Determining optimal police patrol deployments

A simulation-based optimisation approach combining Agent-Based Modelling and Genetic Algorithms

Natacha Laure Chenevoy

The University of Leeds

Faculty of Social Sciences

School of Law

Submitted in accordance with the requirements for the degree of Doctor of Philosophy

December 2022

Intellectual Property

The candidate confirms that the work submitted is her own and that appropriate credit has been given where reference has been made to the work of others.

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

© 2022 The University of Leeds, Natacha Laure Chenevoy

Acknowledgements

This work would not have been possible without the guidance and support of many people around me. First of all, I would like to thank my supervisors, Daniel Birks and Nick Malleson who have spent many hours talking, reading, and advising. Thank you both for your much needed encouragement and expertise. I also want to express my gratitude to the Economic and Social Research Council for funding this research.

I am especially grateful to my PhD "siblings" Keiran and Verity whose support has been invaluable through for the highs and the lows. Keiran, thank you for letting me pick your brain on various maths problems and for sharing your daily tea brews with me. Verity, thanks for guiding me through this journey with an ever-kind listening ear.

Undertaking postgraduate studies can be a trying experience, especially when locked down in one's apartment during a global pandemic for the better part of it. I want to thank my family and all the wonderful friends who have kept my spirits up and provided much needed distraction from this research, including: Wilson, Caz, Yann, Josh, and Tania, among many others. I must also thank Tim and Gail – my British family – for their kindness and generosity over the years.

And finally my biggest thank you is to Iain for supporting and encouraging me through this research. I would not have not made it without you by my side.

Abstract

One of the most important tasks faced by police agencies concerns the strategic deployment of patrols in order to respond to calls whilst also deterring crime. Current deployment strategies typically lack robustness as they are often based on tradition. As police agencies are encouraged to improve the effectiveness and efficiency of their services, it is essential to devise advanced patrol deployments that are based on recent scientific evidence.

Most existing models of patrol deployments are too simplistic, and are thus unable to provide a realistic representation of the complexity of patrol activities. Furthermore, past studies have tended to focus on individual aspects of patrol deployment such as efficiency, reactive effectiveness or proactive effectiveness, rather than consider them all together as part of the same problem.

This thesis proposes to develop a decision-support tool for informing better patrol deployment designs. This tool consists of a simulation-based optimisation approach combining two key components: (1) an agent-based model (ABM) of patrol activities used to evaluate the performance of the system under a given deployment configuration and (2) a genetic algorithm (GA) which seeks to speed up the search for optimal deployments. While the developed framework is designed to be applicable to any police force, a case study is provided for the city of Detroit in order to demonstrate its potential.

The developed decision-support tool shows considerable potential in informing more cost-effective patrol deployments. First, the ABM of patrol activities allows for exploration of the impact of various deployment decisions that police agencies are unable to experiment with in the real world. Second, the GA makes it possible to optimise patrol deployments by identifying 'good' solutions, which provide faster responses to incidents and deter crime in key areas, in reasonable time.

Contents

List of Figures

List of Tables

Glossary

ABM Agent-Based Model.

CAD Computer Aided Dispatch.

CFS Calls For Service.

COP Constrained Optimisation Problem.

GA Genetic Algorithm.

GIS Geographic Information Systems.

HMICFS Her Majesty's Inspectorate of Constabulary and the Fire Service.

HPC High Performance Computing.

HQM Hypercube Queuing Model.

IPV Intimate Partner Violence.

LP Linear Programming.

MCLP Maximum Covering Location Problem.

OR Operations Reasearch.

PCAM Patrol Car Allocation Model.

PDOP Patrol Deployment Optimisation Problem.

RSS Random Sampling Subset.

RST Random Sampling Technique.

Chapter 1

Introduction

Police agencies are tasked with deploying patrols across the force in order to meet demand. Their performance in doing so is often evaluated on the basis of two metrics: efficiency and effectiveness. A key objective of a police force is thus to deploy patrols in a cost-effective manner. Current deployment strategies are often not optimised and instead rely on tradition. Furthermore, randomised field trials are often not practical for police agencies to implement, as they present logistical and ethical challenges.

This thesis proposes to build a decision-support tool to explore the problem of patrol deployment optimisation. Key to this tool is a simulation model which is able to emulate patrol activities across the force and evaluate their performance. In order to automate the search through many deployment alternatives, the tool is composed of a search-optimisation algorithm that is able to find better deployment designs in reasonable time.

This chapter provides the introduction to this dissertation. In it, the background to, and motivation behind this research are first described. Then, the aim and objectives are defined, followed by an overview of the general structure of the thesis.

1.1 Background

Policing: an overview

At a high level, policing involves maintaining public order and safety as well as investigating and preventing crime. Modern policing was established in the 19th century as a response to the inadequacy of traditional forms of law enforcement, such as private security forces and citizen militias. Policing has a centralised command structure with a hierarchical organisation, uniformed officers, and police stations as bases of operation. Patrol and investigation are critical functions of policing.

The basic structure and organisation of policing have remained largely unchanged since its establishment. Policing has expanded to include activities such as community policing and social services, but the organisation is still focused on maintaining public order and safety through a centralised and professional law enforcement agency.

This thesis focuses mainly on the model of policing that is implemented in the UK and the US. While the basic structure and organisation of policing are similar across many countries, some principles of policing may not apply universally. Factors such as the role of police officers, the organisation of police forces, and cultural and social factors can all contribute to differences in policing practices.

The pressure on police agencies

Police agencies have reported struggling to balance supply and demand due to the pressure of recent funding cuts (College of Policing, 2015). With less money to spend on resources, a more cost-effective approach to how police services are provided is needed (NPCC, 2017), as inefficient solutions can mean poor performance, wasted money, and loss of lives and property. As such, police agencies have been encouraged to optimise the use of their existing workforce by implementing processes that are both effective and efficient (Home Office News Team, 2017; NPCC, 2017; Winsor, 2019).

There is no universal definition of 'police demand' in the literature (Laufs et al., 2020). Instead, the term has been defined in various ways in different studies (Davies and Bowers, 2019). In simple terms, police demand essentially consists in the need for police resources or the "actions expected of the police with the goal of maintaining safety and public order" (Laufs et al., 2020, p. 9). This thesis is concerned with external demand received from the public. This typically includes reactive demand – i.e. responding to arising incidents, and proactive demand – i.e. preventing crime (College of Policing, 2015; NPCC, 2017).

Patrol deployment

One key area where improvements could drastically enhance the efficiency and effectiveness of police agencies relates to the deployment of patrols. Patrol units are deployed across the force to prevent and reduce crime through the act of patrolling as well as respond to arising emergencies and disasters. Although foot patrols may be used in the context of community policing, most patrol units that are tasked with responding to incidents are motorised to provide faster responses, with one or two officers on board.

While on patrol, police officers must balance intersecting responsibilities. First, they are required to proactively patrol a designated area to prevent disorder (Fleming and Grabosky, 2009). A number of studies have shown that visible patrols to key high-crime areas can effectively deter crime via raising the perception of risk of potential offenders (see e.g. Braga, 2002; Braga and Weisburd, 2010; Braga, 2001; Braga et al., 1999; Cook, 1980; Eck, 1997; Eck, 2002; Ratcliffe et al., 2011; Sherman and Weisburd, 1995; Skogan and Frydl, 2004; Weisburd and Eck, 2004) as well as promote perceptions of safety (Bradford et al., 2009b; Hawdon and Ryan, 2003) and reduce citizen fears concerning local neighbourhood crime (Zhao et al., 2002).

Second, patrol units are tasked with responding to real-time incidents with the view to stopping crime and anti-social behaviour as it happens, as well as apprehending and bringing offenders to justice (HMICFRS, 2019). An average police force in England and Wales receives approximately 338 '999' calls a day, for reasons ranging from noise complaints to significant emergencies (College of Policing, 2015). These calls typically undergo a process of triage at a police force's command and control room. Emergency calls (i.e. dangerous in-progress crime or linked with life-threatening injuries) require immediate response from the closest and most appropriate available unit, while lower-priority calls (e.g. reporting of stolen goods, traffic and parking disputes) may be held in a queue (Edleston and Bartlett, 2012).

Considering both the reactive and proactive aspects of police responsibilities, a key problem for police agencies relates to the strategic deployment of patrol units across the force so as to maximise the effectiveness of their services at a minimal cost. It is this problem, here called the Police Deployment Optimisation Problem (PDOP), that is the focus of this thesis.

Current methods of police patrol deployment tend to be based on tradition rather than evidence. With the recent growth of computational techniques and data collected, new data-driven approaches can be used to inform a more efficient and effective service. Various studies have identified spatial (see e.g. Clarke and Harris, 1992; "Hot spots of predatory crime: routine activities and the criminology of place" 1989; Hunter and Jeffrey, 1992; Pease, 1991; Weisburd, 2015), temporal (Boulton et al., 2017; Tompson and Bowers, 2013; Vaughan et al., 2018) and spatiotemporal (Polvi et al., 1991; Ratcliffe, 2002; Sagovsky and Johnson, 2007) patterns in police demand. This suggests that preempting when and where demand will arise and deploying patrols to these specific areas and times may yield a more cost-effective service. In doing so, police can improve (1) their effectiveness by positioning patrols in a configuration which is most suited to meet proactive and reactive demand and (2) their efficiency by reducing the number of patrols required.

Limitations of existing research on patrol deployment

Research on police patrol deployment has increased in recent years. However, these efforts feature a number of limitations which can be summarised around two points:

First, while a number of models of patrol activities have been developed, these have tended to be too simplistic to fully emulate the complexity of police patrols and thus provide meaningful insights into the problem of patrol deployment. These models are typically equation-based (see for instance Chaiken and Dormont, 1978b; Curtin et al., 2010; Edleston and Bartlett, 2012; Green and Kolesar, 1984; Leigh et al., 2019) or low-fidelity simulations of police patrol activities (e.g. Birks and Townsley, 2018; Bosse and Gerritsen, 2009; Reis et al., 2006).

Second, most of the existing research relevant to the PDOP has tended to focus on one of the following aspects of the problem:

- *efficiency*: minimising the number of officers required on a given shift to provide the service (e.g. Chaiken and Dormont, 1978b; Edleston and Bartlett, 2012; Green and Kolesar, 1984; Taylor and Huxley, 1989),
- *reactive effectiveness*: minimising travel distance to Calls For Service (CFS) (see for instance Chow et al., 2015; Curtin et al., 2010; Mitchell, 1972) or
- *proactive effectiveness*: maximising crime deterrence through patrolling (e.g. Chen et al., 2018; Chen et al., 2015; Chen et al., 2017; Reis et al., 2006).

These aspects of police deployment designs are arguably all interdependent parts of the same problem and should as such be considered alongside each other when seeking to optimise patrol deployment (Leigh et al., 2019). An in-depth literature review on the existing research related to the problem of police deployment is provided in Chapter 2 of this thesis.

1.2 Research aim and objectives

The idea for this research emerged from the need to explore a version of the PDOP which explicitly considers efficiency, reactive effectiveness and proactive effectiveness together. Therefore, the main aim of this work is to develop a decision-support tool for informing efficient police patrol deployments that effectively deter crime while also providing timely responses to arising incidents.

This aim is broken down in this thesis into several smaller objectives, which can be summarised as follows:

1. Defining and formulating the Police Deployment Optimisation Problem.
2. Designing and validating a high-fidelity model (agent-based model) of police patrol activities in which the performance of various deployment strategies can be accurately evaluated.
3. Applying the model to explore the outcome of various deployment designs for the case study of a real police force.
4. Designing an efficient metaheuristic algorithms (genetic algorithm) from which to derive high-quality solutions to the Police Deployment Optimisation Problem in an acceptable time.
5. Applying the resulting optimisation tool to the case study of a real police force.

1.3 Thesis structure

This thesis is organised into seven chapters. The contents of the chapters are as follows:

Chapter 2 introduces the challenges of police deployment and formulates the PDOP central to this thesis. The chapter then provides a review of the existing studies on police deployment and, in the process, justifies the methodology chosen in this thesis: a simulation-based optimisation approach which combines an Agent-Based Model and a Genetic Algorithm. Having highlighted a number of methodological and theoretical gaps in the literature, the chapter ends with a list of key areas in which this thesis contributes to the body of research on police patrol deployment.

Chapter 3 describes in details the ABM built in this thesis which simulates the activities of motorised police patrols throughout their shift. The chapter follows the popular "Overview, Design concepts and Details" (ODD) protocol (Grimm et al., 2020) to enable model reproducibility.

Chapter 4 introduces the exemplar police force of Detroit Police Department (DPD) on which the decision-support tool built in this thesis is applied. The chapter presents the various data sources obtained for Detroit and the steps undertaken to pre-process them prior to using in the models. Finally, examining the content of the data, the chapter provides various visualisations of the temporal and spatial aspects of supply and demand in Detroit.

In Chapter 5 the ABM is applied to the city of Detroit. First, the sensitivity of the ABM to perturbations in the values of some key chosen parameters is assessed. Then, the Agent-Based Model (ABM) is validated by comparing the population-level patterns of incident dispatch and travel time produced by the model against those observed in the real system (Detroit). Finally, a series of simulation experiments are conducted in which the ABM is used to explore the impact of several deployment designs on the performance of the system. These experiments highlight the potential of ABM as a computational laboratory in which various deployment decisions can be tested and their consequences anticipated away from logistical and ethical constraint.

Chapter 6 introduces the need to employ a metaheuristic search-optimisation algorithm to greatly speed up the search for optimal solutions to the PDOP. The chapter introduces Genetic Algorithms (GAs) – the chosen metaheuristics in this thesis – including their key concepts, advantages and limitations. Then, the design decisions that were made when designing the GAs in this thesis are detailed. The chapter ends by introducing the logistical decisions that were put in place to monitor performance and prevent over-fitting. The resulting tool is a simulation-based GA in which an ABM is used by the GA to evaluate the performance of solutions at each step of the learning process.

In Chapter 7 the simulation-based GA is applied to the case study of Detroit. The chapter presents results from two GA variants proposed to address two distinct versions of the PDOP. The first one is a single-objective GA which seeks to identify the best deployment design to minimise response time given a maximum number of available patrols. The second one is a multi-objective GA. It searches for solutions to the PDOP which satisfy multiple conflicting objectives around reactive effectiveness, proactive effectiveness and efficiency.

Chapter 8 summarises the findings presented in previous chapters and highlights their implications with regards to the PDOP. The chapter concludes by discussing a number of limitations to the chosen methodology and proposes potential paths for future enquiry.

Chapter 2

Literature review: police patrol deployment

2.1 Introduction

Police agencies need to deploy on-duty patrols so that they may both respond to calls and deter crime before it takes place. It is this problem, here called the Patrol Deployment Problem, that is the focus of this thesis.

Section 2.2 begins this chapter by introducing the objectives of effectiveness and efficiency inherent to the police as a public service. Then, Section 2.3 formalises the Patrol Deployment Problem and introduces the concept of 'hotspot patrolling' commonly used to deploy patrols based on historical demand. Importantly, the section introduces a variant of the Patrol Deployment Problem, called the Patrol Deployment Optimisation Problem (PDOP) in this thesis, which will be explored in later chapters. Section 2.4 highlights existing research on the problem of patrol deployment and introduces the methodology chosen in this thesis. Finally, Section 2.5 discusses how this thesis contributes to the field of patrol deployment studies by addressing some of the theoretical and methodological gaps identified in the literature.

2.2 Objectives of police agencies

A primary goal of police agencies is to safeguard the public by preventing crime and providing rapid responses to calls for service, despite increasingly constrained resources. The police have two main performance concerns: (1) effectiveness and (2) efficiency, which translate in a variety of performance metrics (see Davis, 2012 for a report of international performance evaluation), some of which are detailed below.

As previously stated, this thesis focuses primarily on UK and US models of policing. While there are commonalities in how policing is organised across countries, it is important to recognise that there can be differences in how policing is done in different parts of the world, and that some principles may not apply universally.

