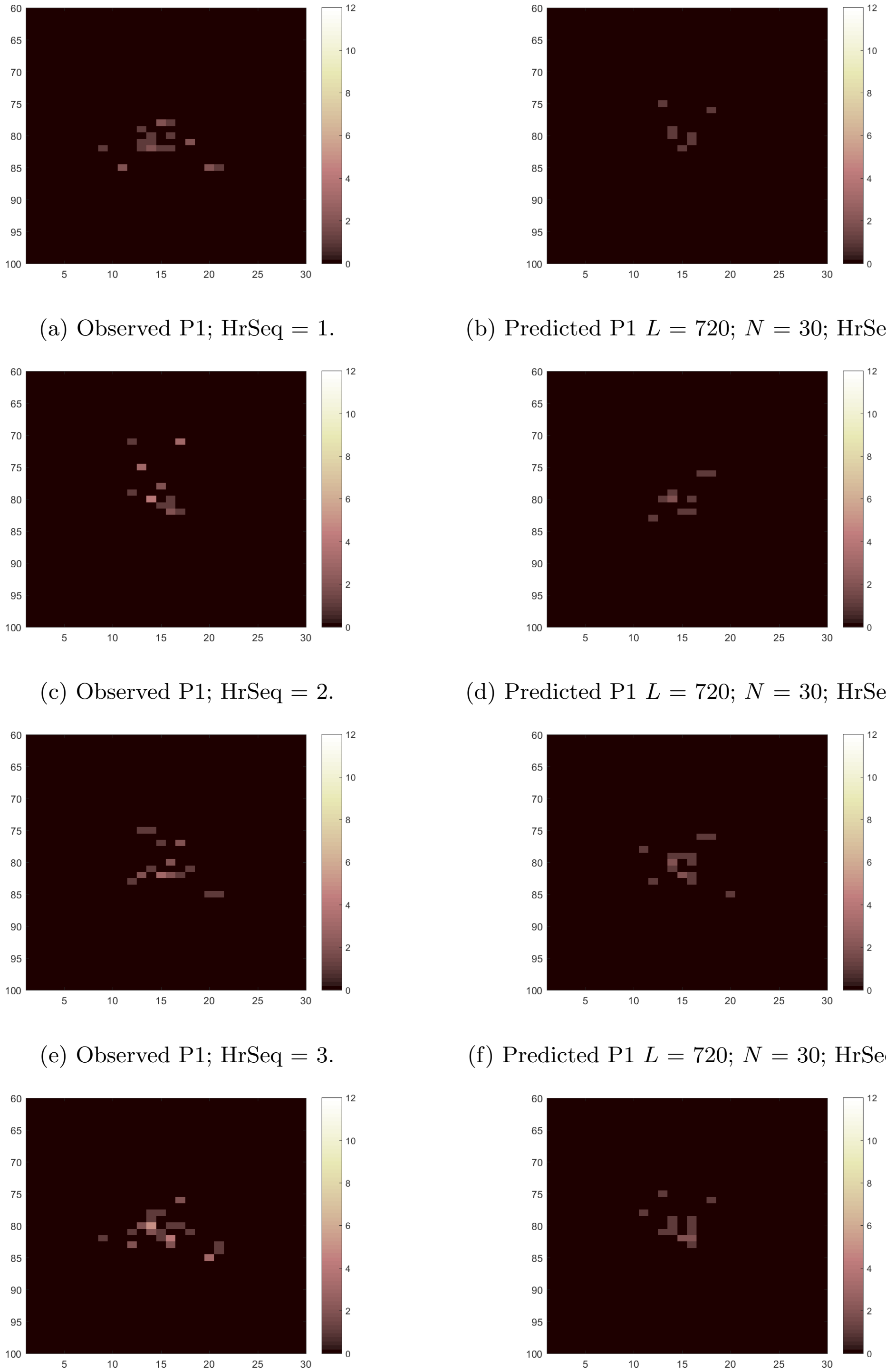
Figure E.7: Graphs for forecasting P1 with 10,000 nodes and $L = 720$; $N = 30$.



(a) Line graph 10,000 nodes; $L = 720$; $N = 30$: P2 hourly demand rate.

(b) Scatter graph 10,000 nodes; $L = 720$; $N = 30$: P2 demand rate per hour per node.

Figure E.8: Graphs for forecasting P2 with 10,000 nodes and $L = 720$; $N = 30$.



(a) Observed P1; HrSeq = 1.

(b) Predicted P1 L = 720; N = 30; HrSeq = 1.

(c) Observed P1; HrSeq = 2.

(d) Predicted P1 L = 720; N = 30; HrSeq = 2.

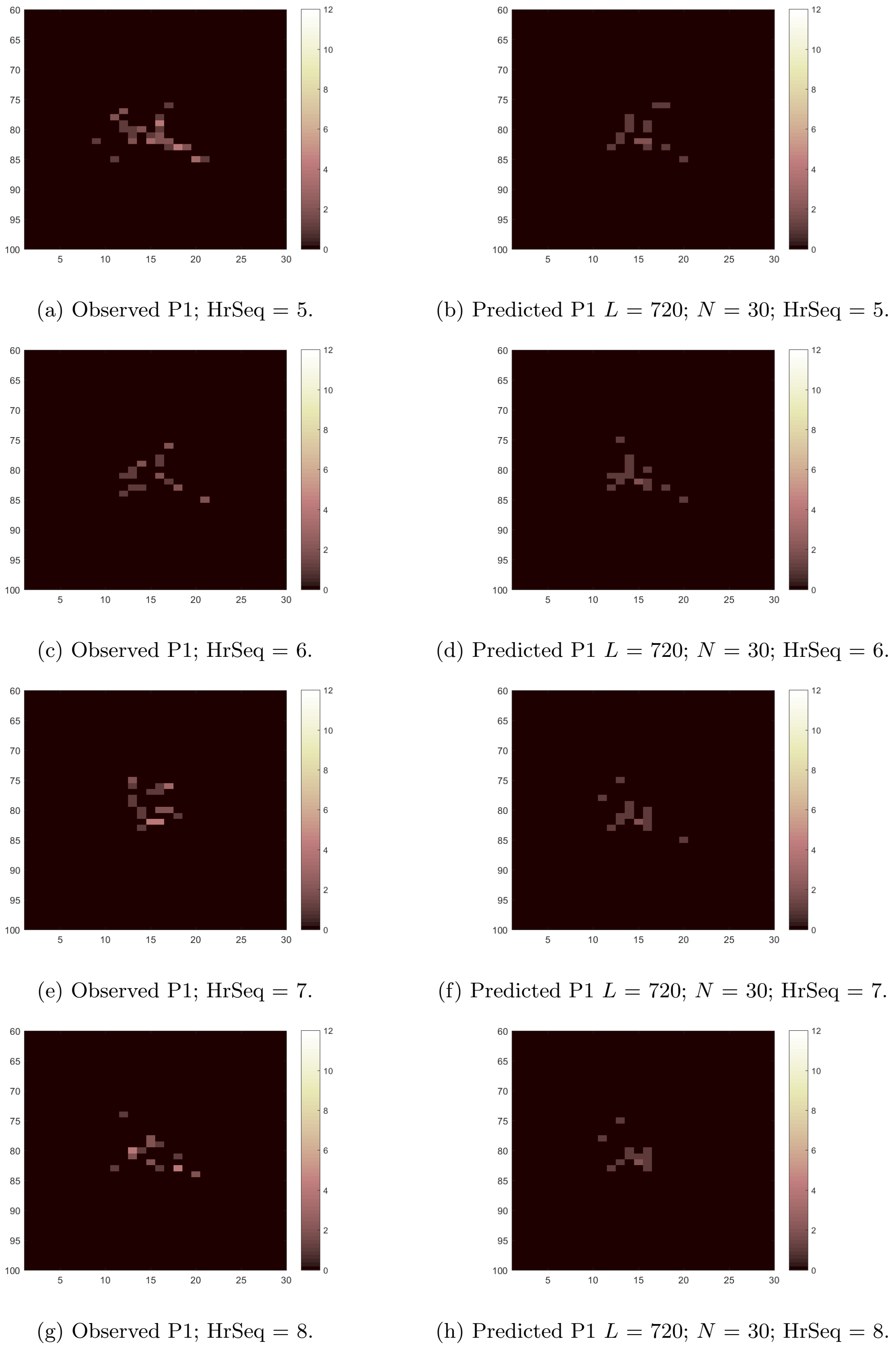
(e) Observed P1; HrSeq = 3.

(f) Predicted P1 L = 720; N = 30; HrSeq = 3.

(g) Observed P1; HrSeq = 4.

(h) Predicted P1 L = 720; N = 30; HrSeq = 4.

Figure E.9: Heat maps HrSeq 1 - 4: Nodes 10,000 Observed P1 VS Predicted P1 L = 720; N = 30.



(a) Observed P1; HrSeq = 5.

(b) Predicted P1 L = 720; N = 30; HrSeq = 5.

(c) Observed P1; HrSeq = 6.

(d) Predicted P1 L = 720; N = 30; HrSeq = 6.

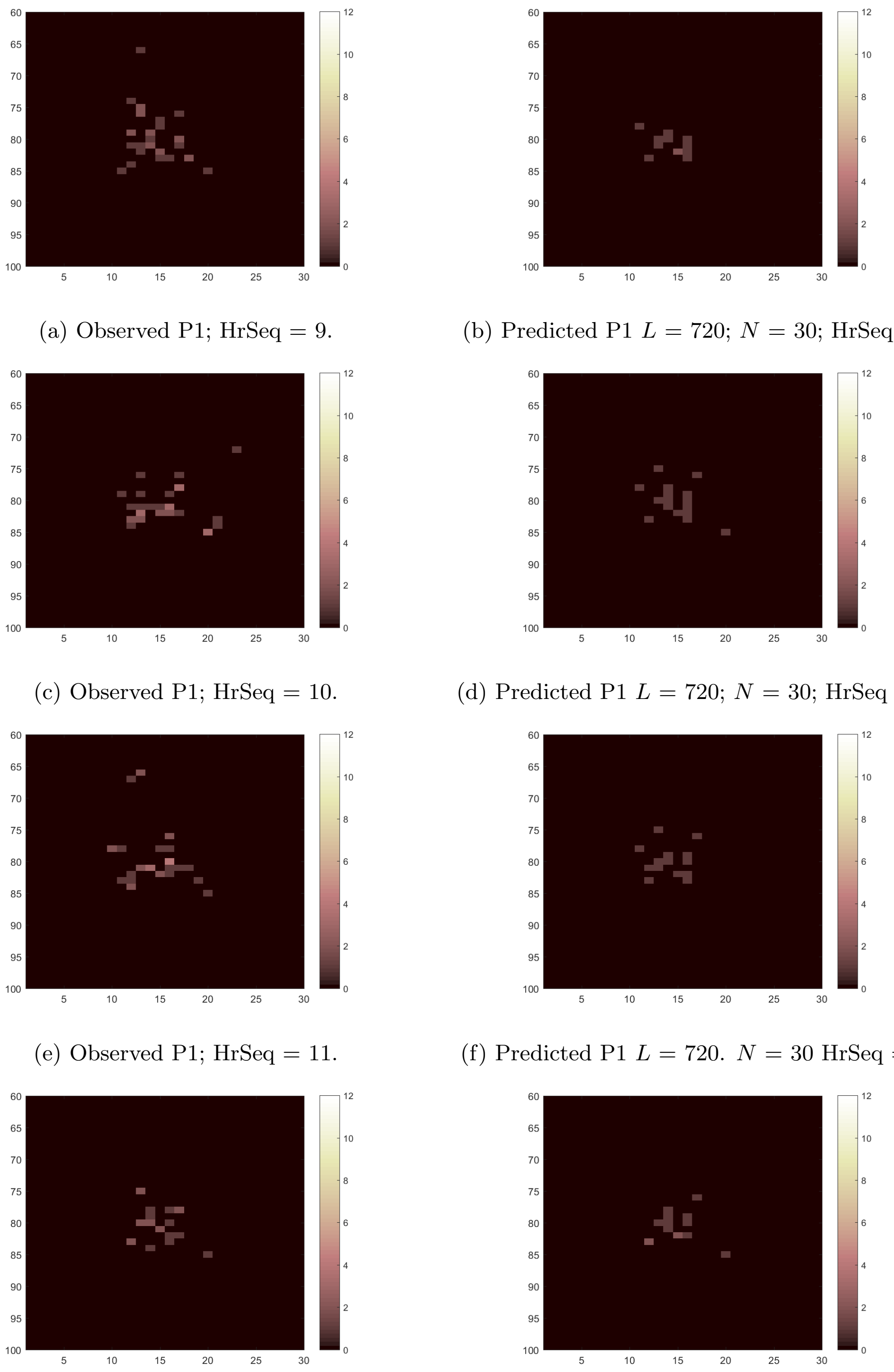
(e) Observed P1; HrSeq = 7.