Effectiveness

Evaluating effectiveness consists in assessing whether the provided services fulfil their goal in reducing crime or increasing security (Cordner, 1989; NPCC, 2017). There are two distinct aspects to consider when assessing police effectiveness: (1) reactive effectiveness (calls are effectively responded to) and (2) proactive effectiveness (crimes are effectively prevented).

One possible metric that applies to both reactive and proactive policing is the public's perception of the police. Policing is a public service and as such, the public's perception of police work is often seen as an indicator of police effectiveness (Fielding and Innes, 2006). Survey data offer valuable insights into citizen's satisfaction. In England and Wales for example, the levels of public satisfaction are quantified at the national level by the anonymous Crime Survey for England and Wales (Office for National Statistics, 2019) and at the borough level in London by the Metropolitan Police Public Attitude Survey (METPAS) (BMG Research, 2014).

Nonetheless, public satisfaction alone does not suffice in assessing the effectiveness of a police system. In recent years, harm-focused policing approaches have emerged, seeking to provide a range of criteria – based on estimated harm – that can be used to prioritise the response to or prevention of certain incidents or crimes (Ratcliffe, 2015; Sherman, 2013). Ideally, the assessment of reactive and proactive effectiveness would directly relate to the prevention of harm, by prioritising certain calls (reactive effectiveness) or patrolling certain areas (proactive effectiveness). However, measuring the harm that may have been prevented as a direct consequence of police actions is not a trivial task. As such, studies on patrol deployment have instead focused on surrogate performance measures. For reactive effectiveness, for instance, one commonly used metric is the response time (Leigh et al., 2019; Mukhopadhyay et al., 2016; Surkis et al., 1970).

Reactive effectiveness (response time)

Response time is defined as the time interval between the call coming in and responders arriving at the scene (D'Amico et al., 2002; Stevens et al., 1980; Zaki et al., 1997). It thus encompasses both (1) the dispatch time of the calls – i.e., the time from receiving the call to dispatching a unit, and (2) the travel time to the location of the call (Chen et al., 2019).

Rapid response to CFS has long been an integral part of the toolkit used by police forces (Karn, 2013). A quick response can (1) provide immediate lifesaving intervention, (2) increase the likelihood of catching the offender at the scene or nearby (3) improve the chances of identifying and locating witnesses (4) provide immediate gathering of physical evidence, (5) create citizen satisfaction with the police, and (6) enhance the reputation of the police department (Karn, 2013). In fact, police response time has been identified as the strongest predictor of citizen satisfaction with police actions (Spelman and Brown, 1984). In turns, satisfied citizens are more likely to quickly report a crime to the police (Bradford et al., 2009a; Bradford et al., 2009c; Spelman and Brown, 1984), which ultimately increases the odds of a subsequent arrest and reinforces a positive citizen satisfaction with police services. All in all, response time provides a tangible measure of reactive effectiveness which relates to both harm prevention and citizen satisfaction (Bodily, 1978; Green and Kolesar, 1984).

Police agencies are often evaluated against response time targets, as it is one of the easiest metrics to review and compare either within a single jurisdiction or between jurisdictions (Goldberg, 2004). For instance, guidelines in England and Wales suggest that an emergency incident should be attended within 15 mins in urban area or 20 mins in rural settings (NPCC, 2017). This means that a response unit is required to reach the scene of emergency incidents within the target time, or the response will be classed as 'failed' (NPCC, 2017). Police forces are assessed and ranked annually based upon the percentage of calls they received which have met these targets. As such, minimising response time to meet national standards is a key goal in designing police effective patrol deployments (Winsor, 2019).

Response time can be assessed using data produced by Computer Aided Dispatch (CAD) systems, which monitor the location and status of units in the field using GPS trackers on vehicles (McEwen et al., 2004). The produced data includes the time and location of each incident as well as information about the response, such as the time of dispatch, time of arrival, and time completed. The availability of such data can ultimately provide evidence of the success or failure of a deployment plan (Church et al., 2001). In the absence of CAD data, response distance is sometimes used as a proxy for response time. This assumption is based on studies that showed a correlation between response times and response distances in the US (Priest and Carter, 1999). However, while this assumption may hold true for cities with grid-like road networks, it may not apply elsewhere.

Proactive effectiveness

Within the field of policing, researchers agree that the presence of police is particularly important in the prevention of criminal activities (see Kelling et al., 1974; Koper, 1995). In their literature review, Dau et al. (2021) identified 49 studies that examined the effects of police presence or evaluated its measurement, and found that police presence had a positive impact on crime reduction, particularly for motor theft, property, violence, and guns.

However, assessing whether a preventative programme has been effective is challenging. Although studies have historically looked at metrics such as call volume, crime rate, or number of arrests, the interpretation of these statistics with regards to proactive effectiveness can be ambiguous (Davis, 2012; Kelling, 1992). For example, a high number of arrests could either indicate that (1) the police are engaging in aggressive enforcement or that (2) they are not being proactive and are allowing crimes to occur. Similarly, crime rates may not provide an exact picture of the crime landscape as a decrease in the crime rate of a certain area may be the result of the displacement of crime to another area. Furthermore, an apparent decrease in call volume may be linked to the decrease in the public's trust in the police (under-reporting), as opposed to a true reduction in crime. As previously mentioned, patrolling officers can effectively increase the safety of citizens by deterring crime (Cook, 1980). As such, the amount of officer time spent on their patrol routes represent another possible metric of proactive effectiveness. Finally, crime rates may also change for reasons that are unrelated to policing, due to environmental or socio-economic factors for example.

Efficiency

While effectiveness is the priority, police forces in England and Wales have reported struggling to meet demand with the resources at hand (College of Policing, 2015). Police funding has decreased by 19% between 2010/2011 and 2018/2019 (Winsor, 2019) and police officer numbers have fallen by 19,569 (14%) since the peak in 2009 (Home Office, 2020). As per 2017/2018, there was approximately one officer per 480 members of the public (Home Office, 2020; Winsor, 2019), which is 50 more than in 2010 (Winsor, 2019). This places high pressure on police forces to balance supply and demand and meet their increasing responsibilities with fewer resources per member of the public (College of Policing, 2015).

Police efficiency can be interpreted as the cost effectiveness of the service provided, i.e. the correct utilisation of available resources (NPCC, 2017; Sun, 2002). Since the cost of manpower is the single largest cost factor in providing emergency services, it is also one of the central elements in service efficiency. In other words, efficiency can be evaluated by looking at the total number of officers on duty at a given time.

With less money to spend on resources, a more efficient approach to how policing services are provided is needed (NPCC, 2017), as inefficient solutions can mean poor performance, wasted money, and loss of lives and property. Police forces have been encouraged to improve the performance of their processes by optimising the use of their existing workforce (Home Office News Team, 2017; Winsor, 2019). According to Her Majesty's Inspectorate of Constabulary and the Fire Service (HMICFS), police forces have to understand the resources available to them (supply) and the demand they need to meet (Winsor, 2019). They should know what they can achieve within a particular budget; i.e. what level of service they can provide within current resources and be able to assess the level of service they could provide with more resources – or less (Winsor, 2019).

Summary

Finding a unifying metric for assessing the performance of a police system is a significant challenge. To quantify effectiveness, researchers have turned to a variety of metrics, as summarised in Table 2.1. However, with the reduction of resources, police agencies seek more efficient operations to provide the best possible service with the resources at hand.

Table 2.1: Possible performance measures of a police reactive and proactive effectiveness.

Type of effectiveness	**Metrics**	**Possible datasets**
Reactive	Response time	CAD data
Proactive	Call volume	CAD data
	Crime rate	Crime data
	Time spent on patrol	CAD data
Proactive and reactive	Citizen satisfaction	Survey data (METPAS or CSEW)

Police effectiveness and efficiency are conflicting interdependent measures (Taylor and Huxley, 1989). For example, having too many officers on duty (surpluses) may waste resources, while having too few (shortages) increases response times and over-burdens officers. As such, when designing police deployment strategies, effectiveness and efficiency should not be looked at in isolation to assess police performance. Instead, a multifaceted measurement system of police performance – one that assesses both reactive and proactive effectiveness as well as overall efficiency – can arguably provide policy makers with a more complete understanding of the quality of their service (Davis, 2012; Moore and Braga, 2003).

All in all, police agencies are concerned with providing a cost-effective service which keeps the public safe at a minimum cost. A key consideration when achieving this goal relates to the deployment of patrols. This task, here called the Patrol Deployment Problem is detailed in the next section.

2.3 The Patrol Deployment Problem formulation

2.3.1 Patrol deployment

In the policing model adopted in Western societies, a police force is commonly divided into police command areas (e.g. precincts, districts, divisions, etc.). These are the largest areas of a police force which typically contain a police station and a number of small patrol beats – or sectors (Larson, 1978). Units typically patrol the streets of designated beats and are dispatched in response to incidents arising within the boundaries of their districts. The various geographical boundaries of a police force intrinsically shape patrol activities and as such, they are inherent to police deployment decisions.

Patrol deployment encompasses the many important decisions associated with the patrol function, including where, when, and in what number patrol officers should be deployed to both deter crime and provide timely responses to emergency calls. Patrol deployment issues are inherently complex because police agencies must make a wide range of long-term, short-term and live decisions, the outcome of which is often not easily anticipated (Goldberg, 2004).

Long-term deployment decisions concern the overall design of the force such as the location of stations, the boundaries of the districts and the shift rosters. Although these decisions ultimately influence both effectiveness and efficiency, they rarely undergo changes due to obvious logistical challenges. Indeed, moving a police station or completely redesigning district boundaries can be highly disruptive for police agencies.

Shorter-term decisions, on the other hand, tend to be made on a daily basis. They include for example the number of on-duty patrols that should be deployed to a given area of the force during a particular shift. Finally, some decisions are made on the fly and involve moving resources around the force during the course of a shift or deciding which unit to dispatch to a given call. While live dispatching decisions are generally aided by CAD systems, there is less guidance available to police agencies to inform them on shift-by-shift short-term deployment decisions.

The Patrol Deployment Problem defined in this thesis is solely concerned with the short-term decisions involving the assignment of resources to a fixed geography. It does not seek to address long-term decisions such as designing better beat boundaries or patrol routes. More specifically, the Patrol Deployment Problem seeks to improve the cost-effectiveness of the service through designing better patrol deployments.

The problem encompasses two interconnected questions. The first one relates to staffing and scheduling and involves identifying the number of units required in each district of the force to provide a timely response to emergency calls. The second question relates to the positioning of these units within each district by asking where they should be deployed to provide faster responses (reactive) while deterring crime (proactive). The Patrol Deployment Problem is thus concerned with the number of patrols as well as their spatial positioning across the patrol beats of the force.

In an attempt to formalise the Police Deployment Problem, this thesis introduces the concept of deployment configuration, which represents the staffing of the patrol beats of a police force for a given shift. In the model developed in this thesis, patrol beats are staffed with at most one patrol unit. A deployment configuration thus encompasses both aspects of the Patrol Deployment Problem: (1) the number of deployed units and (2) the patrol beat that each unit is deployed to. An example of random deployment configuration for the example police force of Detroit Police Department is provided in Figure 2.1.

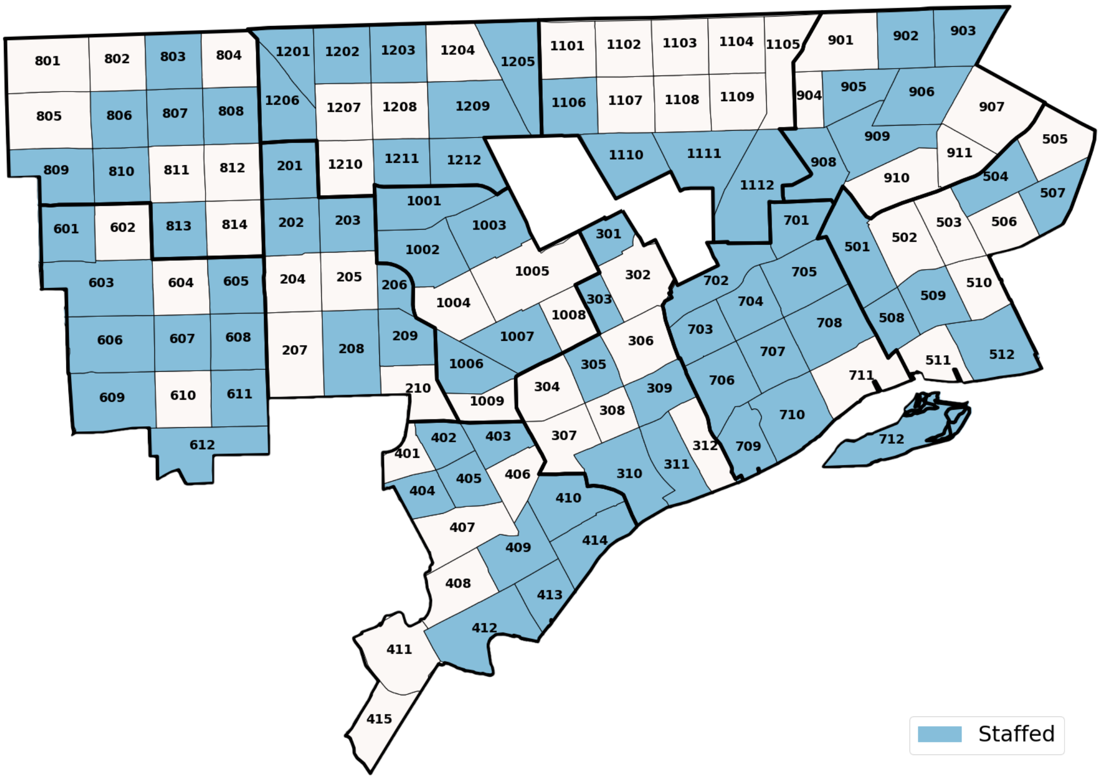

Figure 2.1: Example of deployment configuration for the city of Detroit. The blue patrol beats are staffed with a car.

The traditional approach to patrol deployment design is often based on empirical evidence (i.e. using datasets showing historical demand) and complex historical or political factors. Specifically, many police departments choose where to deploy resources using an approach called 'hotspot policing', based on the evidence that crimes cluster in space and time. The subsequent section highlights the evidence pertaining to the clustering of demand in space and time.

2.3.2 Spatial and temporal clustering of demand

Spatial clustering

In recent decades, empirical research has demonstrated that most types of crime are clustered in specific areas (Brantingham and Brantingham, 1993; Brantingham and Brantingham, 1984; Brantingham et al., 1976; Pyle, 1976; Pyle and Hanten, 1974; Rengert, 1980), which can range in size between entire neighbourhoods, particular streets (Rosser et al., 2017), or certain facilities (Bichler et al., 2013; Bowers and Johnson, 2005; Wilcox and Eck, 2011). These are areas, often referred to as crime hotspots or hot-places (Block and Block, 1995; Braga, 2005; Brantingham and Brantingham, 1982; Eck et al., 2005; "Hot spots of predatory crime: routine activities and the criminology of place" 1989; Weisburd et al., 1992), feature "a greater than average number of criminal or disorder events" (Eck et al., 2005, p. 2). This uneven distribution of crime within specific neighbourhoods has been reported in studies of a variety of crime types including drug selling (Weisburd and Green, 1994), burglary (Pease, 1991), robbery (Hunter and Jeffrey, 1992), and auto theft (Clarke and Harris, 1992). In their work on Minneapolis (USA), "Hot spots of predatory crime: routine activities and the criminology of place" (1989) found that 50% of calls for service originated from 3.3% of the city's addresses or intersections. This empirical piece is one of the first and most notable in identifying the concentration of crime at micro places within the larger area. These results were supported by a more recent study from (Weisburd, 2015), which identified that 50% of crime in several sampled cities (including Seattle) is consistently generated by around 5% of addresses in those cities.