(f) Predicted P1 L = 720; N = 30; HrSeq = 7.

(g) Observed P1; HrSeq = 8.

(h) Predicted P1 L = 720; N = 30; HrSeq = 8.

Figure E.10: Heat maps HrSeq 5 - 8: Nodes 10,000 Observed P1 VS Predicted P1 L = 720; N = 30.



(a) Observed P1; HrSeq = 9.

(b) Predicted P1 L = 720; N = 30; HrSeq = 9.

(c) Observed P1; HrSeq = 10.

(d) Predicted P1 L = 720; N = 30; HrSeq = 10.

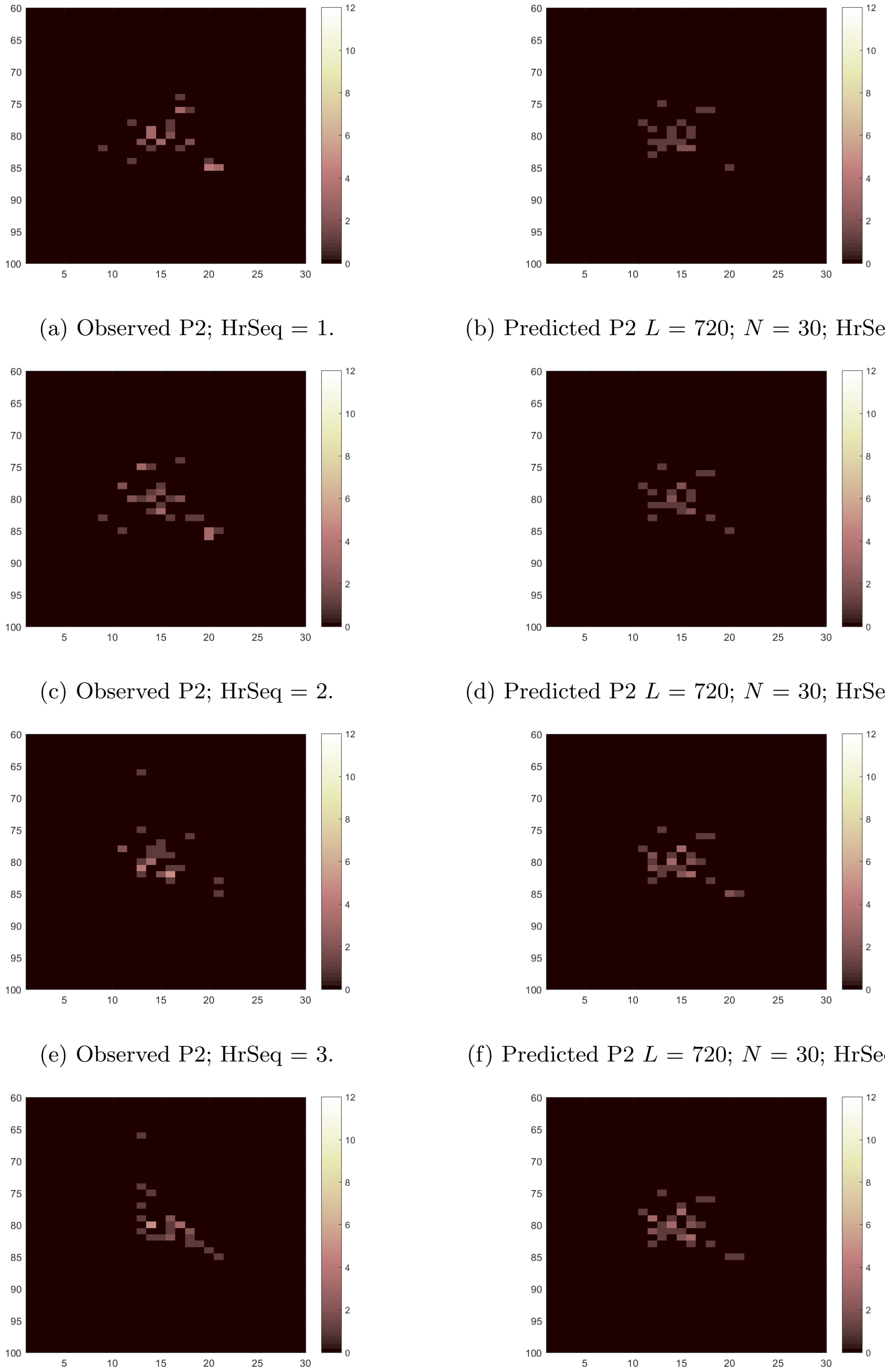
(e) Observed P1; HrSeq = 11.

(f) Predicted P1 L = 720. N = 30 HrSeq = 11.

(g) Observed P1; HrSeq = 12.

(h) Predicted L = 720; N = 30; HrSeq = 12.

Figure E.11: Heat maps HrSeq 9 - 12: Nodes 10,000 Observed P1 VS Predicted P1 L = 720; N = 30.



(a) Observed P2; HrSeq = 1.

(b) Predicted P2 L = 720; N = 30; HrSeq = 1.

(c) Observed P2; HrSeq = 2.

(d) Predicted P2 L = 720; N = 30; HrSeq = 2.

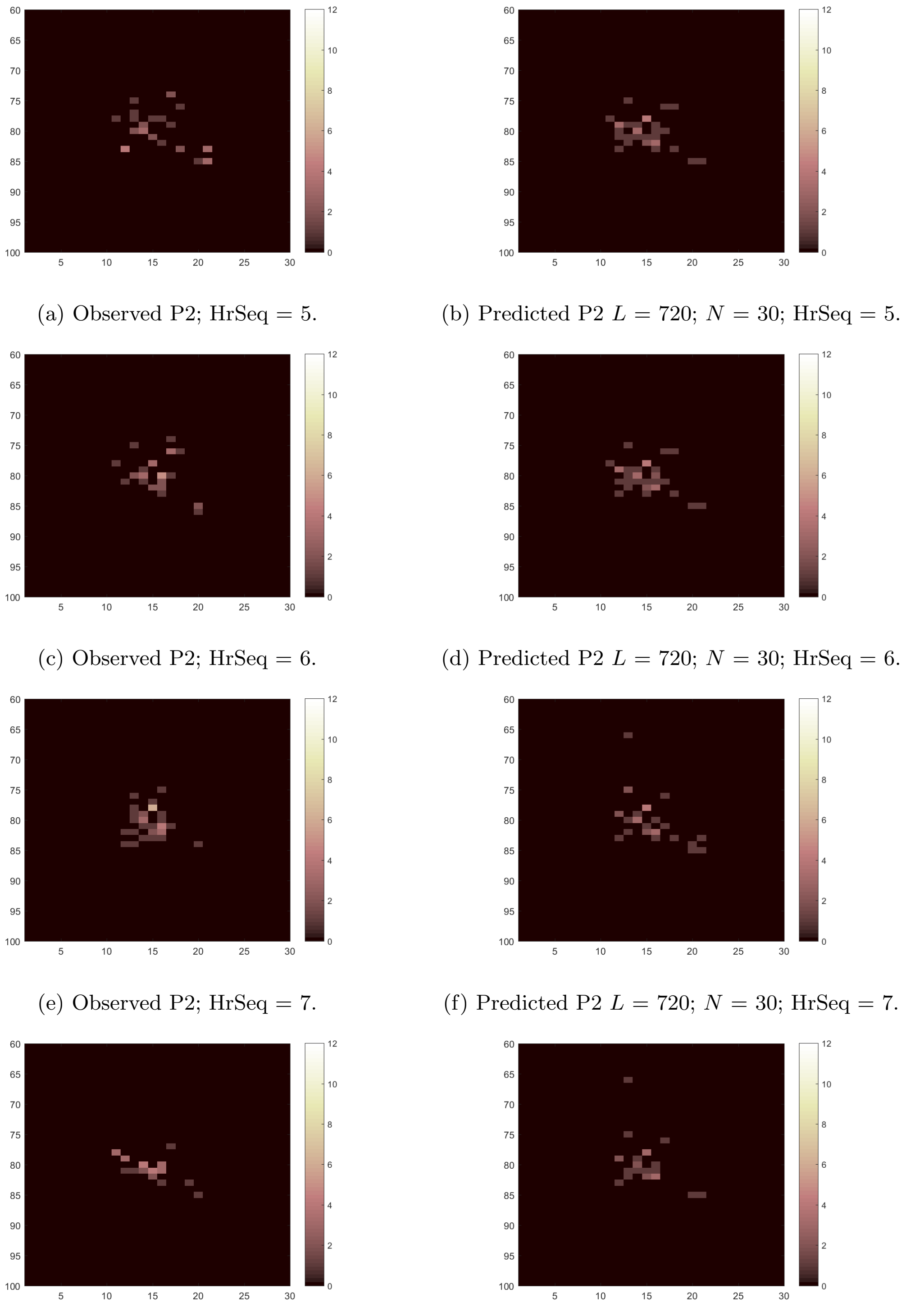
(e) Observed P2; HrSeq = 3.

(f) Predicted P2 L = 720; N = 30; HrSeq = 3.

(g) Observed P2; HrSeq = 4.

(h) Predicted P2 L = 720; N = 30; HrSeq = 4.

Figure E.12: Heat maps HrSeq 1 - 4: Nodes 10,000 Observed P2 VS Predicted P2 L = 720; N = 30.



(a) Observed P2; HrSeq = 5.

(b) Predicted P2 L = 720; N = 30; HrSeq = 5.

(c) Observed P2; HrSeq = 6.

(d) Predicted P2 L = 720; N = 30; HrSeq = 6.