Temporal clustering

Crime also exhibits temporal fluctuations, with clusters observed at several levels of granularity over time (NPCC, 2017; Polvi et al., 1991; Sagovsky and Johnson, 2007). At the lowest level, the concentration of crime varies by time of day (Ratcliffe, 2002; Tompson and Bowers, 2013) as a result of changes in typical activity patterns. In addition to daily trends, crime can also cluster in weekly patterns dictated by the way in which routine activities differ between days of the week (Felson and Poulsen, 2003). For instance, alcohol-related disorders are more likely to occur on Friday and Saturday nights, as more potential victims and offenders are brought together in bars and nightclubs. Over a longer scale, many crimes also exhibit seasonality (Cohn and Rotton, 2000; Farrell and Pease, 1994; Hipp et al., 2004; Landau and Fridman, 1993; Perry and Simpson, 1987), with certain months of the year featuring a disproportionate number of offences. For example, Hird and Ruparel (2007) found that 25 of the 29 crime types they studied experienced a level of seasonal trend.

With regards to calls for service, in their study of calls for service in Surrey, BC, Vaughan et al. (2018) identified distinct temporal patterns between types of calls. In particular, they found that mental health calls were shown to peak during in the mid-afternoon in the middle of the week, while Intimate Partner Violence (IPV) calls peak on Saturday and Sunday between 6:00 pm and 2:00 am. Similarly, Boulton et al. (2017) found that Lancashire Constabulary (UK) typically receives the most calls for service in the time window between 11am and 8pm.

2.3.3 Hotspot patrolling

Hotspot policing involves the deployment of police resources to demand hotspots (see "Hot spots of predatory crime: routine activities and the criminology of place" 1989; Ratcliffe, 2004; Weisburd, 2005). Areas that have experienced more crime or CFS demand in the past may be identified using hotspot mapping techniques such as kernel density distribution (see for instance Chainey et al., 2008; LeBeau, 2001; Leigh et al., 2019) or statistical tests inspired from the field of epidemiology (Johnson et al., 2009).

Amongst the various types of hotspot policing interventions, perhaps the most important innovation in policing in recent years concerns hotspot patrolling (Weisburd and Eck, 2004). Hotspot patrolling 'focuses on small geographic places or areas where crime is concentrated' (Koper, 2014) with the view to deterring crime. The idea is to preempt where demand is likely to arise and position officers with access to high demand areas.

The Minneapolis Hot Spots Patrol experiment (Sherman and Weisburd, 1995) was the first to offer compelling evidence about the effectiveness of policing at identified crime hot spots. It showed that roughly doubling the level of patrol in crime "hot spots" resulted in modest, but statistically significant, reductions in total calls for service, ranging from 6% to 13%, in treatment places relative to control places (Sherman and Weisburd, 1995). Since then, a growing body of research has provided evidence that hotspot patrolling can indeed produce significant crime prevention gains at high-crime "hot spots" (see, e.g., Braga, 2002; Braga and Weisburd, 2010; Braga, 2001; Braga et al., 1999; Eck, 1997; Eck, 2002; Ratcliffe et al., 2011; Skogan and Frydl, 2004; Weisburd and Eck, 2004).

Alongside deterring crime, officers on patrol should arguably also be positioned in a configuration which allows them to effectively respond to possible arising CFS. Although CFS are usually considered to belong to reactive policing, there is a proactive element in preempting where they will arise so units can provide a faster response. Those emergency incidents which take place outside of patrolled areas need to be reached by units within response time targets. As such, there is a need for patrol deployment designs that take into account both (1) crime deterrence through patrolling crime hotspots and (2) timely response to arising CFS.

2.3.4 The Patrol Deployment Optimisation Problem (PDOP)

Hotspot policing mainly addresses the positioning of a predetermined number of patrols to key areas of the force with the view to providing an effective service. However, with efficiency in mind, it is critical for police agencies to also keep the number of officers low as they incur a considerable cost to the agency. In this context, the Patrol Deployment Problem can be interpreted as a multi-objective optimisation problem which seeks to identify the optimal number and spatial positioning of patrols across the patrol beats of a police force – i.e. a deployment configuration – to maximise effectiveness at a minimal cost. This formulation of the problem is called the Patrol Deployment Optimisation Problem (PDOP) in the rest of this thesis, and is detailed in this section.

Optimisation problems definition

The goal of an optimisation problem is to find the best parameters that either minimise or maximise a desired objective function. The choice of objective function depends on the optimisation problem at hand. Depending on the number of objectives considered, the optimisation problem can be described as single-objective or multi-objective.

In a single-objective variant of the PDOP, the problem is simplified to optimising a specific metric exclusively – such as the average response time – given a maximum number of available patrols to deploy.

The multi-objective variant of the PDOP, on the other hand, seeks to optimise various conflicting metrics. As the PDOP is concerned with both reactive and proactive performance as well as effectiveness and efficiency, its objective function may include any of the following objectives:

- *minimising* the average response time.
- *minimising* the number of 'failed' responses. As previously explained, these are the responses where the response time exceeded a predefined threshold (e.g. 15 mins).
- *maximising* the crime deterrence using patrol time as a proxy, or better yet a crime deterrence score that takes into account historical crime levels on patrolled street segments (see Chapter 3 for details about how this is calculated in this thesis).
- *minimising* the number of patrols deployed.

The search space of an optimisation problem consists of all the possible parameter values (Eiben and Smith, 2015) and can be very large depending on the problem. The aim is to find a point (or a set of points) in this search space which gives the optimal solution; i.e., which satisfies all the chosen objectives. The search space of the PDOP contains the set of all possible deployment configurations. The PDOP is a constrained problem, as police agencies are understandably limited in the number of units they are able to deploy on a given shift, due to financial constraints. As such, the number of agents deployed cannot exceed this constraint, which is specific to the chosen police force.

A problem's search space may be composed of one or more local and global optima, i.e. solutions which optimise the problem's objectives. Single-objective optimisation problems are typically unimodal, i.e. the desired outcome is the single best solution called the global optimum. Multi-objective optimisation problems, however, tend to be multimodal ones. The preferred outcome is composed of multiple 'good' feasible solutions (local optima) as opposed to a single best solution (global optimum), since the latter may not always be implementable in the real world. For instance, some police agencies may only have a certain number of officers available on a particular day, or they may have obligations to keep the average response time below a certain threshold.

The need for models

Searching for 'better' deployment configurations, in the context of the PDOP, involves evaluating the performance of many candidate configurations. Although police agencies can assess the performance of their system under the current patrol deployment, they are unable to determine whether better alternatives exist without first implementing these through randomised controlled trials. This is the idea behind 'evidence-based policing', a concept first introduced by Sherman in 1998. The core idea of this approach is that police practice can be made far more effective by repeated controlled field experiments.

While field experiments are often considered to be the 'gold standard' method of determining effectiveness, there have been very few such experiments in police agencies due to the challenges they present (see for instance Ratcliffe et al., 2011). First, the costs and logistics of the trials may dissuade police agencies who already have their resources stretched. Second, field experiments come with some ethical concerns, as some citizens will not receive the same level of service as others. Furthermore, anticipating the consequences of patrol deployment in the real world – e.g. predicting the possible delays or shortage of staff that might arise over time or result from moving resources around the force – is a difficult task. As a result, experimenting with a poor configuration may bring potential risk to people's lives (Goldberg, 2004; Miller and Knoppers, 1972).

In addition to implementation challenges, field trials show limitations when solving optimisation problems such as the PDOP. First, depending on the size of the police force, the number of possible configurations may be far too great for them all to be tested *in vivo*. Second, it is impossible to directly compare the performance of two configurations implemented in trials conducted on different days. This is because every day is different and as such it is not possible to control for either the fluctuation in demand, or the impact of environmental or societal aspects.

All in all, these challenges dramatically hinder the optimisation of police deployment, and as a result, agencies tend to resort to what they know rather than experimenting with new configurations. The advent of computer models brings a new prospect into police experimentation for the PDOP.

A model represents a simplified version of a real system, built to inform decisions about the system in question. With a model, many alternative deployment configurations may safely be evaluated *in silico*, something that would be very time consuming or even impossible for police agencies to implement in the real world. Furthermore, unintended consequences can be discovered within a model before deployment configurations are implemented in the real world. The next section highlights relevant existing research on modelling and optimising police patrol deployment.

2.4 Existing research on modelling and optimising patrol deployment

This section provides a review of past studies focusing on the modelling of police patrol deployment. Existing models typically fall into two categories: they are either deterministic (used for planning purposes) or stochastic/probabilistic (used in operational scenarios). Additionally, models are either (1) descriptive – they help understand the real-world system and evaluate its performance under certain conditions, or (2) prescriptive – they conduct an optimisation analysis and propose better deployment solutions.

2.4.1 Equation-based models

Equation-based mathematical models have been widely used to optimise emergency services such as the ambulance and fire services. There has been extensive research in this area, some of which is relevant to the PDOP, as other emergency services are faced with similar problem when positioning their resources to meet demand.

Queuing models (probabilistic)

Queuing models, are mathematical models based on probabilities which aim to capture the fundamental characteristic that emergency units may be unable to intervene in some cases as they may be already occupied. Queuing models are capable of measuring the proportion of times that a server is busy and are typically used to plan the number of resources needed to provide a service.

Based on spatially distributed queuing theory, the Hypercube Queuing Model (HQM) (Larson, 1974) is the first probabilistic model developed for facility location problems. The model is able to estimate both numbers of vehicles needed and their posting patterns by calculating selected performance measures for individual vehicles (travel time to incidents, workload, and proportion of dispatches outside its assigned beat). The HQM has been widely applied to problems such as ambulance locationing (Batta et al., 1989; Daskin, 1983) and police patrol beat design (Kwak and Leavitt, 1984; Sacks, 2000). The Rand Corporation also supported a series of research publications on a HQM for police deployment (Larson, 1975). The HQM is a non-optimising model. While it is able to evaluate a variety of performance measures given the locations of vehicles, it cannot prescribe an optimal configuration for the system. The HQM is also unable to handle call priorities or to use time-dependent rather than steady-state input data.

Green (1984) proposed a multi-priority queueing model for dispatching multiple police units in response to emergency incidents. This queuing model provides a more detailed representation of the police system than the HQM and is particularly useful for cities in which multiple-car responses are prevalent. However, much like the HQM, it remains a descriptive tool. While these descriptive models are useful for informing police on how a change in patrol deployment may impact system performance, they are not able to assist in solving the PDOP.

Some queuing models were developed in order to fulfil a prescriptive function. Perhaps the most popular of these models is the Patrol Car Allocation Model (PCAM) developed by Chaiken and Dormont (Chaiken and Dormont, 1978a; Chaiken and Dormont, 1978b). These queuing models can calculate the number of units needed to satisfy predetermined efficiency objectives set by a police department. For instance, the models can estimate how many patrol units would be needed to achieve an average travel time of five minutes to all CFS. Alternatively, the model can also estimate how many units would be needed to assure a predetermined patrol time objective for preventing crime or always having a predetermined minimum number of patrol units available to respond to emergency calls.

There are a number of limitations to the PCAM as a prescriptive tool. First, the model does not incorporate some of the complex policing behaviour such as intersector dispatching, in which command centres may dispatch units to respond to incidents across district boundaries in case of shortage in a particular district. In the PCAM, districts are taken into consideration in isolation. The model would recommend the deployment of x units in district y based on the number of historic calls received for this particular district. Furthermore, the recommended deployment does not specify which exact patrol beats within the district units should be deployed to. Finally, the PCAM can only solve single-objective variations of the PDOP. Indeed, the model's objective is to minimise the number of units to deploy. Other metrics such as response time, patrol time or minimum number of patrols are all fixed to a predetermined value. This means that the response time itself, for instance, cannot be minimised. Overall, while the PCAM is a useful tool for police agencies to quickly determine the number and placement of required patrols, it features an over-simplistic model of the police system and cannot address multi-objective versions of the PDOP.

Linear Programming (deterministic)

Within the field of Operations Reasearch (OR) – a discipline that began in the mid 1960's and that deals with the application of advanced analytical methods for optimisation – there have been many formulations of the resource allocation problem for emergency services. In particular, a very large body of OR literature has emerged around the use of Linear Programming (LP) – a type of mathematical modelling for solving optimisation problems featuring a linear objective function.

One particular type of LP problem relevant to the PDOP is concerned with location optimisation. These include p-median problems and Maximum Coverage problems. p-median models locate p facilities (or vehicles) over n demand areas such that the average distance between the facilities and the demand areas is minimised (ReVelle and Swain, 1970). The Maximum Covering Location Problem (MCLP) seeks to deploy a fixed number of facilities (or vehicles) to maximise their coverage over a given number of demand centroids (Church et al., 1974).

Location optimisation models have been widely applied to the problem of police deployment. Mitchell (1972) employed a formulation of the p-median problem to select optimal police patrol beats in Anaheim (California) that minimises travel distance to expected calls. Aly (1979) used a distance minimisation formulation for locating two new police stations in Oklahoma. Chow et al. (2015) used both p-medians and MCLP along with Geographic Information Systems (GIS) techniques to identify the optimal locations of police facilities in the Greater London Area.

In order to help design optimal police patrol beats, Curtin et al. (2010) proposed the Police Patrol Area Covering (PPAC) model combining GIS and the MCLP, that minimises the travel distance between patrol beat centroids and incidents. However, as previously discussed, the use of travel distance as a metric of response effectiveness presents limitations compared with that of response times. Furthermore, the deployment decision is solely made based on CFS incidents and as such, it ignores the proactive impact of police patrols.

In their study on the design of Leicester police patrol, Leigh et al. (2019) looked at planning patrol routes in a manner that would effectively deter crime whilst also providing quick incident response. To this end, they first identified a list of high crime hotspots using a kernel-density approach. Having determined the hotspots they then solved a MCLP by identifying the best configuration of these hotspots to allocate resources to, in order to meet possible demand within the emergency response time targets.

Another application of linear programming, which is also relevant to the PDOP, concerns the problem of scheduling. Taylor and Huxley (1989) developed an optimisation-based decision support system for San Francisco Police Department called the Police Patrol Scheduling System (PPSS). The system attempted to minimise both the number of officers and the number of shortages at the hourly level. In return, the system prescribed a minimum number of officers needed each hour of the week for a given district of the force, which produced approximately a 50% reduction in shortages and surpluses. More recently, Edleston and Bartlett (2012) devised a linear programming optimisation tool to the staffing roster problem applied to Leicestershire police force. The tool aims to (1) create demand profiles that quantify the demand on staff members and (2) to minimise the number of staff required in order to meet expected demand across a series of shifts.

For simple problems with few spatio-temporal components, LP offers an easy implementation and a fast optimisation framework. This is why they have been widely used methods in OR for emergency services, in particular for ambulance and fire services where vehicles only travel to and from the stations in response to incidents. However, there are several limitations to the LP approach which makes it not suited to the PDOP explored in this thesis. These limitations relate to its over-simplistic representation of police systems.

First, LP is based on the assumption that relationships between factors in the real world are linear. However, this is not always the case. In policing specifically, increasing the number of vehicles in service does not necessarily increase the deterrent effect in a linear fashion. For instance, according to the Koper Curve Principle (Koper, 1995), which emanated from the Minneapolis Hot Spots Policing experiment, while deterrence may be optimised by conducting random 10-15 minute patrols at least every two hours in hot spots, longer presences showed diminishing effects.

Second, LP is a deterministic approach, unlike the queuing models previously described. As such, LP models do not take into account the stochasticity of call arrival, and the increase of demand throughout the day.

Limitations of equation-based models and the need for simulation models

Table 2.2 summarises the strengths and weaknesses of LP and queuing models. Overall, while equation-based models offer a fast implementation, they can be contrived to make a number of simplifying assumptions in order to adapt the intricate dynamic processes of complex systems into mathematical formulae. This can create an undesirable level of abstraction that sets the model too far apart from the real system being studied (Epstein and Axtell, 1996).

The police dispatching system features unique properties which make it particularly challenging to model using statistical and mathematical approaches. First, police activities show numerous

Table 2.2: Strengths and weaknesses of queuing and LP models

Method	**Strengths**	**Weaknesses**
Queueing models	Include stochasticity of call arrival Good performance evaluation tools	Limited optimisation capacity Limited spatial granularity (district level)
LP models	Fast optimisation tool for simple linear models	Assumes system linearity No stochasticity

micro-level spatio-temporal interactions between multiple heterogeneous actors who are connected at multiple scales, and whose actions are interdependent. Such actors may be the various specialised police units on duty, or response units engaged in different tasks (e.g. patrolling, responding, filling in paperwork etc.).

Second, the police system, much like other emergency services, is dynamic, with the location of actors changing over time. Police services, however, differ from other emergency services in the travelling patterns of patrols which must be visible to deter crime and improve the public's feeling of safety. Patrol vehicles spend much of their shift driving along the road network, with which they interact as they estimate the shortest route, and obey traffic rules (in non emergency situations). As such, the state of police units is constantly changing in space and time, making it difficult to model with static closed-form equations (Zhang and Brown, 2013).

Third, the police system is complex, with entities interacting in non-linear and non-deterministic ways (Axelrod, 2006; Holland, 2006). This complexity can be explained, in part, by the following properties of police dispatching activities:

- *Feedback*: the outputs of one system entity (e.g. police unit) may directly or indirectly influence the inputs of another. Indeed, police units usually need to coordinate their decision-making in order to achieve optimal performance of the group as a whole. For instance, police units interact with each other through radio channels, and they make decisions and adapt their behaviours accordingly. Once an incident is allocated to a response unit, it is removed from the queue and consequently becomes unavailable for further dispatches.

- *Path dependence:* The system exhibits collective memory where future system states are constrained by previous ones. This happens when one model decision leads to other events or decisions. For instance, a unit is dispatched to incident i instead of j, and upon resolving the incident, it finds itself close to incident k, to which it is dispatched. The latter dispatch was only possible because of the first one. These series of events and decisions are difficult to anticipate without modelling the individual-level behaviours of the system (e.g. behaviour of individual units, or the state of individual incidents).

- *Non-linearity:* entities of the system interact dynamically and in rich non-linear ways. In other words, the relationship between inputs and outputs is not simple. For example, tipping points may occur where the build up of individual actions result in a backlog of unattended incidents. The system may also show signs of diminishing returns. For example, in their 1965 Beat Patrol Experiment, Bright (1969) observed that increasing the number of patrolling officers from none to one officer led to a significant drop in crime rates while further increases in officer numbers did not show any significant improvement. Similarly, a minimum number of units may need to be deployed in a given district for there to be any visible effect on response time.

All in all, equation-based models arguably require levels of abstraction that limit their utility in certain applications. Because they operate at an aggregate level, they are unable to account for individual-level behaviours pertaining to individual patrols or individual incidents. Instead, modelling the police system requires new ways of thinking and new methodological approaches, ones that are able to represent the real-world of police deployment by explicitly considering the heterogeneity, dynamics and complexity of police activities. With the advance of computing, an important body of research has turned to the use of simulations to model the police system (see Eck and Liu, 2008, for a literature review of the use of simulation in crime prevention interventions).

Simulation models present many advantages over equation-based models with regards to modelling the complexity of police system. First, simulation models are capable of developing dynamic models in which factors are able to change over time (Bonabeau, 2002; Epstein, 1999). Second, simulation models are constructed using programming languages, which are commonly less abstract and more expressive than mathematical equations. This allows for the decomposition of system complexity into manageable sub-processes. Additionally, computational models can be parameterised with empirical data (Hedstrom, 2005) instead of using arbitrary values. This is particularly useful with non-linear systems (Bonabeau, 2002).

Capturing all elements of a complex system at fine levels of granularity may initially be overwhelming (Gilbert and Troitzsch, 2005). Simulation models allow for the exploration of systems at multiple levels of abstraction, an approach called hierarchical decomposition. They may initially be built from high orders of abstraction, and details incrementally added, thus allowing effective management of model complexity (Jennings, 2001).

Simulation provides a number of strengths that overcome some of the weaknesses associated with more traditional attempts to understand complex dynamic social systems (Epstein, 1999; Epstein and Axtell, 1996). Nonetheless, as will be discussed in details in Chapter 5 of this thesis, simulation models are notoriously challenging to validate (Eck and Liu, 2008), that is, to determine if a simulation's results are valid depictions of reality. Table 2.3 provides a summary of the strengths and weaknesses of both static (equation-based) and dynamic (simulation) models.

Table 2.3: Strengths and weaknesses of mathematical and simulation models

Method	**Strengths**	**Weaknesses**
Mathematical models	Simpler and fast to implement	Too abstract (oversimple and static)
Simulation models	High-fidelity modelling Complexity modelling Hierarchical decomposition Dynamic modelling	More complex and computationally demanding Difficult to calibrate and validate

One particular simulation technique called Agent-Based Modelling (ABM) affords the ability to effectively model systems at the individual level, thus dealing with the aforementioned heterogeneity aspect of the police system. The next section will provide a description and review of the literature related to applications of this method to police deployment problems.

2.4.2 Agent-based modelling

Definition

Agent-based modelling is a simulation technique that seeks to capture how individual behavioural units interact with each other and with their environment, allowing for the emergence of aggregate behaviour from their interactions (Epstein and Axtell, 1996). With Agent-Based Models (ABMs), researchers construct synthetic environments and populate them with virtual decision makers (referred to as agents) designed to represent key system actors. According to the definition of Wooldridge and Jennings (1995), an agent is a computational system interacting with an environment, in which it is capable of flexible, autonomous action in order to meet its design objectives.

With ABM, researchers have access to individual-level information; for example, the travel time of individual units or the time they spend tending to an individual incident. ABMs thus allow researchers to explore phenomena from the bottom-up (Macy and Willer, 2002). They combine the detailed description of microscopic processes (i.e. individual behaviours of the patrol units) with the observation of their macroscopic effects. In the specific context of police deployment, ABMs allow to emulate the behaviour of individual patrol units and observe the overall average response time at the end of a shift.

ABMs are able to model large numbers of agents, which can be heterogeneous and autonomous in nature (Epstein, 1999; Epstein and Axtell, 1996). These agents may be imbued with different characteristics (e.g. various specialised response units) and behavioural rules (e.g. dispatchers versus patrol units). This ability to model heterogeneous agents is of great importance, as there are no other methods able to capture the heterogeneity inherent in social systems. Furthermore, these agents can be autonomous; each agent in the simulation perceives, reasons and acts individually. While agents may exchange information directly or indirectly through the environment, no centralised controller regulates their behaviour.

Advantages of ABMs

ABMs are a natural metaphor for heterogeneous, dynamic and complex systems such as the police system (Zhang and Brown, 2013). One important advantage of ABM over other analytical methods is its ability to provide policymakers with a unique scenario-testing environment (Axelrod, 1997; Groff and Birks, 2008). ABM allows for experiments to be performed that would otherwise be impossible due to ethical or logistical constraints (Gilbert and Troitzsch, 2005). With ABMs, researchers can prototype, test, and refine proposed intervention prior to carrying out field testing (e.g. increase/decrease in number of patrols deployed on a given shift). Once a model is constructed, its parameters calibrated and its level of realism validated, it can be used to conduct simulated experiments whereby populations of agents are instantiated in some particular configuration. Then, researchers can observe the outcome patterns (e.g. average response time) which result from the unpredictable aggregation of the repeated actions and interactions of the agents (Epstein, 1999).

ABM offers social scientists something similar to controlled experiments used to study social phenomena. Researchers can manipulate any number of influencing factors otherwise outside their control in traditional experimentation (Eck and Liu, 2008), thus allowing the exploration of dose-response relationships in endless configurations (Townsley and Birks, 2008; Townsley and Johnson, 2008). Furthermore, ABM experiments can manipulate single characteristics of a model while holding all other characteristics static. This is important because of the 'fundamental problem of causal inference' (Holland, 1984), according to which it is impossible to observe the effect of two rival treatments on the same experimental unit. ABMs allow repeated experiments under identical conditions, save for differences selected by the researcher. As such, ABMs have been widely applied to test seminal environmental criminology theories (Birks and Davies, 2017; Birks et al., 2012; Bosse and Gerritsen, 2010; Brantingham and Tita, 2008; Groff, 2007; Groff, 2008; Malleson et al., 2010; Marchione et al., 2014; Wang et al., 2008; Weisburd et al., 2017).

Finally, ABMs can be performed en masse relatively easily and quickly. Once the model has been built, minor adjustments are simple to perform (Gilbert and Troitzsch, 2005; Townsley and Birks, 2008). Conducting simulations that are equivalent to randomised controlled trials is far simpler and cheaper than in the real world. It is generally well worth the cost of building and running a model rather than trying to experiment on the actual system (Goldberg, 2004).

ABM for police deployment

A number of studies have focused on the use of ABMs for police intervention strategies. Bosse and Gerritsen (2009) proposed an ABM approach to comparing various crime prevention strategies such as moving guardians to a new location based on the density of criminals at that location, or based on the assault rate in that location, for example. The strategies were tested under different scenarios and the resulting rate of non-prevented assaults was compared. While this study provided insights into the preventative effect of moving resources throughout the force, only 9 potential strategies were tested. Birks and Townsley (2018) built a simple ABM to illustrate the potential use of such models for prototyping police deployment strategies. They conducted a series of simulated experiments using the model, including quantifying the impact of various call prioritisation strategies, numbers of responders, and idling strategies when not responding to calls. Their work aimed to demonstrate how ABM can be productively used in the field of crime science as a means to support decision-making in complex systems.

Overall, the ABMs implemented in these studies fulfil a purely descriptive function instead of prescribing better patrol deployments. In addition, these studies are simplistic in nature. In Birks and Townsley (2018) for instance, the environment is represented as a grid-based matrix. This type of environment carries a number of shortcomings when modelling patrol activities (see details below). Instead, network-based environments are arguably preferable to the study of patrol deployment.

Generally speaking, the movements of patrols are dictated by the locations to which they go, and the routes that they take to reach these locations. As a result, the street network constitutes an essential part of the urban space and is likely to play a key role in both shaping patrol movement patterns and influencing the distribution of demand itself (Davies and Bowers, 2019).

First, since streets vary substantially in terms of location and usage type, it is to be expected that relationships may be observed between their properties and travel times. For instance, motorways allow vehicles to traverse vast distances in a short amount of time while one-way roads may force cars to take longer detours. As such, although response distance – the distance between the incident and the patrol vehicle at time of dispatch – is sometimes used as a proxy for travel time, it may not be applicable to all urban settings. Because they allow for the movement of patrols along real road configurations to be modelled, network-based environments provide a more accurate estimation of travel times than grid-based ones (Davies and Bowers, 2019).

Second, street networks have been shown to influence short-term dynamics of crime (Davies and Bishop, 2013; Johnson and Bowers, 2014) as well as long-term crime patterns (Davies and Johnson, 2015; Summers and Johnson, 2017). Steenbeek and Weisburd (2016) demonstrated that 58-69% of the variability of crime could be explained at the street level. As such, two streets in the same arbitrary areal unit (e.g. grid cell) may in fact experience very different crime risks. Network-based models enable the study of demand at the street level. In the context of police deployment, this can help in designing effective patrol routes that visit those streets with the highest risk, instead of deploying units to randomly patrol broader perimeters.

Despite the advantages of network-based ABMs for modelling patrol deployment, few studies have sought to use streets as the basic units (e.g. Chen et al., 2015; Chen et al., 2017; Wise and Cheng, 2016). Of relevance to this thesis, Wise and Cheng (2016) built a network-based ABM of police response to CFS in the London borough of Camden. Their goal was to offer a way to accurately model the movement of officers by taking into account the complexity of daily tasks (e.g. patrolling, responding, transporting convicts back to station, etc.). Using the GPS data of real police vehicles in Camden, they demonstrate that their model is better able to represent the deterrent effect of patrols vehicles along the road network compared with a random patrolling model. The model developed in this thesis (which is described in detail in Chapter 3) is inspired in part by the results of this study, especially with regards to the changing roles of patrol units throughout their shift – i.e. either patrolling (available), responding (unavailable), at the scene (unavailable), returning to patrol beat to resume patrolling (available).

Summary

By allowing to incorporate the complexity and the dynamic aspect of policing, ABMs hold much potential as a decision-support tool for designing police deployment in a scientific, evidence-based manner. With ABMs, researchers can alter factors normally beyond their control, implement new interventions, and explore dose–response relationships beyond logistic and financial constraints. However, merely using simulation models for descriptive purposes does arguably not alone justify the effort to build them.

ABM provides a means for answering many 'what-if' questions, i.e. what happens with the model behaviour when some parameters are changed. For instance, policy makers can observe the effect on average response time of increasing/decreasing the number of deployed patrols. The task of exploring a model's parameter space and discovering the impact of different parameter settings can be difficult and time-consuming. Exhaustively running the model with all combinations of parameter settings is generally infeasible, but assessing model outcome by varying one parameter at a time risks overlooking complex nonlinear interactions between parameters.

To find solutions to the PDOP, the exploration of model parameters needs to be automated with a search algorithm. Such an approach, called simulation-based optimisation, involves combining (1) a search algorithm used to guide the search for solutions with (2) a simulation model (an ABM in the case of this thesis) used at each step to evaluate system outcome.

2.4.3 Simulation-based optimisation

The automated search algorithms which can be combined with a simulation model range from exact methods, seeking to find the true optimal solution, to heuristic ones, aiming to instead find 'good' solutions in a shorter amount of time.

Exact algorithms

The most intuitive method to find optimal solutions is to exhaustively explore the search space using a linear search (also called 'brute force'). In a linear search, each of the parameters associated with a solution (e.g. the number of patrols deployed to a given beat) is incremented across the entire possible range of values to find the value leading to the best outcome. Because they always find the optimal solution, these algorithms are referred to as 'exact' algorithms.

Exact algorithms guarantee to find the optimal solution in a finite amount of time. However, it is noteworthy that such 'finite amount of time' may increase exponentially depending on the dimensions of the problem (i.e. number of parameters). The PDOP is combinatorial, with a large and complex parameter space as well as highly non-linear objectives and constraints. This makes it too computationally expensive to run exact algorithms, as enumerating the entire solution space would be infeasible in polynomial time (see Pham and Karaboga, 2000, for a detailed discussion).

In such cases, a better approach to determine the optimal parameter values is to use heuristic algorithms to conduct a guided search of the solution space rather than try all possible values.

Heuristic algorithms

Heuristic methods aim to find a good solution faster than their exact counterparts, in cases where the latter are too slow or fail in solving the problem. Because they trade accuracy for speed, they are particularly useful in solving computationally demanding problems. Heuristic algorithms are thus well suited to the PDOP. Unlike exact algorithms, heuristics do not have the guarantee of finding the optimal solution in a finite amount of time. Instead, they generally find 'good' solutions in a 'reasonable amount of time' rather than finding the optimal solution.

In conventional heuristic search algorithms – such as the 'hill climbing' algorithm, new solutions that are worse than the current best are usually rejected outright. However, by doing so, these algorithms often get stuck in local optima as they always choose the best option available at each step, even if that option may not lead to the best overall solution. As a result, they would always miss a better solution separated from the current solution by a 'hill' (see illustration in Figure 2.2). A sophisticated optimisation algorithm has to include a technique for temporarily accepting a candidate solution worse than the current best solution. To this end, more recent developments in police deployment problems will now be highlighted that have have focused on

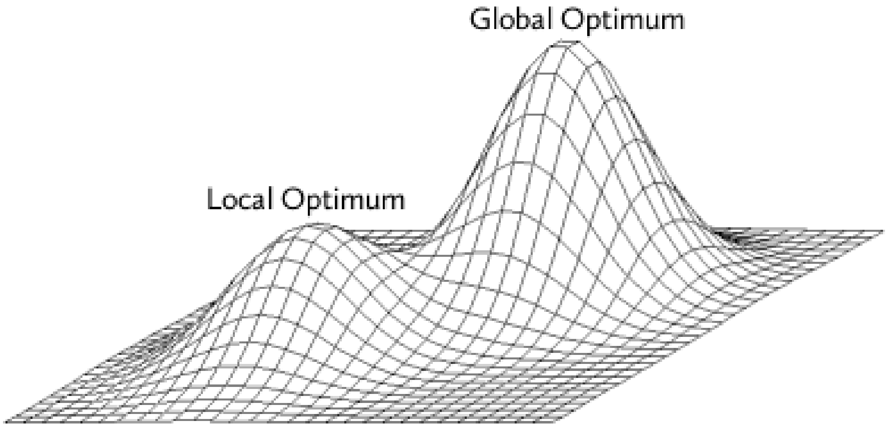

Figure 2.2: Diagram illustrating the difference between local and global optima in a problem's search space

the use of a type of heuristic algorithms called metaheuristics.