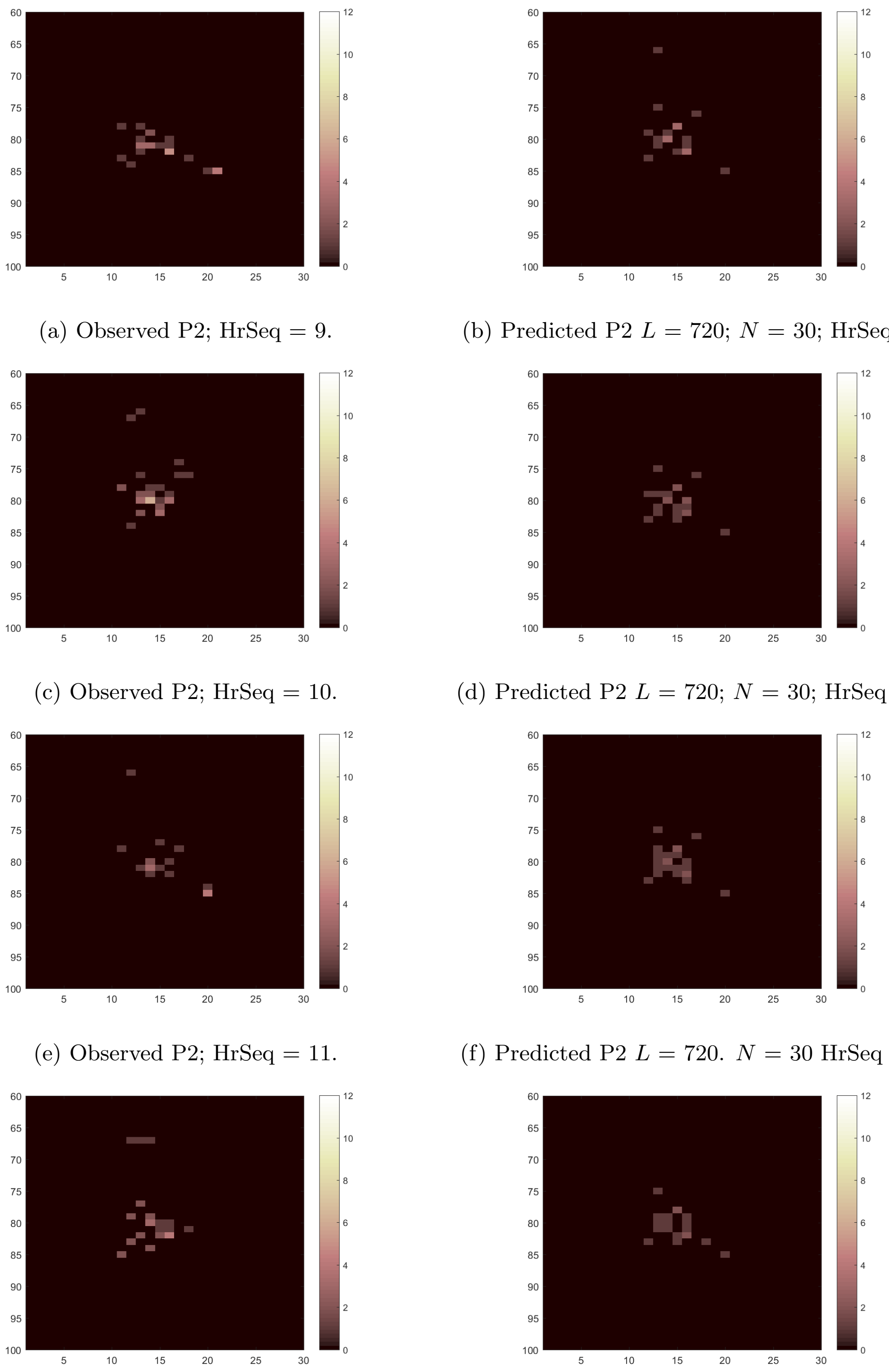
(e) Observed P2; HrSeq = 7.

(f) Predicted P2 L = 720; N = 30; HrSeq = 7.

(g) Observed P2; HrSeq = 8.

(h) Predicted P2 L = 720; N = 30; HrSeq = 8.

Figure E.13: Heat maps HrSeq 5 - 8: Nodes 10,000 Observed P2 VS Predicted P2 L = 720; N = 30.



(a) Observed P2; HrSeq = 9.

(b) Predicted P2 L = 720; N = 30; HrSeq = 9.

(c) Observed P2; HrSeq = 10.

(d) Predicted P2 L = 720; N = 30; HrSeq = 10.

(e) Observed P2; HrSeq = 11.

(f) Predicted P2 L = 720. N = 30 HrSeq = 11.

(g) Observed P2; HrSeq = 12.

(h) Predicted L = 720; N = 30; HrSeq = 12.

Figure E.14: Heat maps HrSeq 9 - 12: Nodes 10,000 Observed P2 VS Predicted P2 L = 720; N = 30.

Appendix F

Matlab code

This appendix contains the Matlab code for four of the '*.m'-files that form part of the DST. These '*.m'-files are added to provided assistance to any who might wish to do further work on this project. The first code is that which can be found in *Thesis.m*.

```
clc;
clear;
close all;

%% Add subfolders to directory path to be able to access them
%

currentFolderContents = dir(pwd);       % Returns all files and folders
                                        % in the current folder
currentFolderContents (¬[currentFolderContents.isdir]) = [];
                                        % Only keep the folders

% Start with 3 to avoid '.' and '..'
for i = 3:length(currentFolderContents)
   addpath(['./' currentFolderContents(i).name])
end

%% Import data from Excel
% Check that path to file and that Excel file names is correct

path = 'C:\...\MatlabThesisFiles\';

matpath = [path 'MatFiles\'];

```

```
if exist('impdata.mat','file')≠ 2
    tic                                  % Start timer

    filenameRaw = 'ExcelFolder\RawData.xlsx';
    filenameMat = 'impdata.mat';
    % Call import function
    ImpData([path filenameRaw],[matpath filenameMat]);

    importT = toc;                       % Stop timer and save importT
    save([matpath 'checkImport.mat'],'importT')
end

%% Save scenario information

% Scenario information
% Save information on the scenario in writable scInfo.mat

sc = matfile('scInfo.mat','Writable',true);

% Scenario = 1 ; all demand P1
% Scenario = 2 ; demand P1 or P2
Scenario = 1;                    % Scenario
% Needed for ProcessData
num = 102;                       % Num to create (num-2)^2 nodes

sc.path = path;
sc.matpath = matpath;
sc.num = num;
sc.Scenario = Scenario;
sc.StartWkDate = 20160101;

sc.alloRelax = 1;                % if 0 = original constr; 1 = relaxed
sc.selectRelax = 1;              % if 0 = original constr; 1 = relaxed

%% Process imported data
% Process data and create demand arrays required for forecasting
processY = 0;

if processY == 1

    fileMatProData = 'processdata.mat';
    % Call process imported data function
    tic
    ProcessData(num,[sc.matpath fileMatProData]);
    proDataT = toc;

    fileMatProDem = 'processdemand.mat';
    % Call process demand function
    tic
    ProcessDemand([sc.matpath fileMatProDem]);
    proDemT = toc;

    processT = proDataT+proDemT;

    save([sc.matpath 'checkProcess.mat'],'proDataT','proDemT','processT')
end

%% Forecast
% First set SSA parameter values

forecastY = 0;

if forecastY == 1
    % SSA parameters
    L = 720;                       % Window length; L ≤ Length(Y)/2
    N = 20;                        % Number of reconstructed components

    % Load processed demand
    filename = 'processdemand.mat';
    pdem = matfile(filename);

    SSA_VarD = pdem.SSA_VarD;
    SSA_P1 = pdem.SSA_P1;
    SSA_P2 = pdem.SSA_P2;

    if sc.Scenario == 1
        tic                             % Start timer
        [ForecastVarD,meanRMSE,maxRMSE] = SSAmain(SSA_VarD,L,N);
        forecastT = toc;                % Stop and save timer value

        [RMSE,HrlyObsD,HrlyFrcD,ObsDFrcD] = calcRMSE(pdem.VarD, ...
            ForecastVarD, sc.StartWkDate);
        %'forecastdemandS1.mat'
        save([sc.matpath 'forecastdemandS1.mat'],'forecastT',...
            'ForecastVarD','meanRMSE','maxRMSE','RMSE','HrlyObsD',...
            'HrlyFrcD','ObsDFrcD')
    else
        tic                             % Start timer
        % Forecast P1 demand
        [ForecastP1,meanRMSEP1,maxRMSEP1] = SSAmain(SSA_P1,L,N);
        forecastTP1 = toc;
        [RMSEP1,HrlyObsDP1,HrlyFrcDP1,ObsDFrcDP1] = calcRMSE(pdem.P1,...
            ForecastP1, sc.StartWkDate);

        % Forecast P2 demand
        tic
        [ForecastP2,meanRMSEP2,maxRMSEP2] = SSAmain(SSA_P2,L,N);
        forecastTP2 = toc;
        [RMSEP2,HrlyObsDP2,HrlyFrcDP2,ObsDFrcDP2] = calcRMSE(pdem.P2,...
            ForecastP2, sc.StartWkDate);

        save([sc.matpath 'forecastdemandS2.mat'],'forecastTP1',...
            'forecastTP2','ForecastP1','meanRMSEP1','maxRMSEP1',...
            'RMSEP1','HrlyObsDP1','ObsDFrcDP1','HrlyFrcDP1',...
            'ForecastP2','meanRMSEP2','maxRMSEP2','RMSEP2','HrlyObsDP2',...
            'HrlyFrcDP2','ObsDFrcDP2')
    end
end

%% Save scenario information

sc.alpha1 = 0.95;                 % Service reliability value for P1
sc.alpha2 = 0.95;                 % Service reliability value for P2
sc.Beta = 0.015;                  % Relocation influence on objv
sc.MCN = 1000;                    % max ABC cycles
sc.ColSiz = 40;                   % colony size
sc.NumAmbDay = 70;
sc.NumAmbNight = 55;
sc.ambSpd = 60;                   % Average km/hr
sc.rP1 = 15;                      % response time target P1
sc.rP2 = 30;                      % response time target P2
sc.hnum = 30;                     % (hnum-2)^2 hnodes
sc.hcapacity = 10;                % Holding site node ambulance capacity
sc.NumSelect = 15;

clearvars -except sc

%% Create holding site node array
% Create hnodes and HoldV. Base the creation of the holding site ...
    nodes on
% the demand for Scenario 1 and use it for both scenarios

holdY = 0;

if holdY == 1
```

```
    tic                                 % Start timer

    % Create Holdingsite variable (mapping toolbox required)
    [HoldV,hnodes] = CreateH(sc.hnum,sc.hcapacity,sc.matpath);
    % holding site 648 is in the ocean
    HoldV(any(HoldV == 648,2),:)=[];

    DummyH(1,1) = size(hnodes,1)+1;     % Nodenum
    DummyH(1,2) = size(hnodes,1)+1;     % Hnum
    DummyH(1,3) = -33.9348108;          % Lat
    DummyH(1,4) = 18.4894037;           % Lon
    DummyH(1,5) = 100;                  % hcap

    % Calculate distance between holding site nodes
    [distHH] = DistanceHH([HoldV;DummyH],sc.ambSpd,sc.hnum);

    save([sc.matpath 'holdingSiteNodes.mat'],'HoldV','distHH','hnodes'...
        ,'DummyH')

    createHT = toc;

    save([sc.matpath 'checkCreateH.mat'],'createHT')
end

%% Holding site location and ambulance allocation and relocation

% Load processdata
pdat = matfile('processdata.mat');
nodes = pdat.nodes;
tcall = pdat.tcall;
yymmdd = pdat.yymmdd;

% Load Var
v = matfile('Var.mat','Writable',true);
v.tcall = tcall;

% Lood holdingSiteNodes
h = matfile('holdingSiteNodes.mat');
v.HoldV = h.HoldV;
v.hnodes = h.hnodes;
v.DummyH = h.DummyH;
v.distHH = h.distHH;

% Load scInfo
sc = matfile('scInfo.mat');%,'Writable',true);
```

```

% Planning horizon
[rStr] = find(yymmdd == sc.StartWkDate);
wk = yymmdd(rStr:(rStr+6),1);

tic
abcT = zeros(size(wk,1),1);

for d = 1:2
    v.ShiftDate = wk(d);
    if sc.Scenario == 1
        load forecastdemandS1
        % Create neighbouthoods
        [v.W,v.V,v.N] = CreateNeigh(ForecastVarD,v.ShiftDate,v.HoldV,...
            sc.rP1,nodes,sc.ambSpd);
        v.VarD = ForecastVarD(logical(ForecastVarD(:,3) == ...
            v.ShiftDate),:);
    else
        load forecastdemandS2
        ForecastVarD = [ForecastP1;ForecastP2];

        [WP1,VP1,NP1] = CreateNeigh(ForecastP1,v.ShiftDate,v.HoldV,...
            sc.rP1,nodes,sc.ambSpd);
        [WP2,VP2,NP2] = CreateNeigh(ForecastP2,v.ShiftDate,v.HoldV,...
            sc.rP2,nodes,sc.ambSpd);

        P1 = ForecastP1(logical(ForecastP1(:,3) == v.ShiftDate),:);
        sizP1 = size(P1);
        P2 = ForecastP2(logical(ForecastP2(:,3) == v.ShiftDate),:);
        sizP2 = size(P2);

        v.VarD = [[P1 ones(sizP1(1),1) zeros(sizP1(1),1)];...
            [P2 zeros(sizP2(1),1) ones(sizP2(1),1)]];
        v.W = [[WP1 ones(size(WP1,1),1) zeros(size(WP1,1),1)];...
            [WP2 zeros(size(WP2,1),1) ones(size(WP2,1),1)]];
        v.V = [[VP1 ones(size(VP1,1),1) zeros(size(VP1,1),1)];...
            [VP2 zeros(size(VP2,1),1) ones(size(VP2,1),1)]];
        v.N = [[NP1 ones(size(NP1,1),1) zeros(size(NP1,1),1)];...
            [NP2 zeros(size(NP2,1),1) ones(size(NP2,1),1)]];
    end

    % solve for v.ShiftDate
    tic

    ABCcon();

    abcT(d) = toc;
end

horizonT = toc;

save([sc.matpath 'WeekTime.mat'],'abcT','horizonT')
```