Metaheuristics algorithms

Unlike other heuristic techniques, metaheuristic algorithms are not greedy, i.e. they can accept a temporary deterioration of the solution. This allows them to explore more thoroughly the solution space, and thus reach a better solution without getting trapped in a local optimum. As such, metaheuristics are well-suited to tackling computationally demanding problems such as the PDOP.

Metaheuristics are general purpose algorithms that make few or no assumptions about the problem being optimised (Talbi, 2009). They may be viewed as upper level general methods that can be used as a guiding strategy in designing underlying heuristics (Talbi, 2009). Examples of metaheuristics algorithms include simulated annealings (Kirkpatrick et al., 1983; Metropolis et al., 1953), tabu searches (Glover, 1986), particle swarm filters (Kennedy and Eberhart, 1994; Shi and Eberhart, 1998) and evolutionary algorithms (Holland, 1984).

Tabu search algorithms avoid being trapped in local optima by building a tabu list that forbids the selection of already visited solutions and their neighbourhoods (similar solutions). Introduced by Metropolis et al. (1953), then first applied to optimisation problems by Kirkpatrick et al. (1983), simulated annealing is a stochastic search method analogous to the annealing technique in metallurgy. Evolutionary algorithms are a family of population-based metaheuristics inspired by Darwinian evolutionary theory. They are typically designed to mimic the phenomenon of natural selection using mechanisms such as reproduction, mutation, recombination, and selection. The most popular type of evolutionary algorithm is the Genetic Algorithm (GA), which is the chosen metaheuristics in this study (see Chapter 6 for details).

Much of past research on police deployment has focused on the application of metaheuristics for equation-based problems. In solving LP locationing problems, studies have used tabu searches (Adenso-Díaz and Rodríguez, 1997; Berman et al., 2009; Chen et al., 2018; Gendreau, 1997; Hansen and Mladenović, 1997; Leigh et al., 2019; Mladenović et al., 2003; Rolland et al., 1997), simulated annealings (Chiyoshi and Galvão, 2000; D'Amico et al., 2002; Murray and Church, 1996), or GAs (Alp et al., 2003). GAs have also been combined to GIS to optimally locate ambulance facilities in Niigata, Japan (Sasaki et al., 2010) and hospitals in Hong Kong (Li and Yeh, 2005), for example.

Of relevance to the spatial positioning aspect of the PDOP, Chen et al. (2018) formulated the problem of patrol route design as a Min-Max Multiple-Depot Rural Postman Problem (MMMDRPP) and developed a tabu-search-based algorithm to solve it. However, the study only considers individual aspects of the PDOP described in this thesis, which is composed of both (1) a staffing problem (how many patrols to deploy) and (2) a spatial positioning problem (where to deploy them). Furthermore, the study relies on an equation-based model, and as a result, lacks the high-fidelity modelling that can be gained by using an ABM.

The methodology chosen in this thesis is a simulation-based optimisation one, combining an ABM with a metaheuritic search algorithm. This combination harnesses both (1) the fidelity of individual-level modelling and (2) the efficiency of a metaheuristic search for optimal solutions to the PDOP. In what follows, the simulation-based optimisation approach is introduced alongside examples of relevant studies in which this methodology has previously been used.

Simulation-based optimisation of police deployment

The simulation-based optimisation approach has gained wide acceptance among researchers. Outside of the policing context, Baesler et al. (2015) applied a simulation-based optimisation approach to operating room scheduling using simulated annealing. A multi-objective simulation-based optimisation approach using simulated annealing was also presented by Mattila and Virtanen (2014) for the maintenance schedule problem of a fleet of fighter aircrafts. Simulation-based optimisation approached using GAs specifically have been applied to a variety of problems such as scheduling problems (Castiglione et al., 2007; Geyik and Dosdoğru, 2013; Korytkowski et al., 2013), designing a hospital management system (Helm et al., 2010), a shop floor layout (Tompkins and Azadivar, 1995), or a process plant (Faccenda and Tenga, 1992), as well as optimising steelworks (Paul and Chanev, 1998).

In the policing world, the simulation-based optimisation approach has primarily been employed in studies that seek better designs of patrol routes. Reis et al. (2006) developed a program called GAPatrol to assist in the planning of patrol routes. The program finds hotspots iteratively through the construction of visual maps and proposes various sets of routes. Each set of routes gives rise to a series of simulation executions using a grid-based ABM to evaluate the crime prevention performance of the routes. In the ABM, a set of criminals frequently try to commit crimes while officers try to prevent crimes. A GA is employed that seeks to optimise the patrol routes by minimising the number of crimes occurring throughout the force. To the extent of this author's knowledge, their study is the only one in the field of police deployment to utilise a simulation-based optimisation approach combining a GA and an ABM. However, their program evaluates a set of patrol routes through a simplistic grid-based models. As previously mentioned in this section, these models, unlike network-based ones, are unable to fully emulate the police system in a realistic manner. Furthermore, their study only focuses on the proactive effectiveness aspect of the PDOP and does not consider the reactive aspect of policing nor the question of efficiency that is key when deploying patrols.

Chen et al. (2015) developed a strategy called BAPS combining Bayesian methods and an ant colony algorithm to solve the problem of police patrol routing. They devises a process to design a patrol route organically using a Bayesian method to determine which hotspot to patrol in the next stage. This decision relies partly on a measure of pheromone level in order to stop repeat hotspot visiting within short spaces of time whilst also tracking when another visit is required. Finally, they use a simple network-based ABM to assess the system performance. However, the model could only optimise the patrol route of a single agent at a time. Chen et al. (2017) thus proposed an improvement upon Chen et al. (2015) by including multiple agents to the ABM and tested an emergency scenario in which patrols in the neighbourhood of an emergency are required to interrupt their patrolling to respond to it.

The above studies have primarily focused on the design of patrol routes that are beneficial to deterring crimes. However, this only constitutes one aspect of the PDOP, that is the proactive element of policing previously mentioned in Section 2.3. These studies do not consider minimising the number of patrols to deploy (i.e. efficiency) or positioning these patrols in a manner that also minimises response time to arising incidents. To the extent of this author's knowledge, the questions of staffing and positioning patrols with regards to reactive demand have primarily been studied with LP (see part on equation-based models above), and have often been the subject of individual studies rather than being considered together.

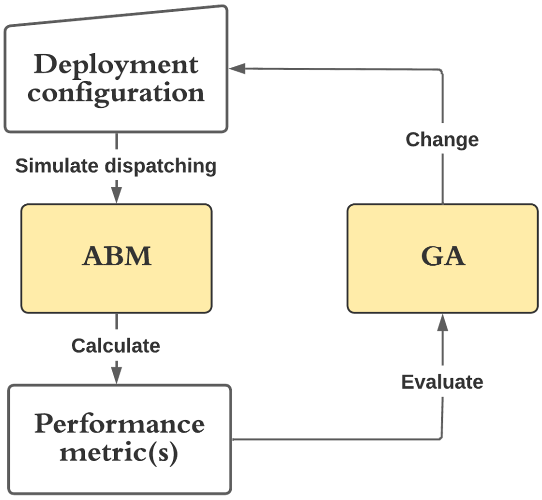

Figure 2.3: Methodology of this thesis: a simulation-based optimisation combining a GA and an ABM

In this thesis, a simulation-based optimisation approach is developed to explore the PDOP. GA is amongst the few metaheursitic algorithms capable of handling the complexity and requirements of simulation-based optimisation, and it has proven particularly useful for multidimensional optimisation problems where there are several variables to be optimised (Reeves and Rowe, 2002). As such, it is the GA that is the chosen optimisation method in this thesis. As illustrated in Figure 2.3, in the methodology chosen in this thesis, the GA conducts a search of the problem space by repeatedly evaluating candidate deployment configurations through the ABM and selecting those with the best performance. Further insights into the algorithm details will be provided in Chapter 6.

2.4.4 Summary: existing research on modelling and optimising patrol deployment

This section has introduced the various techniques that have been developed to study police patrol deployment problems and has listed the studies in which these techniques have previously been employed. Table 2.4 summarises the 6 studies that are most relevant to this research and highlights their contribution(s) to the exploration of the PDOP.

Overall, previous relevant studies have either (1) created a realistic modelling of police activities (Wise and Cheng, 2016) but without seeking to optimise police deployment or (2) sought to optimise police deployment but (a) considered only some aspects of the PDOP such as patrol route planning (Chen et al., 2018; Chen et al., 2015; Chen et al., 2017; Curtin et al., 2010; Reis et al., 2006) or staff rostering (Edleston and Bartlett, 2012) and (b) used simplistic models of police activities such as equation-based models (Chen et al., 2018; Edleston and Bartlett, 2012; Leigh et al., 2019) or ABMs with a grid-based environment (Reis et al., 2006). Additionally, these optimisation models were often concerned with either proactive or reactive policing (with the exception of Leigh et al. (2019)) and their objective function only included either efficiency or effectiveness as the sole goal. Considering these gaps in the literature, the next section highlights the contribution of this thesis in exploring the PDOP.

Table 2.4: Summary comparison of the most relevant studies to the PDOP

Study	**Problem formulation**	**Model of police system**	**Policing type**	**Optimisation technique**	**Metric**
Chen et al. (2015)	Patrol route planning	Single-agent network-based ABM	Proactive	BAPS	Effectiveness
Chen et al. (2017)	Patrol route planning	Multi-agent network-based ABM	Proactive	BAPS	Effectiveness
Chen et al. (2018)	Patrol route planning	Equation-based	Proactive	Tabu search	Effectiveness
Leigh et al. (2019)	Patrol route planning	Equation-based	Proactive and reactive	Tabu search	Effectiveness
Edleston and Bartlett (2012)	Staff rostering	N/A	Reactive	Linear programming	Effectiveness and efficiency
Reis et al. (2006)	Patrol route planning	Grid-based ABM	Proactive	GA	Effectiveness
Wise and Cheng (2016)	Modelling police activities	Network-based ABM	Proactive and reactive	N/A	N/A

2.5 Contributions of this thesis

In this section, the specific contributions of the research will be outlined. These can be divided into three categories: 1) theoretical contributions to police patrol deployment problems with the formulation of the PDOP, 2) methodological contributions to modelling police patrol and response activities 3) methodological contributions to optimising police patrol deployment.

2.5.1 Theoretical contributions to police patrol deployment problems with the PDOP

Previous studies have typically considered individual aspects of police deployment in isolation. For example, how many officers are required (queuing models), where to send officers to provide the best coverage of reactive demand (coverage models, p-median models) or which patrol routes provide the best deterrence (Chen et al., 2018; Chen et al., 2015).

However, police effectiveness and efficiency are conflicting interdependent measures which should arguably not be considered in isolation when making police deployment decisions. Additionally, most studies (apart from Leigh et al., 2019) have focused on optimising either reactive effectiveness or proactive effectiveness. These two metrics are also conflicting and interdependent as officer time is limited and split between either responding to calls or patrolling.

This study proposes a new formulation of the problem of police deployment which takes into considerations these conflicting interdependent metrics. The formulated PDOP is concerned with minimising operational costs (efficiency) while optimising system performance (effectiveness). More specifically, the PDOP seeks to identify the best deployment configuration(s) to (1) minimise the number of patrols, (2) optimise reactive effectiveness (e.g. response time, number of 'failed' responses) and (3) optimise proactive effectiveness (e.g. deterrence score).

2.5.2 Methodological contributions to modelling patrol activities with an ABM

To the extent of this author's knowledge, the few existing studies that have attempted to consider the multiple aspects of police deployment have done so using equation-based LP models (in particular Leigh et al., 2019). There is thus a gap in the literature relating to building models that accurately capture the complexity of patrol activities (apart from Wise and Cheng, 2016). In an attempt to further work in exploring the police patrol deployment problem – a context for which simulation (of the like of ABM) offers much greater flexibility and realism than closed-form models – this thesis proposes to build a high-fidelity model of patrols activities.

In order to emulate the real behaviour of police patrols, the ABM built in this thesis features:

1. the real road network for a spatially explicit model in which the driving time can be realistically estimated

2. individual agents for modelling heterogeneity and individual-level behaviours: e.g. patrolling, responding to emergency incidents

This ABM, which provides complete control and flexibility over the behaviour of individual agents, is designed for the prototyping and assessment of police deployment strategies in a realistic-looking world environment, and allows researchers to evaluate their likely performance under various demand scenarios.

2.5.3 Methodological contributions to optimising police patrol deployment

In order to explore solutions to the PDOP in acceptable time, this thesis proposes a simulation-based optimisation approach which combines an ABM with a GA. This approach, which is novel in the field of policing, harnesses the high-fidelity modelling offered by ABMs along with the high-efficiency of GA searches in complex multimodal problems of the likes of the PDOP. There is a gap in the methodological literature relating to the combined application of these two powerful techniques to the problem of police deployment.

The aspiration is to work towards a decision-support tool which offers much flexibility, providing any police agency with a portfolio of configurations from which to choose based on their particular priorities. Policy makers may thus choose tradeoffs between proactive versus reactive effectiveness or between effectiveness and efficiency.

Chapter 3

Building an agent-based model of patrol activities

3.1 Introduction

Chapter 2 introduced the challenges of police deployment and formulated the PDOP, which is the problem that this thesis focuses on. As was demonstrated in Chapter 2, agent-based modelling is arguably amongst the best suited methods available to explore patrol deployment, as it allows for the complexity of police patrol activities and responsibilities – i.e. patrolling and responding to calls – to be modelled at the individual level.

This chapter describes how the ABM central to this thesis was designed and built. The model – developed in the Python language – aims to simulate the activities of motorised police patrols throughout the course of their shift, with the view to exploring the effect of various deployment configurations on police performance. The code for this ABM is available at https://github.com/mednche/police-deployment-optimisation/src/ABM.

A visit to a command and control centre at Durham Constabulary provided a firsthand look at how police agencies typically operate (in a UK context), allowing for a deeper understanding of the workflows and challenges faced by dispatchers. This qualitative research was instrumental in shaping the design of the model, as it provided a foundation for the incorporation of real-world scenarios and considerations.

The ABM will be described according to the latest guidance from the "Overview, Design concepts and Details" (ODD) protocol (Grimm et al., 2006; Grimm et al., 2020) which is one of the most popular methods for outlining and documenting ABMs. The aim of such a protocol is to allow researchers to share and reproduce their models via a standardised format. The protocol consists of three main components:

1. *Overview*: providing an overview of the model (see Section 3.2).
2. *Design concepts*: detailing which of the common ABM design concepts are relevant to the model (see Section 3.3).
3. *Details*: elaborating on the internal mechanics of the model (see Section 3.4).

3.2 Overview

3.2.1 Aims

The ABM simulates a fleet of motorised police units, which patrol along designated routes and are dispatched by a centralised system in response to incoming calls for service. The model was built to serve as a performance evaluation tool for police resource deployment configurations, with the view to comparing multiple configurations over the same demand scenarios.

More specifically, the purpose of the ABM is to study how deployment configurations impact police reactive and proactive effectiveness. The effectiveness can be assessed using various metrics (as discussed in Chapter 2) depending on the type of policing being considered. For reactive policing, the metrics used in this model are the average response time and percentage of 'failed' responses (those for which response time was greater than a threshold). For proactive policing, a score of crime deterrence through patrolling is created to evaluate proactive effectiveness (see details below).