The following Matlab code shows the code found in *ABCcon.m*.

```
%
% Copyright (c) 2015, Yarpiz (www.yarpiz.com)
% All rights reserved. Please read the "license.txt" for license terms.
%
% Project Code: YPEA114
% Project Title: Implementation of Artificial Bee Colony in MATLAB
% Publisher: Yarpiz (www.yarpiz.com)
%
% Developer: S. Mostapha Kalami Heris (Member of Yarpiz Team)
%
% Contact Info: sm.kalami@gmail.com, info@yarpiz.com
%

% Adapted for constrained optimisation. For guidelines read
% "Improved Artificial Bee Colony Algorithm for Constrained Problems" by
% Brajevic et al. and "A modi?ed Arti?cial Bee Colony (ABC) algorithm for
% constrained optimization problems" by Karaboga et al.

function[]=ABCcon()
    % Load Var
    v = matfile('Var.mat');
    HoldV = v.HoldV;
    ShiftDate = v.ShiftDate;

    % Load scInfo
    sc = matfile('scInfo.mat');
    Beta = sc.Beta;

    % Create Structure
    ANS.Position = [];
    ANS.NumAmb = [];
    ANS.pA = [];
    ANS.Constraints = [];
    ANS.Violation = [];
    ANS.HrObjV = [];
    ANS.ShiftObjV = [];
    ANS.Relocation = [];
    ANS.HrReloCost = [];
    ANS.ShiftReloCost = [];
    ANS.TotObjVCost = [];
    ANS.YZ = [];            % Indicates which demand nodes are covered
                            % by which selected holding site nodes

    % Solution structure
    Solution = repmat(ANS,[2,1]);

    sizH = size(HoldV);    % Size of possible holding site nodes array

    % Loop through both shifts of the ShiftStartDate
    for sh = 1:2
        rng('shuffle')
        %% HOLDING SITE NODE LOCATION AND AMBULANCE ALLOCATION
        nVar = sizH(1);                 % Num Decision Var (Dimensions)

        VarSize = [1 nVar];             % Decision Variables Matrix Size

        % Binary
        VarMinH = 0;                     % Decision Variables Lower Bound
        VarMaxH = 1;                     % Decision Variables Upper Bound

        % Integer, may not exceed holding site node ambulance capacity
        VarMinA = 0;                    % Decision Var Lower Bound
        VarMaxA = max(HoldV(:,5));      % Decision Var Upper Bound

        %% ABC SETTINGS

        MCN = sc.MCN;                   % Maximum Cycle Number

        ColSiz = sc.ColSiz;             % Colony Size
        nPop = ColSiz/2;                % Population Size (Colony Size)

        nOnlooker = ColSiz/2;           % Number of Onlooker Bees

        Limit = round(0.6*ColSiz*nVar); % Abandonment Limit Parameter
                                        % [0.5*ColSiz*nVar, ColSiz*nVar]

        MR = 0.7;                       % Modification rate ...
            (determines if
                                        % variable will be changed or ...
```

```
                                    not)
                                % between [0.3, 0.8]

SPP = round(0.1*ColSiz*nVar);   % Scout Production Period ...
    (was 0.5)
                                % [0.1*ColSiz*nVar, ...
                                    2*ColSiz*nVar];

a = 1;                          % Acceleration Coeff Upper Bound

%% Initialization
% Initialize Population Array
pop = repmat(ANS,[nPop,1]);
newbee = repmat(ANS,[1,1]);

% Initialize Best Solution Ever Found
BestSol.TotObjVCost = inf;
initialise = 1;

% Create Initial Population
for i = 1:nPop
    % Create possible holding site node placement
    pop(i).Position = round(unifrnd(VarMinH,VarMaxH,VarSize));

    [AmbANS] = ...
        AmbAllo(pop(i),VarMinA,VarMaxA,VarSize,HoldV,sh,...
        initialise);

    pop(i).YZ = AmbANS.YZ;                        % Hnodes ...
        coverage
    pop(i).NumAmb = AmbANS.NumAmb;                % Amb alloc for
                                                  % sh and ...
                                                      Position
    pop(i).pA = AmbANS.pA;                        % Num avail amb
    pop(i).Constraints = AmbANS.Constraints;      % Constraints
    pop(i).Violation = AmbANS.Violation;          % Num constr
                                                  % violated
    pop(i).HrObjV = AmbANS.HrObjV;                % Hrly obj ...
        value
    pop(i).ShiftObjV = AmbANS.ShiftObjV;          % Shift obj ...
        value

    % Determine relocations and cost
    [R] = Relo(pop(i).NumAmb,pop(i).pA,Beta);
    pop(i).Relocation = R.r;                      % ...
```

```
        Relocations req
    pop(i).HrReloCost = R.ReloCost;               % Hrly relo ...
        cost
    pop(i).ShiftReloCost = sum(pop(i).HrReloCost);% Shift ...
        relo cost

    % ShiftObjV + ShiftReloCost for all hrs
    pop(i).TotObjVCost = pop(i).ShiftObjV + pop(i).ShiftReloCost;

    if pop(i).TotObjVCost ≤ BestSol.TotObjVCost
        BestSol = pop(i);
    end
end

% Abandonment Counter
failure = zeros(nPop,1);

% Array to Hold Best Cost Values
BestTotObjVCost = zeros(MCN,4);
%TotObjVCost,Violation,ObjV,ReloCost

%% ABC Main Loop
for cycle = 1:MCN
    %% EMPLOYED BEES
    % Search for new food sources having more nectar within the
    % neighbourhood of the current food source. Evaluate a
    % neighbouring food source.
    initialise = 0;
    for i = 1:nPop
        % Choose k randomly, not equal to i
        K = [1:i-1 i+1:nPop];
        k = K(randi(numel(K)));

        % Define Acceleration Coeff. (random num between [-a, a])
        phi = a*unifrnd(-1,+1,VarSize);

        newbee.Position = pop(i).Position;
        newbee.NumAmb = pop(i).NumAmb;

        changed = 0;
        for d = 1:nVar
            % Uniformly distr random real number in [0,1]
            Rd = rand;

            if Rd < MR
```

```
                        % New Bee Position
                        newbee.Position(d) = round(pop(i).Position(d) ...
                            + ...
                            phi(d).*(pop(i).Position(d) - ...
                            pop(k).Position(d)));

                        if newbee.Position(d)<0
                            newbee.Position(d) = 0;
                        else
                            if newbee.Position(d) > 1
                                newbee.Position(d) = 1;
                            end
                        end

                        for tp = 1:12
                           newbee.NumAmb(tp,d) = ...
                               round(pop(i).NumAmb(tp,d) ...
                               + phi(d).*(pop(i).NumAmb(tp,d) - ...
                               pop(k).NumAmb(tp,d)));

                           if newbee.NumAmb(tp,d) < 0
                               newbee.NumAmb(tp,d) = 0;
                           end

                        end
                        changed = changed + 1;
                    end
                end

                % if no dimension value was changed change at least one
                if changed == 0
                    d = round(1 + (nVar-1)*rand);
                    newbee.Position(d) = round(pop(i).Position(d) + ...
                        phi(d).*(pop(i).Position(d) - ...
                            pop(k).Position(d)));
                    for tp = 1:12
                       newbee.NumAmb(tp,d) = ...
                           round(pop(i).NumAmb(tp,d) ...
                           + phi(d).*(pop(i).NumAmb(tp,d) - ...
                           pop(k).NumAmb(tp,d)));
                    end
                end

                % Allocate ambulances and evaluate obj value for the ...
                    shift
```

```
[AmbANS] = ...
    AmbAllo(newbee,VarMinA,VarMaxA,VarSize,HoldV,...
    sh,initialise);

newbee.YZ = AmbANS.YZ;                    % Hnodes ...
    coverage
newbee.NumAmb = AmbANS.NumAmb;            % Amb alloc for
                                          % sh and ...
                                              Position
newbee.pA = AmbANS.pA;                    % Num avail amb
newbee.Constraints = AmbANS.Constraints;  % Constraints
newbee.Violation = AmbANS.Violation;      % Num constr
                                          % violated
newbee.HrObjV = AmbANS.HrObjV;            % Hrly obj ...
    value
newbee.ShiftObjV = AmbANS.ShiftObjV;      % Shift obj ...
    value

% Determine relocations and cost
[R] = Relo(newbee.NumAmb,newbee.pA,Beta);
newbee.Relocation = R.r;                  % ...
    Relocations req
newbee.HrReloCost = R.ReloCost;           % Hrly relo ...
    cost
newbee.ShiftReloCost = sum(newbee.HrReloCost); % ...
    Shift relo
                                          % cost

% ShiftObjV + ShiftReloCost for all hrs
newbee.TotObjVCost = newbee.ShiftObjV + ...
    newbee.ShiftReloCost;

% Comparison and tournament selection
if newbee.Violation == 0 && pop(i).Violation == 0
    % if both are feasible choose the better TotObjV
    if newbee.TotObjVCost ≤ pop(i).TotObjVCost
        pop(i) = newbee;
        failure(i) = 0;
    else
        failure(i) = failure(i)+1;
    end
else
    if newbee.Violation > 0 && pop(i).Violation > 0
        % if both are infeasible choose smaller
        % infeasibility
```

```
                    if newbee.Violation < pop(i).Violation
                       pop(i) = newbee;
                       failure(i) = 0;
                    else
                        if newbee.Violation == pop(i).Violation

                            if newbee.TotObjVCost ≤ ...
                                pop(i).TotObjVCost
                                pop(i) = newbee;
                                failure(i) = 0;
                            else
                                failure(i) = failure(i)+1;
                            end
                        end
                    end
                else
                    if newbee.Violation == 0 && pop(i).Violation ...
                        > 0
                        pop(i) = newbee;
                        failure(i) = 0;
                    else
                        failure(i) = failure(i)+1;
                    end
                end
            end
        end

        %% FITNESS AND SELECTION PROBABILITIES

        fit = zeros(nPop,1);
        vio = zeros(nPop,1);
        sumfit = 0;
        sumvio = 0;
        for i = 1:nPop
           fit(i,1) = Fitness(pop(i).TotObjVCost);
           vio(i,1) = pop(i).Violation;
           if pop(i).Violation == 0
               sumfit = sumfit + fit(i,1);
           else
               sumvio = sumvio + vio(i,1);
           end
        end
        pm = zeros(nPop,1);
        for i = 1:nPop
            if pop(i).Violation == 0
```

```
                pm(i) = 0.5 + (fit(i,1)/sumfit)*0.5;
            else
                pm(i) = (1-(vio(i,1)/sumvio))*0.5;
            end
        end

        %% ONLOOKER BEES
        initialise = 0;
        for m = 1:nOnlooker
            % Select Source Site
            i = RouletteWheelSelection(pm);

            % Choose k randomly, not equal to i
            K = [1:i-1 i+1:nPop];
            k = K(randi(numel(K)));

            % Define Acceleration Coeff. (random num between [-a, a])
            phi = a*unifrnd(-1,+1,VarSize);

            newbee.Position = pop(i).Position;
            newbee.NumAmb = pop(i).NumAmb;

            changed = 0;
            for d = 1:nVar
                % Uniformly distr random real number in [0,1]
                Rd = rand;

                if Rd < MR
                % New Bee Position
                    newbee.Position(d) = round(pop(i).Position(d) ...
                        +...
                        phi(d).*(pop(i).Position(d) - ...
                        pop(k).Position(d)));

                    if newbee.Position(d)<0
                        newbee.Position(d) = 0;
                    else
                        if newbee.Position(d) > 1
                            newbee.Position(d) = 1;
                        end
                    end

                    for tp = 1:12
                       newbee.NumAmb(tp,d) = ...
                           round(pop(i).NumAmb(tp,d) + ...
                           phi(d).*(pop(i).NumAmb(tp,d) - ...
                           pop(k).NumAmb(tp,d)));

                       if newbee.NumAmb(tp,d) < 0
                           newbee.NumAmb(tp,d) = 0;
                       end
                    end
                    changed = changed + 1;
                end
            end

            % if no dimension value was changed change at least one
            if changed == 0
                d = round(1 + (nVar-1)*rand);
                newbee.Position(d) = round(pop(i).Position(d) + ...
                    phi(d).*(pop(i).Position(d) - ...
                        pop(k).Position(d)));
                for tp = 1:12
                   newbee.NumAmb(tp,d) = ...
                       round(pop(i).NumAmb(tp,d) + ...
                       phi(d).*(pop(i).NumAmb(tp,d) - ...
                       pop(k).NumAmb(tp,d)));
                end
            end

            % Allocate ambulances and evaluate obj value for the ...
                shift
            [AmbANS] = ...
                AmbAllo(newbee,VarMinA,VarMaxA,VarSize,HoldV,...
                sh,initialise);

            newbee.YZ = AmbANS.YZ;                          % Hnodes ...
                coverage
            newbee.NumAmb = AmbANS.NumAmb;                  % Amb alloc for
                                                            % sh and ...
                                                                Position
            newbee.pA = AmbANS.pA;                          % Num avail amb
            newbee.Constraints = AmbANS.Constraints;        % Constraints
            newbee.Violation = AmbANS.Violation;            % Num constr
                                                            % violated
            newbee.HrObjV = AmbANS.HrObjV;                  % Hrly obj ...
                value
            newbee.ShiftObjV = AmbANS.ShiftObjV;            % Shift obj ...
                value