Patterns for model usefulness

As mentioned in Chapter 2, accurately measuring response time is key to improving reactive effectiveness. As such the model's usefulness is here evaluated by its ability to produce realistic-looking patterns of response time. This is done by comparing patterns produced by the ABM with those observed in real-world data collected for a given police force (Detroit Police Department in the case of this thesis). The following simple and general population-level patterns are used:

- Pattern 1: The simulated average incident *dispatch* time throughout a time period aligns with the average incident travel time observed in the real system.
- Pattern 2: The simulated average incident *travel* time throughout a time period aligns with the observed average incident travel time observed in the real system.

More details on the validation of the ABM using these patterns are provided in Chapter 5.

3.2.2 Entities and state variables

To represent the context in which police patrol activities are conducted, the model is made up of the following four entities:

- **The environment**: this is the static model environment for a given police force in which agents patrol and respond to CFS incidents. It is made up of:
  - the police precincts in which agents respond to CFS;
  - the police patrol beats to which agents are deployed for patrolling;
  - the road network, represented in the model as a graph along which agents move, accompanied by a dataset describing each road segment (type of road, maximum speed limit, etc.).
- **The agents**: these entities represent the motorised police patrol units.
- **The dispatcher**: this entity represents the centralised police command and control centre where calls are received, triaged and assigned to available patrol units.
- **The occurring CFS incidents**

The road network

The model is designed to be extensible to any urban environment and allows the user to choose a specific locality. The road network for this chosen locality is downloaded from OpenStreetMap and automatically converted to a simplified NetworkX graph using the Python package OSMNX (Boeing, 2017). In graph theory, a graph is a collection of points, called nodes (or vertices), and lines connecting those points, called edges. In the NetworkX graph, the nodes correspond to intersections and the edges represent segments of uninterrupted road between two intersections. When downloading a road network in OSMNX, specifying the option *network_type* = ‘drive’ creates a directed graph in which edges can only be used in a certain direction. In such a graph, one-way roads can be represented as a single unidirectional edge, while two-way roads require two unidirectional edges (one for each direction).

OpenStreetMap was chosen in this thesis as it is an open source dataset in comparison to other alternatives (e.g. the Integrated Transport Network). Although the crowd sourced nature of the dataset means it is harder to control for a consistent quality, its free accessibility brings significant reproducible benefits in a research environment, especially when combined with the convenience of the OSMNX package.

In a typical graph, nodes are not spatially bounded, but defined in relation to their neighbouring nodes. As a result, the graph does not contain information concerning the spatial location of the real world intersection that are represented by the graph nodes. In order for the graph to be embedded into 2-D space, OSMNX couples the graph with two datasets: one for its nodes and one for its edges (see details in Table 3.1 for nodes and Table 3.2 for edges).

The attributes of the graph nodes that are relevant to the model are summarised in Table 3.1. In particular, the node dataset contains information pertaining to the spatial location of the corresponding road intersections in the form of a pair of spatial coordinates. This attribute is used in the model to identify the closest graph node to each occurring CFS incident.

Variable name	Type	Description
`osmid`	Integer	Unique OpenStreetMap id for the node
`geometry`	Point	Pair of *lat-lon* coordinates representing the road intersection corresponding to the graph node

Table 3.1: Attributes of the graph nodes.

The road network is represented in the model as a directed graph. In this type of graph, an edge runs from a start node (called u in OMSNX) and an end node (called v in OSMNX). Table 3.2 summarises the attributes of the graph edges – and the road network segments they represent – that are relevant to this research. Those attributes are composed of information extracted from the OSMNX edge dataset (*osmid*, *u*, *v*, *geometry*, *oneway*, *length*, *speed_kph*), to which the attributes *patrol_beat* and *density_hist_inc* are added prior to running the model (see Section 3.4.1 for details on how these are calculated).

The *geometry* attribute of each edge allows one to faithfully recreate the true spatial shape and location of the road segment represented by each edge. This attribute takes the form of a linestring object: a list of at least 2 points (which are themselves pairs of latitude and longitude coordinates). This attribute is used in the model to identify the *patrol_beat* that each edge of the road network belongs to.

Variable name	Type and units	Description
`osmid`	Integer	Unique OpenStreetMap id for the edge
`u`	Integer	The osmid of the edge's start node
`v`	Integer	The osmid of the edge's end node
`oneway`	Boolean	Whether the road segment is a one-way road
`geometry`	Linestring	List of points (which are themselves pairs of *lat-lon* coordinates) spatially representing the road segment
`length`	Float; m	Length of the road segment
`speed_kph`	Float; km/h	Free-flow travel speed limit on the road segment
`travel_time_mins`	Float; mins	Driving time required to travel along the road segment based on *length* and *speed_kph*
`patrol_beat`	String	Name of the patrol beat in which the street segment is located
`density_hist_inc`	Float	Density of historical crimes that occurred on the road segment (see details in Part 3.4.2)

Table 3.2: Attributes of the graph edges. *Top*: attributes directly extracted from OSMNX edge dataset. *Bottom*: attributes manually added to the dataset.

The precincts

In addition to the road network, the model environment is compartmentalised in a number of areas called precincts. These areas, which are sometimes called districts or divisions by some police agencies, are the main unit of police organisation within which response units are contacted (typically over the radio) and managed (see Chapter 2) in the context of dispatching. In the model, precincts represent exclusive areas within which agents may be dispatched to occurring incidents. In this intra-dispatching context, agents are not dispatched to incidents occurring in neighbouring precincts, even if there are no available agents in that precinct.

Precincts are not represented as explicit entities in the model. Instead, they are implied through other entities such as incidents and patrol beats. In the data sources provided to the model (see Subsection 3.4.1), incidents and patrol beats come with a precinct column indicating the precinct in which they are located. As such, when an agent is assigned to a patrol beat upon initialisation, they indirectly get assigned the corresponding precinct for the duration of the shift.

The patrol beats

While precincts stand as the main unit of police organisation, it is the patrol beats – which are comprised within precincts – that represent the geographical unit at which deployment decisions are typically made by police agencies ahead of each new shift starting. As discussed in Chapter 2, police deployment decisions involve deciding which patrol beats should be patrolled, given a number of available responding units. Patrol beat areas are typically designed to be patrolled by one unit. As such, in the model, a maximum of one agent may be assigned to each patrol beat.

Variable name	Type	Description
`name`	String	Name of the patrol beat
`precinct`	String	Name of the precinct the patrol beat belongs to
`geometry`	Polygon	List of linestring objects spatially representing the shape of the beat
`centroid_node`	Integer	The osmid of the node closest to the spatial centroid of the beat
`patrol_route`	List of node osmids	Ordered list of node osmids in the beat to be visited by a patrolling agent

Table 3.3: Attributes of the police patrol beats

In the model, patrol beat entities are composed of a number of attributes (summarised in Table 3.3). These attribute include the *geometry* of the beat (a polygon shape) and a *patrol_route* visiting a number of pre-defined streets (see Subsection 3.4.2 for how these streets are selected), amongst others.

CFS incidents

The model was built with the view to being used to evaluate the performance of various police deployment configurations for a particular demand scenario. A demand scenario is here modelled by a dataset of CFS incidents occurring in the police force during the user-specified time period (e.g. Friday 3rd December, 2019 between 3pm and 12am). These incidents, which can be historical or synthetic ones (emulating future demand for example), occur as events during the simulation, which trigger the dispatching of agents.

Variable name	Type and units	Description
`call_datetime`	Date-time	Date and time for the call for service coming in
`node`	Integer	The osmid of the graph node that is closest to the incident location
`precinct`	String	Name of the precinct in which the incident occurred
`resolution_time`	Float; mins	The time period required to resolve the incident
`time_being_tended`	Float; mins	The current time spent by at least one agent at the scene of the incident
`dispatch_time`	Float; mins	The current time interval between the incident's *call_datetime* and the moment an agent is dispatched to the incident
`travel_time`	Float; mins	The current time interval between the moment an agent is dispatched to the incident and the moment an agent arrives at the scene
`status`	Integer	The current status of the incident. As part of its life cycle, an incident evolves through 4 sequential statuses: (1) 'unallocated'; (2) 'allocated unattended'; (3) 'being attended'; (4) 'resolved'
`agent`	Agent object	The agent (if any) currently dispatched to the incident. 'None' otherwise.

Table 3.4: Attributes of the CFS incidents

The attributes pertaining to the CFS incidents are summarised in Table 3.4. Among these attributes, some are static (their value is fixed throughout the simulation) such as *call_datetime*, *node*, *precinct*, *resolution_time*, while others are dynamic (their value changes as the simulation runs) such as *dispatch_time*, *travel_time*, *status* and *agent*. Prior to running the model, the spatial coordinates indicating the discrete location of each incident (pair of lat-long coordinates) are translated into the nearest graph node. This allows for incidents to be fully embedded into the road network so that routing calculations can be performed more efficiently while the model runs. As will be shown for the example of Detroit (see Chapter 4), graph edges segments tend to represent small road segments in urban settings and as such, this procedure is not expected to impact model validation nor the results of the experiments.

Incident response time is a key performance metrics in the model for evaluating police reactive effectiveness. It is made up of the following two components, as illustrated in Figure 3.1 showing an incident's life cycle:

1. the **incident dispatch time**: the time taken to dispatch an agent from the moment the call is received. By measuring the dispatch time from the moment the call comes in (as opposed to when it ends), the model seeks to emulate the fact that real dispatchers may

send resources while the call is still ongoing.

2. the **incident travel time**: the time taken by an agent to reach the scene of their assigned incident from the moment they are dispatched.

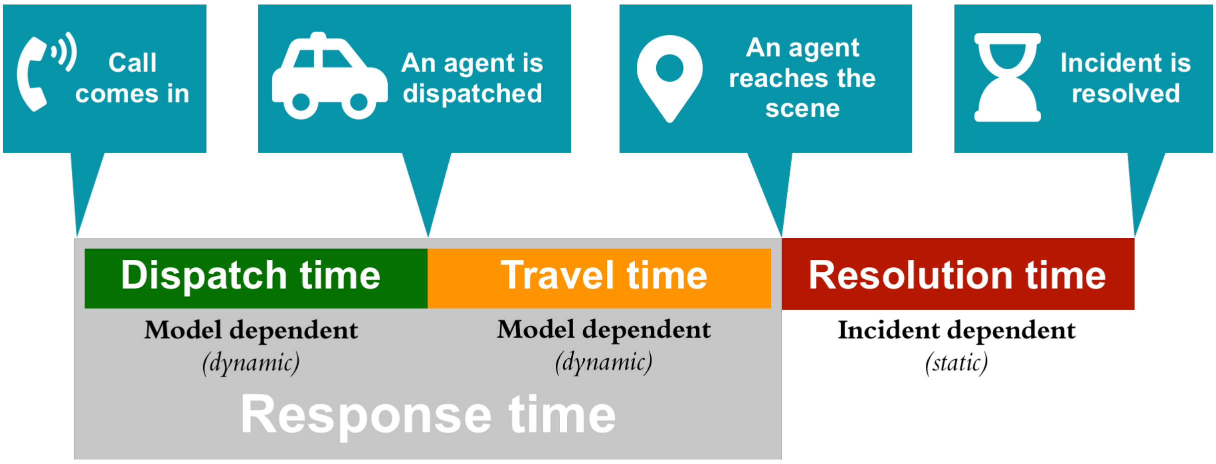

Figure 3.1: Time intervals in the incident's life cycle from call to resolution

The model described in this chapter is concerned with emergency incidents only (highest priority grade). It would not be appropriate to model incidents of multiple priorities in this early version of the model, as it would introduce too much complexity through incident hierarchy. Different incident priorities require different types of police response, with lesser priority ones often not involving police dispatch as they are often resolved via phone call. Furthermore, when responding to these lesser priority incidents, police vehicles are subject to traffic congestion and rules (e.g. traffic lights) as they are not able to use blue lights and sirens. For these reasons, non-emergency incidents are not included in this version of the model.

As a direct consequence of excluding non-emergency incidents from the model, the workload of patrols will be lower in the model than they would be in the real world and as a result. As a result, it is worth noting that performance metrics outputted by the model cannot be directly compared to those from the real world, but instead are intended to be compared across multiple configurations simulated *in vitro*.

The agents

The ABM is composed of artificial agents representing motorised police units that patrol and are dispatched to occurring CFS incidents. An agent in the model thus represents a vehicle rather than an individual officer. As previously mentioned in Chapter 2, the behaviour and movement of real police patrols are constrained by the environment (road network, precincts and patrol beats). The model assumes that agents move along the road network at the speed limit allowed on each road segment. They patrol within the boundaries of their designated *patrol_beat* (assigned upon initialisation) and may only be dispatched to incidents occurring in their *precinct*.

Variable name	Type	Static/dynamic	Description
`patrol_beat`	String	Static	Name of the patrol beat in which the agent should patrol.
`precinct`	String	Static	Name of the precinct in which the agent should respond to CFS.
`pos`	Integer	Dynamic	The osmid of the graph node on which the agent is currently located.
`status`	String	Dynamic	Whether the agent is idle, travelling to an incident or already at the scene of an incident.
`incident`	Incident object	Dynamic	The incident that the agent is currently responding to (if any).
`route`	List of nodes	Dynamic	The current route along which the agent is travelling (either to an incident node or along a patrol route).
`time_on_node`	Float	Dynamic	The time the agent has spent on current node (either a node along their route or the incident's *node* when at the scene).
`crime_deterrence`	Float	Dynamic	The agent's score of crime deterrence through patrolling.

Table 3.5: Attributes of the agents

Agents are characterised by a number of attributes as summarised in Table 3.5. One important attribute of the agents is their *status* which dictates which actions they perform when activated. An agent's status can take up one of three values:

- **'idle'**: the agent is patrolling and is available to be dispatched to occurring incidents;

- **'travelling'**: the agent has been dispatched and is currently travelling to an incident;
- **'at_scene'**: the agent has reached the scene of the incident and is currently tending to it. Agents stay at the scene of an incident for the duration of the incident's attribute *resolution_time*. While the dispatcher may dismiss a travelling agent (whose *status* is 'travelling'), it cannot do so once the agent has reached the scene (agent *status* is 'at_scene') as they are already tending to an incident.

Chapter 8 will discuss how the model might be extended in future versions to represent individual officers – rather than vehicles – in order to examine the staffing implications associated with varying numbers of officers onboard vehicles.

The dispatcher

The command and control room where dispatching decisions are made in real-world policing is simplified in the model as a single dispatcher whose role is to dispatch available agents to each incident that remains unattended. The dispatcher seeks to match allocated incidents to available agents (i.e. agent's *status* is 'idle'). The attributes of the dispatcher are summarised in Table 3.6 and further details on the dispatching mechanism are provided in the submodels in Subsection 3.4.2.

Variable name	Description
`incident_queue`	Ordered list of unallocated incidents
`avail_agents`	List of available agents throughout the force

Table 3.6: Attributes of the dispatcher

3.2.3 Model scales

Spatial scale

Building the model environment for a given police force requires importing the road network and the boundaries of the patrol beats into the model. The road network of any locality of choice can easily be imported as a NetworkX graph using the OSMNX package, as previously described. This means that, providing adequate patrol beat boundaries are available, the model can be used to emulate any chosen police force. As such, the spatial scale of the model is not fixed but instead depends upon the chosen police force.

In the real world, the real road network, patrol beats and incidents are all two-dimensional spatial elements (i.e. they possess latitude and longitude coordinates). Prior to running the model, however, these entities are incorporated into the NetworkX graph as nodes and edges (see Subsection 3.4.1 for details). In this process, the real road network becomes a graph of nodes (intersections) and edges (road segments); the patrol beats are incorporated as their centroid (graph node) and a subset of streets to patrol within them (graph edges); and the location of each incident is converted to the closest node on the graph. As such, once running, the ABM operates entirely at the graph scale, independently of spatial coordinates.