```

```
% Determine relocations and cost
[R] = Relo(newbee.NumAmb,newbee.pA,Beta);
newbee.Relocation = R.r;                  % ...
    Relocations req
newbee.HrReloCost = R.ReloCost;           % Hrly relo ...
    cost
newbee.ShiftReloCost = sum(newbee.HrReloCost); % ...
    Shift relo
                                          % cost

% ShiftObjV + ShiftReloCost for all hrs
newbee.TotObjVCost = newbee.ShiftObjV + ...
    newbee.ShiftReloCost;

% Comparison and tournament selection
if newbee.Violation == 0 && pop(i).Violation == 0
    % if both are feasible choose the better TotObjV
    if newbee.TotObjVCost ≤ pop(i).TotObjVCost
        pop(i) = newbee;
        failure(i) = 0;
    else
        failure(i) = failure(i)+1;
    end
else
    if newbee.Violation > 0 && pop(i).Violation > 0
        % if both are infeasible choose smaller
        % infeasibility
        if newbee.Violation < pop(i).Violation
           pop(i) = newbee;
           failure(i) = 0;
        else
            if newbee.Violation == pop(i).Violation

                if newbee.TotObjVCost ≤ ...
                    pop(i).TotObjVCost
                    pop(i) = newbee;
                    failure(i) = 0;
                else
                    failure(i) = failure(i)+1;
                end
            end
        end
    else
        if newbee.Violation == 0 && pop(i).Violation ...
            > 0
```

```
                pop(i) = newbee;
                failure(i) = 0;
            else
                failure(i) = failure(i)+1;
            end
        end
    end
end

%% SCOUT BEES
out =¬rem(cycle,SPP)*(cycle/SPP);
if out ≠ 0
    initialise = 1;
    for i = 1:nPop
        if failure(i) ≥ Limit
            pop(i).Position = ...
                round(unifrnd(VarMinH,VarMaxH,...
                VarSize));

            [AmbANS] = AmbAllo(pop(i),VarMinA,VarMaxA,...
                VarSize,HoldV,sh,initialise);%,alpha1,alpha2);

            pop(i).YZ = AmbANS.YZ;               % Hnodes cov
            pop(i).NumAmb = AmbANS.NumAmb;       % Amb ...
                allo for
                                                 % sh and Pos
            pop(i).pA = AmbANS.pA;               % Num avail
                                                 % amb
            pop(i).Constraints = AmbANS.Constraints; % Constr
            pop(i).Violation = AmbANS.Violation; % Num constr
                                                 % violated
            pop(i).HrObjV = AmbANS.HrObjV;       % Hrly ...
                obj val
            pop(i).ShiftObjV = AmbANS.ShiftObjV; % Shift ...
                obj v

            % Determine relocations and cost
            [R] = Relo(pop(i).NumAmb,pop(i).pA,Beta);
            pop(i).Relocation = R.r;              % Relo req
            pop(i).HrReloCost = R.ReloCost;       % Hrly cost
            pop(i).ShiftReloCost = sum(pop(i).HrReloCost);
                                                  % Shift ...
                                                      cost

            % ShiftObjV + ShiftReloCost for all hrs
            pop(i).TotObjVCost = pop(i).ShiftObjV + ...
                pop(i).ShiftReloCost;

            failure(i) = 0;        % Restart abandonment ...
                array
        end
    end
end

%% UPDATE BEST SOLUTION EVER FOUND
% Comparison and tournament selection
if pop(i).Violation == 0 && BestSol.Violation == 0
    % if both are feasible choose the better TotObjV
    if pop(i).TotObjVCost ≤ BestSol.TotObjVCost
        BestSol = pop(i);
    end
else
    if pop(i).Violation > 0 && BestSol.Violation > 0
        % if both are infeasible choose smaller
        % infeasibility
        if pop(i).Violation < BestSol.Violation
           BestSol = pop(i);
        else
            if pop(i).Violation == BestSol.Violation
                if pop(i).TotObjVCost ≤ BestSol.TotObjVCost
                    BestSol = pop(i);
                end
            end
        end
    else
        if pop(i).Violation == 0 && BestSol.Violation > 0
            BestSol = pop(i);
        end
    end
end

%% STORE BEST TotObjVCost EVER FOUND
BestTotObjVCost(cycle,1) = BestSol.TotObjVCost;
BestTotObjVCost(cycle,2) = BestSol.Violation;
BestTotObjVCost(cycle,3) = BestSol.ShiftObjV;
BestTotObjVCost(cycle,4) = BestSol.ShiftReloCost;

%% DISPLAY CYCLE INFORMATION
disp(['ABC Cycle ' num2str(cycle) ' ' num2str(ShiftDate)...
    ' Shift ' num2str(sh) ...
```

```
                ': Best TotObjVCost = ' ...
                   num2str(BestTotObjVCost(cycle)) ...
                '; ObjV = ' num2str(BestSol.ShiftObjV) ...
                '; ReloCost = ' num2str(BestSol.ShiftReloCost) ...
                '; Violation = ' num2str(BestSol.Violation) ]);
        end     % End of ABC

        ANS = BestSol;
        Solution(sh) = BestSol;

        FName = sprintf('%d Solution Shift %d.mat', ShiftDate, sh);
        save([sc.path 'FinalAnswer\'] FName],'BestTotObjVCost')

    end
    FilName = sprintf('%d Solution.mat', ShiftDate);
    save([sc.path 'FinalAnswer\'] FilName],'Solution')

    Performance;
    Answer;
end
```

The following Matlab code shows the code found in *AmbAllo.m*.

```
function [AmbANS] = AmbAllo(popsol,VarMinA,VarMaxA,VarSize,HoldV,sh,...
    initialise)

    sizH = size(HoldV);    % Size of possible holding site nodes array

    % Create answer structure
    AmbANS.Position = [];
    AmbANS.NumAmb = [];
    AmbANS.pA = [];
    AmbANS.Constraints = [];
    AmbANS.Violation = [];
    AmbANS.HrObjV = [];
    AmbANS.ShiftObjV = [];
    AmbANS.YZ = [];

    AMB = repmat(AmbANS,[12,1]);

    H.Hlocation = [];
    H.AmbAllocation = [];
    H.YZ = [];
    H.Amb = [];

    Hcov = repmat(H,[12,1]);

    % Ambulance allocation
    for tp = 1:12
        AMB(tp).Position = popsol.Position;

        if initialise == 1
            AMB(tp).NumAmb = ...
                round(unifrnd(VarMinA,VarMaxA,VarSize)).*AMB(tp).Position;
            popsol.NumAmb(tp,:) = AMB(tp).NumAmb;
        else
            AMB(tp).NumAmb = popsol.NumAmb(tp,:);
        end

        % Evaluate obj value
        [AMB(tp).ObjV,AMB(tp).pA,AMB(tp).Constraints,yz,Amb] = ...
            AlgthesisAlt(popsol.Position,popsol.NumAmb(tp,:),sh,tp);

        AMB(tp).Violation = sum(AMB(tp).Constraints(2:end,:));
        AMB(tp).YZ = yz;
        Hcov(tp).Hlocation = AMB(tp).Position;
        Hcov(tp).AmbAllocation = AMB(tp).NumAmb;
        Hcov(tp).YZ = AMB(tp).YZ;
        Hcov(tp).Amb = Amb;
    end

    ShiftObjV = 0;
    HrObjV = zeros(12,1);
    ShiftNumAmb = zeros(12, sizH(1));
    ShiftConstr = zeros(13, 11);
    ShiftpA = zeros(12,1);

    % Constraint numbers as header
    ShiftConstr(1,:) = AMB(1).Constraints(1,:);

    for tp = 1:12
        ShiftObjV = ShiftObjV + AMB(tp).ObjV;
        HrObjV(tp,1) = AMB(tp).ObjV;
        ShiftConstr(tp+1,:) = AMB(tp).Constraints(2,:);
        ShiftpA(tp,1) = AMB(tp).pA;
        for h = 1:sizH(1)
            ShiftNumAmb(tp,h) = AMB(tp).NumAmb(h);
        end
    end

    AmbANS.Position = popsol.Position;
    AmbANS.NumAmb = ShiftNumAmb;
    AmbANS.pA = ShiftpA;
    AmbANS.Constraints = ShiftConstr;
    AmbANS.Violation = sum(sum(AmbANS.Constraints(2:end,:)));
    AmbANS.HrObjV = HrObjV;
    AmbANS.ShiftObjV = ShiftObjV;
    AmbANS.YZ = Hcov(1:12);
end
```