Temporal scale

The model runs at a discrete time step of one minute (although this value can easily be changed). For instance, one time step would represent the time between 14:00-14:01. The time period for which to run the model is specified by the user through the *start_datetime*, and *end_datetime* variables (see details in Subsection 3.4.1). Thus, there is no absolute concept of temporal scale in the model. As an example, in the analyses conducted in this thesis on the case study of Detroit, the model is run for a length of time equivalent to a police shift (typically 9 to 10 hours).

3.2.4 Process overview and scheduling

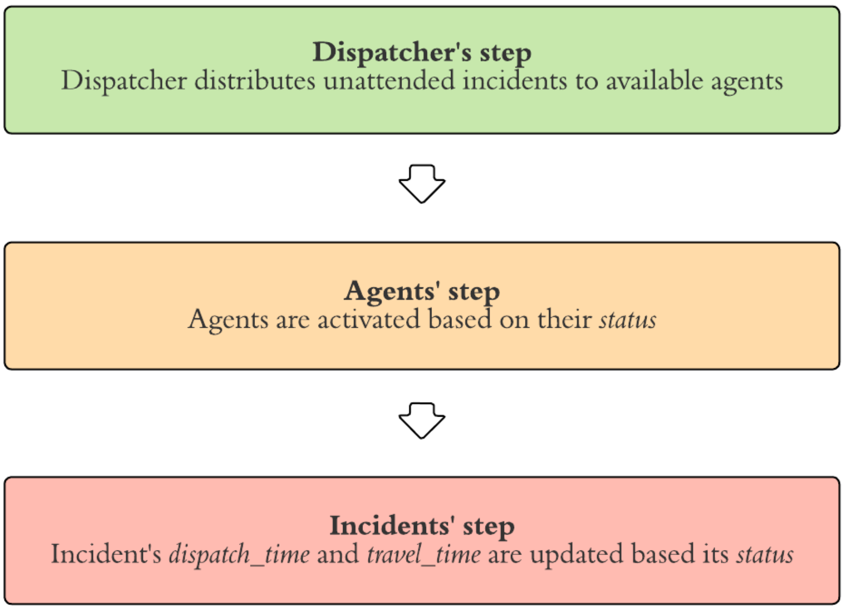

Figure 3.2: Three-step model scheduling

As illustrated in Figure 3.2, the scheduling of each model step is composed of three consecutive processes:

1. **Dispatcher's step**:
   (a) The dispatcher receives a list of incidents that occurred during the time step and adds them to the *incident_queue* (the queue of unallocated incidents).
   (b) The dispatcher executes the "distribute incidents" submodel (detailed in Subsection 3.4.2), in which it distributes each incident in the *incident_queue* to the closest available agent. When an incident gets assigned to an agent through this submodel, its *agent* attribute is updated accordingly and its *status* is changed from 'unalloacted' to 'allocated unattended'.
   (c) For each agent chosen for dispatch in step (b), the dispatcher executes its "dispatch agent to incident" submodel (detailed in Subsection 3.4.2), in which the agent's attributes *route* and *incident* are updated and their *status* becomes 'travelling'.
2. **Agents' step**: all agents are activated as illustrated in Figure 3.3. Agents start each step with a time allowance equal to the step time that they can use to perform one or multiple actions in the following order, depending on their *status*:
   - ***status* is 'travelling'**: travelling agents execute their "move" submodel (detailed in Subsection 3.4.2) in which they move along their *route* towards the *node* of their assigned incident. As they move along the route, their attribute *pos* and *route* get updated. If they reach the scene of the incident in this model step, the *status* of the incident becomes 'being attended' and their own *status* is updated to 'at_scene'.
   - ***status* is 'at_scene'**: agents at the scene of their assigned incident execute their "stay at the scene" submodel (detailed in Subsection 3.4.2) in which they remain on the incident *node* until the incident has been attended for the duration of the incident's *resolution_time*. When that is the case, their *status* is updated to 'idle'.
   - ***status* is 'idle'**: idle agents execute their "patrol" submodel (detailed in Subsection 3.4.2) in which they move (using the "move" submodel) along the *patrol_route* of their designated *patrol_beat*. As they do so, their attributes *pos* and *route* are updated.
3. **Incidents' step**: the incident's dispatch and travel times are updated based on its *status*:

- ***status* is 'unallocated'**: incident *dispatch_time* increases by one step time unit (i.e. one minute), since it has not been allocated to an agent by the dispatcher in this time step.
- ***status* is 'allocated unattended'**: incident *travel_time* increases by one step time unit (i.e. one minute), since no agent has reached the scene in this time step.

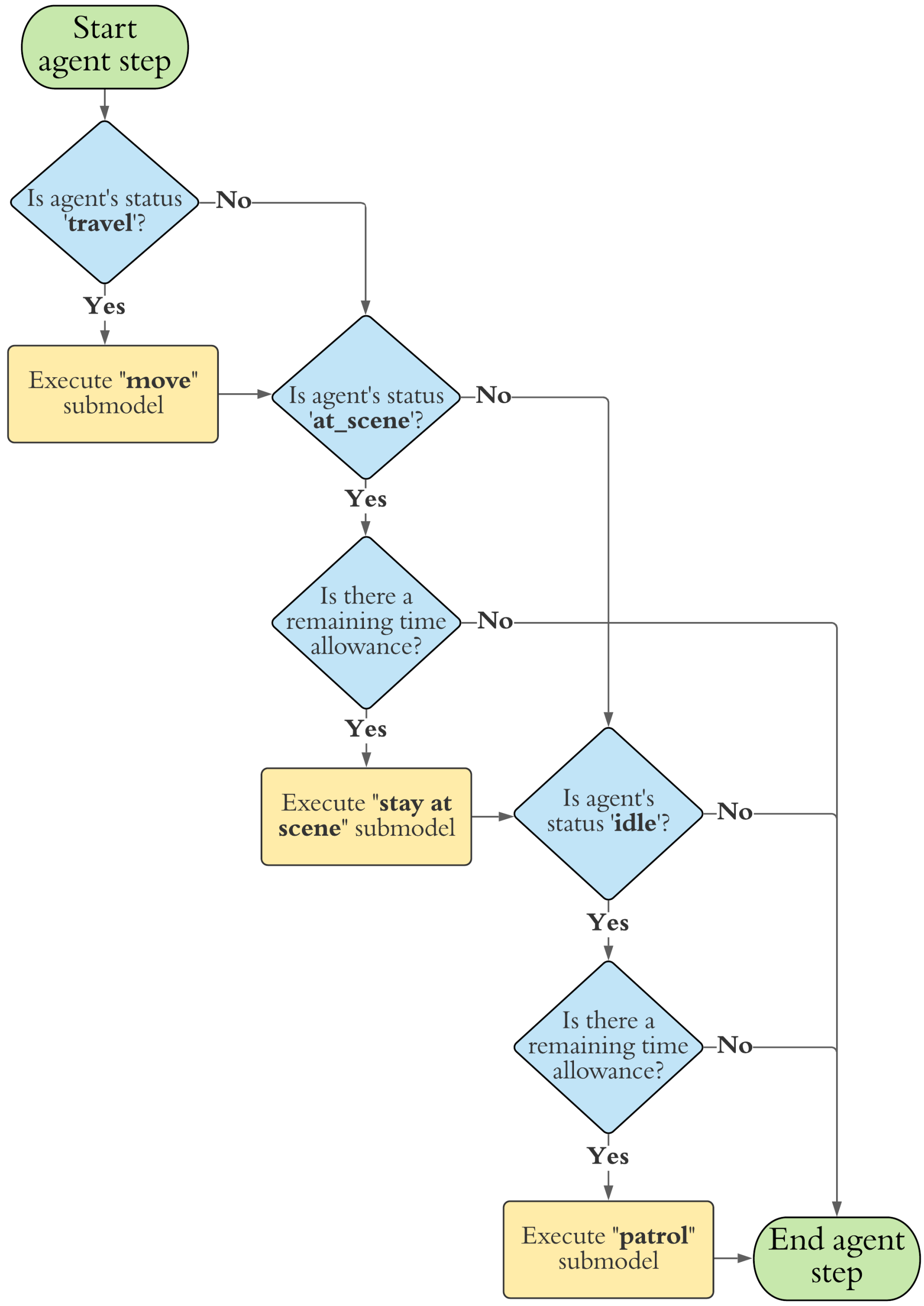

Figure 3.3: Flow diagram of an agent's activation step

The order of these steps is important. At each step of the model, the dispatcher activates first to distribute incidents to available agents. This is done prior to activating the agents, because subsequent agent actions depend on the agent's *status* which may be updated through the dispatcher's step. The *status* of an agent is updated in a cyclic fashion as illustrated in Figure 3.4.

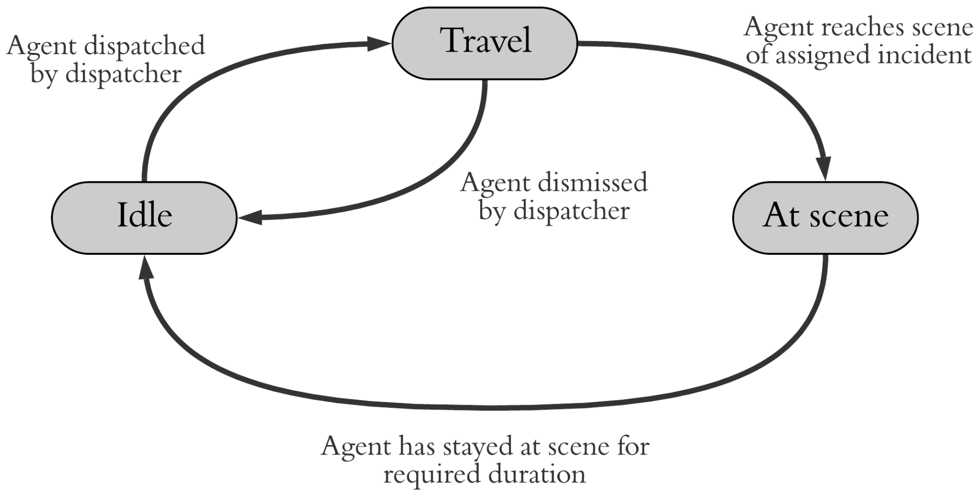

Figure 3.4: Flow diagram of the life cycle of an agent's *status*

As later detailed in Subsection 3.4.2, the dispatcher distributes incidents in order of First In First Out (FIFO). This means that the dispatcher first attempts to dispatch an agent to the oldest incident in the queue (looking at the incident attribute *call_datetime*).

Agents only interact with each other via the intermediate of the dispatcher whose decisions regarding incident distribution ultimately affect the *status* of the agents (see Section 3.3). As such, once the dispatcher has performed its step (distributed the incidents), the agents have no interactions with each other. There is thus no need to shuffle the agents before their activation in the agents' step. Consequently, agents are activated by a basic scheduler: one at a time, in the order they were added.

When activated, each agent may perform a series of actions (depending on their current *status*), in an order that mimics the behaviour of real police responders. For instance, if a travelling agent reaches the scene of their assigned incident within the time step, they can start to engage in the "stay at scene" submodel for the duration of their remaining time allowance (see details in Subsection 3.4.2). Similarly, if an agent whose status was 'at_scene' resolves their assigned incident within the time step, they can start to engage in the "patrol" submodel (detailed in

Subsection 3.4.2).

Finally, in the incidents' step, the *dispatch_time* and *travel_time* for each incident is updated depending on the *status* of the incident. This step is performed last, as both dispatcher and agents may update the *status* of some incidents in their respective steps. The *status* of an incident is updated through four consecutive stages, beginning with the call coming in and ending with its resolution (as illustrated in Figure 3.5).

When an incident occurs, its *status* is set to 'unallocated' and its *dispatch_time* - i.e., the time duration until the dispatcher finds at least one available agent to dispatch to the incident (see Figure 3.1) - begins to increase in increments of one minute every step. If no dispatch is ever made in the simulated time period, the dispatch time keeps on increasing until the end of the simulation.

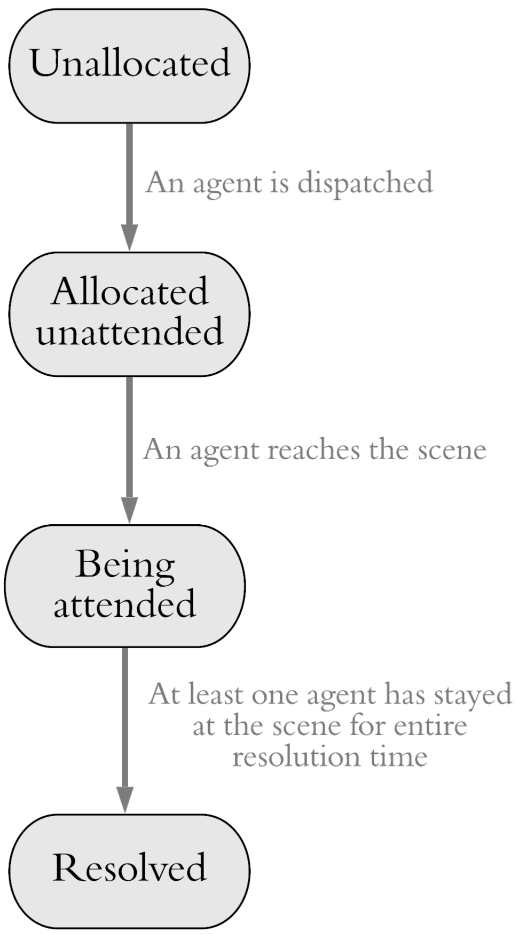

Figure 3.5: Flow diagram of the life cycle of an incident's *status*

As soon as at least one agent is dispatched to the incident, the incident's *dispatch_time* stops increasing. The *travel_time* – i.e. the time duration until an agent reaches the scene of the incident (see Figure 3.1) – begins to increase in increments of one minute per step until an agent reaches the scene.

To illustrate, consider the following example: a call comes in at 16:24 but since no agents are available in the precinct, no dispatch is made yet. After 6 model steps (i.e. simulation time is now 16:30), an agent becomes available and is dispatched to the incident by the dispatcher. The final *dispatch_time* for the incident is thus recorded as 6 minutes. It takes the agent 3 model steps to reach the scene, starting from the current model step at 16:30. When they arrive at the scene at 16:33, the incident's *travel_time* is logged in as 3 minutes. The agent is then required to stay at the scene for the incident's *resolution_time* duration, which is determined by the incident (e.g. 20 minutes).

3.3 Design concepts

Following the standard ODD structure, the following section outlines the key design concepts which underlie the model developed.

3.3.1 Basic principles

Dispatcher's method of incident distribution

As previously mentioned, the policing model that this thesis focuses on is that which is used in the UK and the US. As a result, the specific operational details described in this section may vary between countries.

In real-world policing, the severity of each incident (call) arriving to a command and control centre is first assessed by a call handler in order to assign a priority level to the incident. In England and Wales, this assessment is based on a protocol called THRIVE (Threat, Harm, Risk, Investigation opportunities, Vulnerability of the victim and Engagement level required to resolve the issue) and includes a number of factors such as whether the call relates to a crime, whether the individuals involved are vulnerable and whether there is a risk to public safety (College of Policing, 2021). Lower-priority calls, which include the reporting of stolen goods, traffic and parking disputes, and other less dire matters, may be held in queue (Edleston and Bartlett, 2012). Emergency calls, i.e. dangerous in-progress crime or linked with life-threatening injuries, on the other hand, require immediate response from the closest and most appropriate available unit (Edleston and Bartlett, 2012).

In most police agencies, dispatchers are assisted by a CAD system, which monitors the location and status of units in the field using GPS trackers on vehicles (McEwen et al., 2004). The CAD system provides dispatchers with recommendations on units for assignment based upon the event location, call type, and unit availability. Calls are then dispatched first by priority and then by the time received.

In the model developed for this thesis, the behaviour of the dispatcher revolves around the objective of keeping response time to individual incidents as low as possible, with the view to avoiding 'failed responses' (i.e. where response time is longer than a defined threshold). Since response time is composed of (1) dispatch time and (2) travel time (as explained in Section 3.2), the dispatcher aims to keep both attributes as low as possible for all incidents when executing its "distribute incidents" submodel (see Subsection 3.4.2). First, incident dispatch time is reduced by processing incidents in a First In First Out fashion so that no incident stays in the queue for an excessive length of time. Second, the dispatcher minimises incident travel time by choosing to dispatch the available agent that is closest (in terms of driving time) to the incident.