The following Matlab code shows the code found in *AlgthesisAlt.m*.

```
function [objV,pA,Constraints,yz,Amb] = AlgthesisAlt(z,y,sh,tp)

    % Load Var
    v = matfile('Var.mat');
    VarD = v.VarD;              % Demand nodes and demand rate
    V = v.V;                    % {hnum,dnum,t_{hd}<r}
    W = v.W;                    % {dnum,hnum,t_{dh}<r}
    N = v.N;                    % {dnum,dnum,t_{dd}<r}
    tcall = v.tcall;            % service time (average)
    HoldV = v.HoldV;            % Holding site nodes

    % Load scInfo
    sc = matfile('scInfo.mat');
    Scenario = sc.Scenario;     % if 1 demand is P1, if 2 demand P1 ...
        or P2
    alpha1 = sc.alpha1;         % service reliability for P1
    alpha2 = sc.alpha2;         % service reliability for P2
    NumSelect = sc.NumSelect;

    alloRelax = sc.alloRelax;   % specify whether relaxed allocation
                                % constraint or original
    selectRelax = sc.selectRelax;

    sizPosition = size(z);      % Size of array with possible holding ...
        sites

    sh1 = [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18];    % Shift 1 Hrs
    sh2 = [19, 20, 21, 22, 23, 24, 1, 2, 3, 4, 5, 6];       % Shift 2 Hrs

    if sh == 1
        shift = sh1;            % hrs equal to shift 1 hrs
        pA = sc.NumAmbDay;      % Number of ambulances available for ...
            shift1
    else
        shift = sh2;            % hrs equal to shift 2 hrs
        pA = sc.NumAmbNight;    % Number of ambulance available for ...
            shift2
    end

    obj = zeros(1,2);
    %obj(tp,objective function value)

    VarD_t = VarD(logical(VarD(:,2) == shift(tp)),:);
    % all i \in D at tp (all demand at tp)

    if Scenario == 1
        Dnode_t = VarD_t(:,1);  % Demand nodes with demand at tp
    else
        Dnode_t = [VarD_t(:,1),VarD_t(:,5),VarD_t(:,6)];
        % Dnode_t(Demand node num, 1 if Priority1, 1 if Priority 2)
    end
    sizD_t = size(VarD_t);      % Size of VarD_t

    if isempty(Dnode_t) == 1
        objV = 0;                   % objective function value
        fprintf(1, 'No demand, exiting early.\n');
        return;
    else
        M = zeros(sizD_t(1),3);     % Min num of amb needed to cover each
                                    % Dnum at tp
        %M(i,M,Rho(i,tp))
        %M(Dnum, Min num of amb, congestion rate)

        IC = zeros(pA,sizD_t(1));   % Calculate incremental coverage
        %IC(pA = 0: Dnum1, Dnum2, Dnum3,...)
        %IC(pA = 1: Dnum1, Dnum2, Dnum3,...)
        %IC(pA = etc....)

    %% INITIALISE NEIGHBOURHOODS
        % V(j,tim) (subset of all dnum, i, that can be reached from
        % hnum, j, in less than r at tp)
        V_t = V(logical(V(:,3) == shift(tp)),:);
        sizV_t = size(V_t);

        % W(i,tim) (subset of all hnum, j, that can be reached
        % from dnum, i, in less than r at tp)
        %W_t = W(logical(W(:,3) == shift(tp)),:);

        % N(i,tim) (subset of all dnum, z, that can be reached
        % from dnum, i, in less than r at tp)
        N_t = N(logical(N(:,3) == shift(tp)),:);

        %% QUEUEING THEORY
        % Loop through i \in D with demand at tp
        for i = 1:sizD_t(1)
            if Scenario == 1            % All demand calssified as P1
                N_it = N_t(logical(N_t(:,1) == Dnode_t(i)),:); % N_i^tp
                alpha = alpha1;         % Service reliability for P1
            else                        % Demand classified as P1 or P2
                N_it = N_t(logical(N_t(:,1) == Dnode_t(i,1) & ...
                    N_t(:,8) == Dnode_t(i,2) & N_t(:,9) == ...
                    Dnode_t(i,3)),:);
                if Dnode_t(i,2) == 1
                    alpha = alpha1; % Service reliability for ...
                        Priority 1
                else
                    alpha = alpha2; % Service reliability for ...
                        Priority 2
                end
            end

            lambda_it = sum(N_it(:,7));
            % Sum of demrates of z = demand rate in a queueing system
            % z \in N(Dnode_t(i),tim) that can be reached from i in less
            % than r minutes

            rho_it = lambda_it/(24/tcall);  % Traffic intensity of system

            QT = zeros(1000,2);             % Queuing theory
            %QT(number of ambulances, numerator/denomenator)

            AmbW = zeros(1000,2);           % w - number of ambulances
            %AmbW(number of ambulances, numerator/denomenator)

            denom = 0;                      % Denomenator
            % Loop though 0 to max num of ambulances available in system
            % Taylor series
            for pp = 0:pA
                denom = denom + (rho_it^pp)/(factorial(pp)); % ...
                    Denomenator
                numer = (1/factorial(pp))*(rho_it)^pp;       % Numerator
                QT(pp+1,1) = pp;                             % Num amb
                QT(pp+1,2) = numer/denom;
                if numer/denom ≤ (1-alpha)
                    AmbW(pp,1) = pp;                         % Num amb
                    AmbW(pp,2) = numer/denom;
                end
            end

            QT(all(QT==0,2),:)=[];          % Delete rows with all 0's
            AmbW(any(AmbW==0,2),:)=[];      % Delete rows with any 0's

            M(i,1) = Dnode_t(i);            % Dnum
            M(i,2) = AmbW(1,1);             % Min num of amb needed ...
                in N_it
            M(i,3) = rho_it;                % Traffic intensity of system

            % Calculate incremental coverage
            for jj = 1:pA
                IC(jj+1,i) = QT(jj,2) - QT(jj+1,2);
            end
        end

        newIC = IC([1:max(M(:,2))+1],:);

        % Loop through demand nodes with demand at tp
        for i = 1:sizD_t(1)
            for w = 1:max(M(:,2))           % Loop through newIC
                 if w ≥ M(i,2)+1
                    newIC(w+1,i) = 0;
                 end
            end
        end

        %% Hold suggested solution
        yz = zeros(sizPosition(2),3 + sizD_t(1));
        % yz(Hnum,1 if chosen, num amb placed at Hnum,...
        % print dnum1 if t_hd ≤ r, print dnum2 if t_hd ≤ r, etc)

        % Loop through hnum
        for kk = 1:sizPosition(2)
           yz(kk,1) = HoldV(kk,1);          % Hnum
           yz(kk,2) = z(kk);                % 1 if holding site chosen
           yz(kk,3) = y(kk);                % num of amb at Hnum
        end

        sizYZ = size(yz);

        % Loop through subset of all dnum that can be reached from
        % hnum at tp; V_t.
        for pp = 1:sizV_t
            % Loop through Dnum
            for i = 1:sizD_t(1)
                if Scenario == 1
                    % If V_t's Dnum = Dnode_t's Dnum
                    if V_t(pp,2) == Dnode_t(i)
                        % Loop through yz
                        for kk = 1:sizYZ(1)
                            % If yz's Hnum = V_t's Hnum and z(Hnum) = 1
                            if yz(kk,1) == V_t(pp,1) && yz(kk,2) == 1
                                yz(kk,i+3) = V_t(pp,2);
                                % Dnum is within r min from selected Hnum
                            end
                        end
                    end
                else
                    % If V_t's Dnum = Dnode_t's Dnum
                    if V_t(pp,2) == Dnode_t(i,1) && ...
                            V_t(pp,8) == Dnode_t(i,2) && ...
                            V_t(pp,9) == Dnode_t(i,3)
                        % Loop through yz
                        for kk = 1:sizYZ(1)
                            % If yz's Hnum = V_t's Hnum and z(Hnum) = 1
                            if yz(kk,1) == V_t(pp,1) && yz(kk,2) == 1
                                yz(kk,i+3) = V_t(pp,2);
                                % Dnum within r min from Hnum
                            end
                        end
                    end
                end
            end
        end

        %% Determine obj of suggested solution

        x = zeros(max(M(:,2))+1,sizD_t(1));
        % x(x(w,Dnum), x(w,Dnum), x(w,Dnum), x(w,Dnum)...)
        % 1 if Dnum is covered by at least w ambulances; first row is ...
            w = 0

        numc = zeros(sizD_t(1),1);
        % number of ambulance covering the demand node

        % Loop through demand
        for i = 1:sizD_t(1)
            % Loop through yz
            for kk = 1:sizYZ(1)
                % if yz's Dnum = Dnode_t Dnum and yz's Hnum is selected
                if yz(kk,i+3) == Dnode_t(i) && yz(kk,2) == 1
                    numc(i) = numc(i) + yz(kk,3);
                    % num amb covering Dnum
                end
            end
        end

        % Loop through demand
        for i = 1:sizD_t(1)
            % Loop through minimum required amb
            for w = 0:max(M(:,2))
                if numc(i) ≥ w
                   x(w+1,i) = 1;        % within reach of w amb
                else
                   x(w+1,i) = 0;        % not within reach of w amb
                end
            end
        end

        % Loop through demand
        for i = 1:sizD_t(1)
            % Loop through newIC, incremental coverage
            for w = 1:max(M(:,2))
                 if w ≥ M(i,2)+1       % if w ≥ minimum needed amb + 1
                    x(w+1,i) = 0;
                 end
            end
        end
    %%
        d = zeros(1, sizD_t(1));
        % d(demand rate(Dnum))

        % Loop through Demand
        for jj = 1:sizD_t(1)
            % Loop through Demand
            for pp = 1:sizD_t(1)
                if Scenario == 1
                    if VarD_t(jj,1) == Dnode_t(pp)
```

```
                        d(1,pp) = VarD_t(jj,4);
                    end
                else
                    if VarD_t(jj,1) == Dnode_t(pp,1) && ...
                            VarD_t(jj,5) == Dnode_t(pp,2) && ...
                            VarD_t(jj,6) == Dnode_t(pp,3)
                        d(1,pp) = VarD_t(jj,4);
                    end
                end
            end
        end

        icx = zeros(1,sizD_t(1));         % incremental coverage * x

        % Loop through Demand
        for kk = 1:sizD_t(1)
            icx(1,kk) = sum(x(:,kk).*newIC(:,kk));
        end

        % objective function value (minimisation)
        obj(1,1) = shift(tp);             % HrSeq
        obj(1,2) = sum(-d(1,:).*icx(1,:));% Expected coverage

    %% Inequality constraints (constraints must be converted to ≤ 0)
        Constraints = zeros(11,2);
        %Constraints(Constraint number, 1 if constraint is violated)

        % CONSTRAINT (10), ensure that every dnum, i, is covered at
        % least once within r minutes, at every tp

        % \sum_{j \in V_{i}^{tp}} y_{j}^{t} ≥ 1
        % - \sum_{j \in V_{i}^{tp}} y_{j}^{t} + 1 ≤ 0

        con1 = zeros(sizD_t(1),1);
        %Amb = zeros(sizD_t(1),3);
        Amb = zeros(sizD_t(1),4);
        %Amb(Dnum, Priority, #Demand, Min Amb, #Amb)

        % Loop through demand
        for i = 1:sizD_t(1)
            if Scenario == 1
                V_it = V_t(logical(V_t(:,2) == Dnode_t(i)),:);
            else
                V_it = V_t(logical(V_t(:,2) == Dnode_t(i,1) & ...
                    V_t(:,8) == Dnode_t(i,2) & V_t(:,9) == ...
```

```
                        Dnode_t(i,3)),:);
            end
            sizVit = size(V_it);

            numy = 0;
            % Loop through yz
            for kk = 1:sizYZ(1)
                % Loop through V_it
                for jj = 1:sizVit(1)
                    %if yz's Hnum = V_it's Hnum, then count Amb num
                    if yz(kk,1) == V_it(jj,1)
                        numy = numy + yz(kk,3);
                        % number of ambulances covering i
                    end
                end
            end
            con1(i,1) = -numy + 1;

            %Amb(Dnum, Priority, #Demand, Min Amb, #Amb)
            Amb(i,1) = Dnode_t(i,1);
            if Scenario == 1
                Amb(i,2) = 1;
            else
                if Dnode_t(i,2) == 1
                    Amb(i,2) = 1;
                else
                    Amb(i,2) = 2;
                end
            end
            Amb(i,3) = VarD_t(i,4);
            Amb(i,4) = M(i,2);          % minimum amb required to ...
                cover i
            Amb(i,5) = numy;            % number of amb covering i
        end

        n1 = max(con1); % if any of con1 is positive it means one or more
                        % demand nodes are uncovered

        conrow = 1;
        Constraints(conrow,1) = 10;     % Constraint number
        if n1 > 0
            Constraints(conrow,2) = 1;
            %No ambulance covering one or more demand nodes at tp
            conrow = conrow + 1;
        else
```

```
                Constraints(conrow,2) = 0;
                conrow = conrow + 1;
            end

            % CONSTRAINT (11), ensure that enough ambulances are ...
                allocated to
            % ensure proper coverage to each Dnum i

            % \sum_{j \in V_i^tp} y_j^tp ≥ \sum_{w = 1}^{M_i^tp} x_i^w.tp
            % -\sum_{j \in V_i^tp} y_j^tp + \sum_{w = 1}^{M_i^tp} ...
                x_i^w.tp ≤ 0

            con2 = zeros(sizD_t(1),1);

            % Loop through demand
            for i = 1:sizD_t(1)
                numy = Amb(i,5);    % num of amb covering i
                con2(i,1) = -numy + sum(x(2:end,i));
            end

            % count by how many ambulances demands are uncovered
            n2 = 0;
            % Loop through demand
            for i = 1:sizD_t(1)
                if con2(i,1) > 0
                    n2 = n2 + con2(i,1);
                end
            end

            Constraints(conrow,1) = 11;
            if n2 > 0
                Constraints(conrow,2) = 1;
                % Not enough amb to ensure proper coverage of a demand node
                conrow = conrow + 1;
            else
                Constraints(conrow,2) = 0;
                conrow = conrow + 1;
            end

            % CONSTRAINT (12)
            % Demand node i is covered by w amb only if its is covered by
            % w-1 (w \in {2,3,...,M_i^tp})

            % x_i^w.tp ≤ x_i^(w-1).tp
            % x_i^w.tp - x_i^(w-1).tp ≤ 0
```

```

        sizx = size(x);
        con3a = zeros(sizx(1)-1,sizx(2));

        % Loop through x-axis (Demand)
        for pp = 1:sizx(2)
            % Loop through y-axis (num amb)
            for jj = 2:(sizx(1)-1)
                con3a(jj-1,pp) =  (x(jj+1,pp) - x(jj,pp));
                if con3a(jj-1,pp) < 0
                    con3a(jj-1,pp) = 0;
                end
            end
        end

        con3 = sum(con3a,1);        % if constr not violated it ...
            should be 0
        n3 = sum(con3);

        Constraints(conrow,1) = 12;
        if n3 > 0
            Constraints(conrow,2) = 1;
            % A Dnum is covered by w, but not by (w-1) ambulances
            conrow = conrow + 1;
        else
            Constraints(conrow,2) = 0;
            conrow = conrow + 1;
        end

        % CONSTRAINT (13)
        % Specify that available ambulances must be assigned to selected
        % stations

        con4 = zeros(sizYZ(1),1);

        % Loop through yz's Hnum
        for kk = 1:sizYZ(1)
            con4(kk,1) = -pA*yz(kk,2) + yz(kk,3);
        end

        n4 = max(con4);

        Constraints(conrow,1) = 13;
        if n4 > 0
            Constraints(conrow,2) = 1;
            % Not all amb are assigned to selected holding site nodes
            conrow = conrow + 1;
        else
            Constraints(conrow,2) = 0;
            conrow = conrow + 1;
        end

        % CONSTRAINT (19)
        % Ensure that capacity restrictions at each holding site node is
        % not exceeded

        cap = zeros(sizYZ(1),2);
        % cap(Hnum, Capacity)

        con5 = zeros(sizYZ(1),1);

        % Loop through yz's Hnum
        for pp = 1:sizYZ(1)
            cap(pp,1) = yz(pp,1);           % Hnum
            % If y(j) ≠ 0 unequal to 0, i.e. 1
            if yz(pp,3)≠ 0
                cap(pp,2) = HoldV(pp,5);    % Capacity
            end
            con5(pp) = yz(pp,3) - cap(pp,2);
            if con5(pp) < 0
               con5(pp) = 0;
            end
        end

        n5 = sum(con5);                     % num amb over limit

        Constraints(conrow,1) = 19;
        if n5 > 0
            Constraints(conrow,2) = 1;
            % One ore more Hnum capacity restrictions exceeded
            conrow = conrow + 1;
        else
            Constraints(conrow,2) = 0;
            conrow = conrow + 1;
        end

        % CONSTRAINT (20)
        % Ensure that num amb placed at Hnum j is greater than or
        % equal to 0 (the integer part of this constraint is done inside
        % ABCcon)

        con6 = zeros(sizYZ(1),1);

        % Loop through yz's Hnum
        for kk = 1:sizYZ(1)
           con6(kk) = -yz(kk,3);
           if con6(kk) < 0
               con6 = 0;
           end
        end

        n6 = sum(con6);

        Constraints(conrow,1) = 20; % Constraint number
        if n6 > 0
            Constraints(conrow,2) = 1;
            %Num ambulances at chosen Hnum is less than 0
            conrow = conrow + 1;
        else
            Constraints(conrow,2) = 0;
            conrow = conrow + 1;
        end

    % Relaxed constraint 16
        % CONSTRAINT (16)
        % Number of stations placed less than or equal to pz
        % sum(yz(:,2)) - pZ == 0 => sum(yz(:,2)) - pZ ≤ 0

        if selectRelax == 1
            con7alt = sum(yz(:,2)) - NumSelect;

            Constraints(conrow,1) = 16;
            if con7alt > 0
                Constraints(conrow,2) = 1;
                %Num of Hnum selected > num of Hnum available
                conrow = conrow + 1;
            else
                Constraints(conrow,2) = 0;
                conrow = conrow + 1;
            end
        end

    % Relaxed constrant 18
        % CONSTRAINT (18)
        % Specify that less than or exactly pA amb have to be located in
        % all time periods
        % sum(yz(:,3)) - pA == 0 => sum(yz(:,3)) - pA ≤ 0
        if alloRelax == 1
            con8alt = sum(yz(:,3)) - pA;

            Constraints(conrow,1) = 18;
            if con8alt > 0
                Constraints(conrow,2) = 1;
                % More ambulances located than is available
                conrow = conrow + 1;
            else
                Constraints(conrow,2) = 0;
                conrow = conrow + 1;
            end
        end
    %% Equality constraints (X=0)(, become X≥0 and -X≤0)

        % CONSTRAINT (16)
        % Number of stations placed equal to pz
        % sum(yz(:,2)) - pZ == 0

        if selectRelax == 0
            if sum(yz(:,2)) - NumSelect == 0
                con7 = 0;
            else
                con7 = 1;
            end

            Constraints(conrow,1) = 16;
            if con7 > 0
                Constraints(conrow,2) = 1;
                %Num of Hnum selected > num of Hnum available
                conrow = conrow + 1;
            else
                Constraints(conrow,2) = 0;
                conrow = conrow + 1;
            end
        end

        % CONSTRAINT (18)
        % Specify that pA ambulaces have to be located in all time ...
          periods
        % sum(yz(:,3)) - pA == 0

        if alloRelax == 0
```

```
            if sum(yz(:,3)) - pA == 0
                con8 = 0;
            else
                con8 = 1;
            end

            Constraints(conrow,1) = 18;
            if con8 ==1
                Constraints(conrow,2) = 1;
                %Less than or more than pA ambulances are located
                conrow = conrow + 1;
            else
                Constraints(conrow,2) = 0;
                conrow = conrow + 1;
            end
        end

    %% Integer constraints

        % CONSTRAINT (20)
        % Ensure that int num of ambulances are placed at each ...
            selected j

        con11 = zeros(sizYZ(1),1);

        % Loop through yz's Hnum
        for pp = 1:sizYZ(1)
            if mod(yz(pp,3),1) == 0
                con11(pp,1) = 0;
            else
                con11(pp,1) = 1;
            end
        end

        n11 = sum(con11);

        Constraints(conrow,1) = 20;
        if n11 > 0
            Constraints(conrow,2) = 1;
            %Number of ambulances placed is not integer
            conrow = conrow + 1;
        else
            Constraints(conrow,2) = 0;
            conrow = conrow + 1;
        end
```

```
%% Binary constraints

    % CONSTRAINT (21)
    % Check that all values of x() are 0 or 1

    sizx = size(x);

    con12 = zeros(sizx(1)*sizx(2),1);

    % Loop through y-axis
    for kk = 1:sizx(1)
        % Loop through x-axis
        for mm = 1:sizx(2)
            if x(kk,mm) ≠ 0 && x(kk,mm) ≠ 1
                con12(kk*mm,1) = 1;
            else
                con12(kk*mm,1) = 0;
            end
        end
    end

    n12 = sum(con12);

    Constraints(conrow,1) = 21;
    if n12 > 0
        Constraints(conrow,2) = 1;
        %x_i^tp is not binary
        conrow = conrow + 1;
    else
        Constraints(conrow,2) = 0;
        conrow = conrow + 1;
    end

    % CONSTRAINT (22)
    % Check that all values of z(Hnum) is 0 or 1

    con13 = zeros(sizYZ(1),1);

    for kk = 1:sizYZ(1)
        if yz(kk,2) ≠ 0 && yz(kk,2) ≠ 1
            con13(kk,1) = 100;
        else
            con13(kk,1) = 0;
        end
        end

        n13 = sum(con13);

        Constraints(conrow,1) = 22;
        if n13 > 0
            Constraints(conrow,2) = 1;
            %z_j is not binary
        else
            Constraints(conrow,2) = 0;
        end

        %% Objective function value
        objV = obj(1,2);

        Constraints = Constraints';
    end
end
```

Appendix G

Results

This appendix contains the result tables for the four instances. The results were discussed in Chapter 6.

Table G.1: Scenario 1 results with relaxed ambulance allocation constraint, (4.15).

	Violation	# Demand	Expected coverage	% Expected cov	# Relocations	Relo Cost
20160101						
Shift1	0	510	477.25	93.58%	187	64.73
Shift2	0	435	383.85	88.24%	236	89.14
20160102						
Shift1	0	531	506.40	95.37%	160	42.02
Shift2	0	427	406.51	95.20%	125	32.12
20160103						
Shift1	0	514	488.07	94.96%	177	52.77
Shift2	0	431	405.50	94.08%	171	46.76
20160104						
Shift1	0	526	502.32	95.50%	175	45.17
Shift2	0	461	427.32	92.69%	197	58.65
20160105						
Shift1	0	545	522.75	95.92%	91	22.43
Shift2	0	474	454.48	95.88%	73	16.08
20160106						
Shift1	0	493	472.57	95.86%	50	12.04
Shift2	0	443	425.14	95.97%	79	17.40
20160107						
Shift1	0	518	491.77	94.94%	145	37.54
Shift2	0	451	432.27	95.85%	77	20.43

Table G.2: Scenario 1 results with original ambulance allocation constraint, (3.26).

	Violation	# Demand	Expected coverage	% Expected cov	# Relocations	Relo cost
20160101						
Shift1	13	510	469.84	92.13%	374	136.27
Shift2	11	435	389.27	89.49%	325	120.04
20160102						
Shift1	12	531	483.32	91.02%	277	89.38
Shift2	10	427	402.45	94.25%	256	81.49
20160103						
Shift1	10	514	473.02	92.03%	288	94.13
Shift2	11	431	403.19	93.55%	265	96.26
20160104						
Shift1	11	526	495.97	94.29%	316	105.21
Shift2	11	461	387.92	84.15%	322	120.70
20160105						
Shift1	10	545	492.10	90.29%	313	89.94
Shift2	10	474	445.02	93.89%	243	85.41
20160106						
Shift1	11	493	416.63	84.51%	292	110.25
Shift2	10	443	415.50	93.79%	306	102.56
20160107						
Shift1	11	518	482.85	93.21%	301	104.90
Shift2	12	451	420.21	93.17%	310	109.49

Table G.3: Scenario 1 results for observed data, if solution for relaxed ambulance allocation constraint, (4.15), was implemented.

	Violation	# Demand	Expected coverage	% Expected cov	# Relocations	Relo cost
20160101						
Shift1	3	672	609.58	90.71%	187	64.73
Shift2	6	522	459.52	88.03%	236	89.14
20160102						
Shift1	1	604	565.99	93.71%	160	42.02
Shift2	0	519	484.62	93.38%	125	32.12
20160103						
Shift1	1	545	509.67	93.52%	177	52.77
Shift2	4	481	440.53	91.59%	171	46.76
20160104						
Shift1	4	633	596.89	94.30%	175	45.17
Shift2	3	520	465.71	89.56%	197	58.65
20160105						
Shift1	1	618	589.09	95.32%	91	22.43
Shift2	2	524	494.71	94.41%	73	16.08
20160106						
Shift1	2	501	478.52	95.51%	50	12.04
Shift2	5	509	475.62	93.44%	79	17.40
20160107						
Shift1	2	527	492.72	93.50%	145	37.54
Shift2	3	426	402.45	94.47%	77	20.43

Table G.4: Scenario 1 results for observed data, if solution for original ambulance allocation constraint, (3.26), was implemented.

	Violation	**# Demand**	**Expected coverage**	**% Expected cov**	**# Relocations**	**Relo cost**
20160101						
Shift1	16	672	603.90	89.87%	374	136.27
Shift2	17	522	447.12	85.65%	325	120.04
20160102						
Shift1	12	604	539.82	89.37%	277	89.38
Shift2	12	519	479.30	92.35%	256	81.49
20160103						
Shift1	10	545	495.75	90.96%	288	94.13
Shift2	13	481	449.19	93.39%	265	96.26
20160104						
Shift1	12	633	594.25	93.88%	316	105.21
Shift2	17	520	427.18	82.15%	322	120.70
20160105						
Shift1	13	618	545.78	88.31%	313	89.94
Shift2	12	524	484.68	92.50%	243	85.41
20160106						
Shift1	11	501	412.85	82.41%	292	110.25
Shift2	12	509	458.24	90.03%	306	102.56
20160107						
Shift1	12	527	482.94	91.64%	301	104.90
Shift2	13	426	389.29	91.38%	310	109.49

Table G.5: Scenario 2 results with relaxed ambulance allocation constraint, (4.15).

	Violation	# Demand	Expected coverage	% Expected cov	# Relocations	Relo cost
20160101						
Shift1	0	436	421.29	96.63%	111	30.79
Shift2	0	372	358.05	96.25%	66	15.57
20160102						
Shift1	0	444	428.41	96.49%	89	21.59
Shift2	0	359	347.65	96.84%	84	18.96
20160103						
Shift1	0	460	444.11	96.54%	115	27.39
Shift2	0	361	348.80	96.62%	23	5.74
20160104						
Shift1	0	440	425.69	96.75%	203	65.59
Shift2	0	372	359.89	96.74%	37	7.30
20160105						
Shift1	0	464	448.15	96.58%	188	58.54
Shift2	0	394	380.29	96.52%	51	11.60
20160106						
Shift1	0	447	432.95	96.86%	131	40.58
Shift2	0	363	349.83	96.37%	126	34.03
20160107						
Shift1	0	437	422.78	96.75%	182	50.57
Shift2	0	362	350.14	96.72%	50	11.45

Table G.6: Scenario 2 results with original ambulance allocation constraint, (3.26).

	Violation	# Demand	Expected coverage	% Expected cov	# Relocations	Relo cost
20160101						
Shift1	10	436	417.45	95.75%	311	114.45
Shift2	11	372	358.02	96.24%	275	93.03
20160102						
Shift1	11	444	416.21	93.74%	345	136.74
Shift2	8	359	341.85	95.22%	297	92.15
20160103						
Shift1	10	460	443.66	96.45%	299	119.71
Shift2	11	361	349.13	96.71%	371	116.70
20160104						
Shift1	9	440	419.11	95.25%	254	87.13
Shift2	9	372	354.62	95.33%	292	92.50
20160105						
Shift1	9	464	432.89	93.30%	346	122.78
Shift2	11	394	379.35	96.28%	363	124.17
20160106						
Shift1	12	447	419.06	93.75%	335	119.92
Shift2	11	363	349.97	96.41%	241	80.00
20160107						
Shift1	10	437	422.68	96.72%	261	84.73
Shift2	10	362	347.71	96.05%	267	95.43

Table G.7: Scenario 2 results for observed data, if solution for relaxed ambulance allocation constraint, (4.15), was implemented.

	Violation	# Demand	Expected coverage	% Expected cov	# Relocations	Relo cost
20160101						
Shift1	5	672	631.28	93.94%	111	30.79
Shift2	5	522	490.69	94.00%	66	15.57
20160102						
Shift1	4	604	575.63	95.30%	89	21.59
Shift2	8	519	490.79	94.56%	84	18.96
20160103						
Shift1	2	545	519.27	95.28%	115	27.39
Shift2	6	481	455.14	94.62%	23	5.74
20160104						
Shift1	4	633	600.13	94.81%	203	65.59
Shift2	1	520	498.07	95.78%	37	7.30
20160105						
Shift1	1	618	593.47	96.03%	188	58.54
Shift2	6	524	493.46	94.17%	51	11.60
20160106						
Shift1	2	501	478.28	95.46%	131	40.58
Shift2	2	509	481.53	94.60%	126	34.03
20160107						
Shift1	2	527	506.71	96.15%	182	50.57
Shift2	3	426	405.38	95.16%	50	11.45

Table G.8: Scenario 2 results for observed data, if solution for original ambulance allocation constraint, (3.26), was implemented.

	Violation	**# Demand**	**Expected coverage**	**% Expected cov**	**# Relocations**	**Relo cost**
20160101						
Shift1	15	672	634.42	94.41%	311	114.45
Shift2	11	522	493.90	94.62%	275	93.03
20160102						
Shift1	18	604	555.44	91.96%	345	136.74
Shift2	10	519	489.16	94.25%	297	92.15
20160103						
Shift1	15	545	516.34	94.74%	299	119.71
Shift2	17	481	457.84	95.19%	371	116.70
20160104						
Shift1	12	633	596.19	94.18%	254	87.13
Shift2	11	520	484.74	93.22%	292	92.50
20160105						
Shift1	11	618	565.55	91.51%	346	122.78
Shift2	12	524	498.39	95.11%	363	124.17
20160106						
Shift1	15	501	460.87	91.99%	335	119.92
Shift2	13	509	487.01	95.68%	241	80.00
20160107						
Shift1	14	527	497.73	94.45%	261	84.73
Shift2	11	426	405.01	95.07%	267	95.43