Patrolling hotspots

In real-world policing, patrol units which are not currently dispatched to an incident do not remain idle but instead typically patrol an area with the view to preventing crime. Expert consultation with a UK police force revealed that most police organisations in England and Wales use a targeted patrolling approach – as opposed to random patrolling. As such, although several idle agent behaviours were explored upon building the model – such as agents remaining stationary and returning to their police station – it is the 'patrol' behaviour that was eventually chosen, as it is more in alignment with real-world policing practices.

As mentioned in Chapter 2, hotspot patrolling has been shown in several studies to bring tangible benefits in reducing crime (see, e.g., Braga, 2002; Braga and Weisburd, 2010; Braga, 2001; Braga et al., 1999; Eck, 1997; Eck, 2002; Ratcliffe et al., 2011; Skogan and Frydl, 2004; Weisburd and Eck, 2004). Indeed, in their study on The Hague, Steenbeek and Weisburd (2016) found that 58–69% of the variability of crime can be attributed to individual street segments, with most of the remaining variability at the district level. As such, in the model, agents patrol specific 'hot' streets in their assigned patrol beat with the view to providing the most crime deterrence. These streets form part of a patrol route along which agents drive when idle. Details on how these streets are identified and on the agent patrolling behaviour are provided in the submodel descriptions in Subsection 3.4.

3.3.2 Emergence

There are three main outcomes of the model:

- **the average response time** across all incidents that were reached by an agent during the simulation;
- **the percentage of 'failed' responses**: the percentage of all incidents that were unresolved or were resolved but for which the response time exceeded a pre-defined threshold;
- **the total crime deterrence score** produced by agents throughout the simulation in their idle time.

The percentage of 'failed' responses emerges from prolonged dispatch times and travel times to individual incidents. As explained below, these times are themselves affected by the combination of two factors: (1) the initial deployment configuration (i.e. how many agents are deployed and to which patrol beats?) and (2) the CFS demand (i.e. the volume of occurring CFS events).

In particular, the *dispatch_time* of each incident emerges from:

1. The cumulative number of agents which are deployed (in the precinct and more generally in the force if using inter-sector dispatching): more agents deployed equates to a higher likelihood to find an available agent to dispatch to a given incident, ultimately leading to a shorter dispatch time.
2. The number of incidents occurring in the precinct: more incidents lead to a bigger queue of incidents to distribute to available agents.

Similarly, the *travel_time* of each incident emerges from:

1. The geometry of the precinct: larger precincts (and those with odd shapes) may result in longer travel times.
2. The initial deployment configuration: an idle agent patrolling nearer to a CFS demand hotspot may get to the scene faster when dispatched, thus leading to a shorter incident

travel time.

3. The infrastructure of the road network (edge's attributes *oneway* and *travel_time_mins*). Some road networks may require the agent to take a detour due to one-way roads, or drive at a slower speed (due to speed limit on the segment), leading to a longer incident travel time.

Finally, the amount of crime deterrence performed by an agent emerges from:

1. The initial deployment configuration: an idle agent patrolling a patrol beat featuring more historical crimes will have a higher crime deterrence score than one deployed to a patrol beat with fewer crimes, provided that they patrolled for the same amount of time.
2. The amount of time the agent spent patrolling (idle) versus responding to calls in the rest of the precinct. This depends upon the precinct in which the agent operates; in particular, the level of supply (number of other agents) and CFS demand (volume of calls received) in the precinct. For instance, an agent operating within a precinct alongside many other agents (high supply) and where the CFS demand is low will have more idle time to patrol than one in a high-demand precinct with limited supplies.

3.3.3 Adaptation & Objectives

The dispatcher is driven by its objective to keep the number of 'failed' responses as low as possible. As previously mentioned in Chapter 2, a response is counted as 'failed' when its response time exceeded a predefined threshold. To reduce the number of 'failed' responses, the dispatcher seeks to keep the response time to individual incidents as low as possible. The response time to an incident is the direct result of the dispatch time and travel time, both of which are maintained as low a level as possible by the dispatcher through adaptive behaviours arising from the following two decisions:

1. **Which incident to process first?** In its "distribute incidents" submodel (see Subsection 3.4.2), the dispatcher reduces incident dispatch time by processing incidents in a First In First Out fashion, so that no incident stays in the queue for an excessive length of time. This behaviour is modelled as indirect objective seeking: it reproduces observed behaviours in real police system and assumes that such behaviour leads to shorter dispatch times.

2. **Which available agent to dispatch?** Through its "find nearest available agent" sub-submodel (see Subsection 3.4.2), the dispatcher seeks to always dispatch the closest available agent (in terms of driving time) to each unallocated incident in its *incident_queue* within each precinct. This adaptive behaviour is modelled as direct objective seeking: the dispatcher chooses to dispatch the agent with the fastest route to the incident. To do so, the dispatcher first estimates the time it would take each candidate agent on their fastest route to reach the incident. This is done by first calculating the fastest route to the incident for each candidate agent. This is in itself a direct objective seeking behaviour: for each candidate agent, the dispatcher selects a route that minimises the edge attribute *travel_time_mins*. Then, once a fastest route has been selected for each candidate agent, the corresponding driving time is calculated as $\sum_{i=1}^{n} t_i$ where t_i is the *travel_time_mins* attribute for the i^{th} edge of the route and n is the number of edges on the route. Finally, the dispatcher engages in another direct objective seeking behaviour in which it selects for dispatch the agent whose total driving time is the shortest. More details about the "find nearest available agent" sub-submodel are available in Subsection 3.4.2.

In the model, agents make very few decisions themselves. Most routing calculations are performed either upon model initialisation or by the dispatcher (see above). To some degree, this likely reflects the roles and interactions between real police dispatchers and patrols. One situation where agents make routing calculations themselves is when 'idle' agents return to their designated patrol beat to resume patrolling after responding to an incident. Patrol routes in beats are pre-defined upon initialisation. However, when travelling to the start of the patrol route, agents perform a direct objective seeking routing in which they select a *route* that maximises the edge attribute *density_hist_inc*.

3.3.4 Learning

In this thesis, the model simulates police dispatching over a period of time typically chosen as multiple hours, but no more than a day. In such a short time period, it is not expected that a police agency would change the way it operates. As such, no learning was used in the model. Instead, the adaptive behaviour of the dispatcher and the agents described above remains constant throughout the simulated time period.

3.3.5 Prediction

At each step of the model, the dispatcher executes its "find nearest available agent" sub-submodel for each unattended incident in its *incident_queue* (see details in Subsection 3.4.2). In order to find the nearest candidate agent, the dispatcher predicts (1) the fastest route to the incident for each candidate agent and (2) the corresponding driving time for each candidate agent on their fastest route. Similarly, in their "patrol" submodel, agents returning from responding to an incident predict the crime deterrence that may be produced by choosing a particular path to the start of their patrol route (see details in Subsection 3.4.2).

3.3.6 Sensing

In this model, the dispatcher and the agents are assumed to possess an accurate knowledge of the information that they require. This information includes some of their own attributes, as well as attributes from other entities in the model.

Dispatcher

At each step, incidents enter and exit the dispatcher's *incident_queue*. Then, the dispatcher senses the *status* of all agents in the model in order to update its *avail_agents* list with agents whose current *status* is 'idle'. When executing its "find nearest available agent" submodel for each incident (see details in Subsection 3.4.2), the dispatcher senses the following information about other entities:

- where the incident took place: *node* and *precinct* for the incident;
- where each available agent is currently located: *pos* and *precinct* of available agents;
- the road network information: the *travel_time_mins* for its edges and the *osmid* for its nodes (used to estimate the driving time to the incident for each candidate agent);
- whether there was already an agent dispatched to the incident: incident's *agent* attribute.

Although precincts are not explicit entities in the model, they are sensed by the dispatcher via the *precinct* attribute of both the incidents and the agents when distributing incidents to available agents in the precinct.

Agents

In their "move" submodel (see details in Subsection 3.4.2), agents sense the road network; in particular the *travel_time_mins* attribute of its edges and the *osmid* of its nodes. Agents use this information to evaluate how far they can travel along their current *route* within their remaining time allowance for the model step. When their remaining time allowance is not sufficient to allow them to reach the next node along their route (*time_on_node* attribute), agents can 'memorise' the time spent on the furthest node reached so that they can resume their travelling where they left it at the next model step (see Subsection 3.4.2 for details).

In their "stay at the scene" submodel (see details in Subsection 3.4.2), agents sense the current value of the incident attribute *time_being_tended*. This value keeps track of the length of time since the agent first arrived at the scene and allows the agent to determine whether they are required to stay at the scene for yet another time step.

In their "patrol" submodel, agents can sense the *patrol_route* for their designated *patrol_beat*. Additionally, they sense the road network; in particular, the *density_hist_inc* of its edges and the *osmid* of its nodes. Agents use this information when resuming their patrolling behaviour after responding to an incident as they plan a route of maximum crime deterrence to the start of their designated *patrol_route* (see Subsection 3.4.2 for details).

3.3.7 Interaction

There are two kinds of interactions in this model: (1) direct interactions between the dispatcher and the agents and (2) mediated interactions between the agents themselves. The dispatcher interacts directly with agents by updating their *status*, *incident* and *route* attributes when dispatching or dismissing them. Since the *status* of an agent dictates their behaviour (patrolling if 'idle', moving towards incident if 'travelling', staying at the scene if 'at_scene'), the dispatcher's behaviour has a direct effect on the behaviour of the agents.

The direct interactions between the dispatcher and agents result in mediated interactions between the agents themselves, particularly between dispatched agents and available ones. When an agent is dispatched to a given incident, they take precedence over other available agents that could be dispatched to the same incident. As a result, the *status* of the other available agents that were not chosen for dispatch remain 'idle'. These interactions are local given the chosen intra-sector dispatching framework in this model: they only take place between agents of the same precinct.

3.3.8 Stochasticity

In its current version, the model is deterministic, i.e. it features no stochasticity. This means that the same outcome is yielded for multiple runs of the model with identical inputs. These inputs, detailed in Subsection 3.4.1, consist in the deployment configuration, time period, CFS incidents and historical crimes.

3.3.9 Collectives

There are no collectives in the model.

3.3.10 Observation

As previously mentioned, the *dispatch_time* and *travel_time* attributes of each incident dynamically increase during the simulation as a result of the actions of the dispatcher and the agents. The *dispatch_time* of an incident is incremented at each time step while the status of the incident is 'unallocated'. The *travel_time* of an incident is incremented at each time step while the status of the incident is 'allocated unattended'.

At the end of the simulation, the model outputs the *dispatch_time* and *travel_time* for each incident. Then, the overall reactive effectiveness of the police force under the chosen deployment configuration and demand scenario can be quantified by combining these outputs into the following summary statistics:

- **Average incident response time**: $\frac{\sum_{i=1}^{n}(DT_i+TT_i)}{n}$ where DT_i and TT_i are the *dispatch_time* and *travel_time* for the i^{th} incident and n is the number of incidents that occurred in the simulated time period.
- **Percentage of 'failed' responses**: the percentage of responses for which the incident response time was greater than a threshold, which is here arbitrarily set at 15 minutes.

In order to evaluate the proactive effectiveness of the police force under the chosen deployment configuration and demand scenario, the model also outputs the crime deterrence score for each agent over the time period. This score is calculated by summing the *density_hist_inc* attribute of all edges visited by a given agent while patrolling (i.e. *status* is 'idle'). The scores of individual agents are then summed into an overall crime deterrence score for the police force. In its current form, this deterrence score is merely a direct function of the time agents spend patrolling (idle). Indeed, it assumes that deterrence is a linear function of police presence and is not subject to any effects such as diminishing returns. As a result, in this current version, the deterrence score calculation merely fulfils the role of a ‘placeholder’, and what officers do when patrolling is fairly inconsequential for the model. This limitation as well as avenues to explore in order to improve the deterrence calculation are further discussed in Chapter 8.

All three of the aforementioned metrics (average response time, percentage of ‘failed’ responses and overall crime deterrence score) are emergent model behaviours resulting from the rules of the ABM. As such, their value cannot be predicted prior to running the model.

3.4 Details

3.4.1 Initialisation

This part details the steps to undertake to initialise the model prior to running it. The initialisation of the model is performed in two steps, and follows a stage of pre-processing of the data sources, as illustrated in Figure 3.6. The model initialisation itself is split in two steps: (1) initialising the model environment and (2) initialising the agents and dispatcher. It was ensured that as many pre-processing actions as possible were performed ahead of model initialisation, and that those initialisation steps that do not rely on the deployment configuration inputted by the user (initialisation of the road network, patrol beats and CFS incidents) were all performed in a first instance. Doing so saves considerable time by conducting upfront as many of the computationally expensive tasks as possible. Then, many simulations can be run in parallel in the rest of this thesis, all modelling the same time period but different deployment configurations.

Pre-processing the model environment

The ABM is designed to be generally applicable to any police force. Creating the model environment thus requires to separately load and pre-process the data sources for the chosen police force. These include the road network, the patrol beats, the CFS incidents (used to simulate reactive demand), and the historical crimes (used to simulate proactive demand).

The first step when building the model environment is to import the road network for the chosen police force using the OSMNX package, prior to initialising the model. This can be done by either specifying the name of the locality of interest or by providing a lat-long bounding box. In

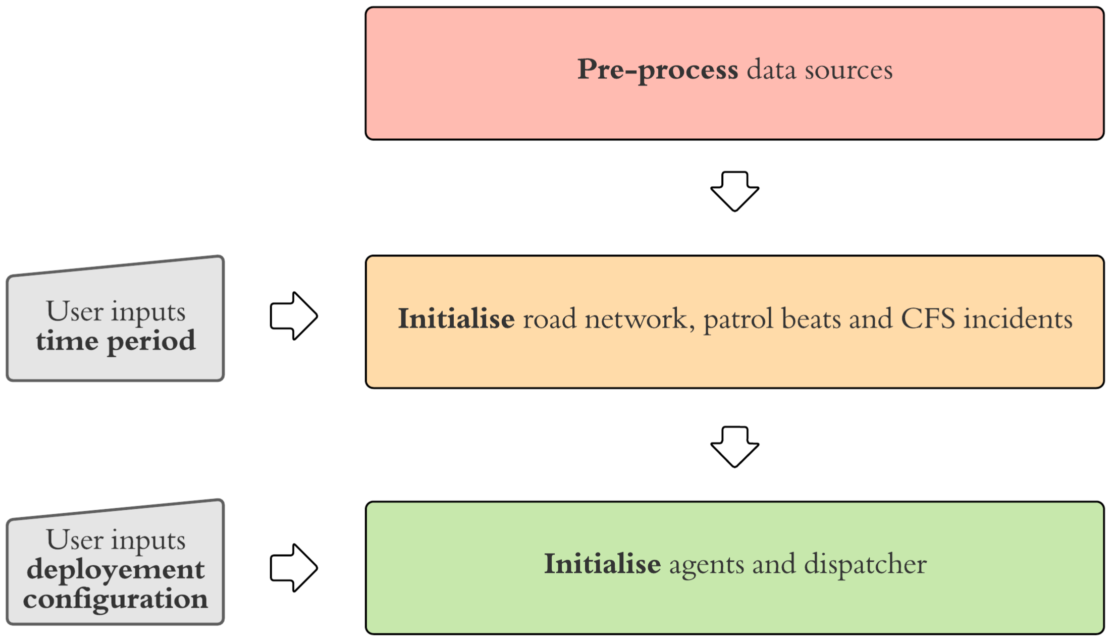

Figure 3.6: Three-step initialisation of the model and user inputs

OSMNX, the network type should be set to 'drive' so as to contain drivable public streets that preserve one-way directionality. Particularly complex road networks may result in dense graphs in which routing calculations may be computationally demanding. As such, a simplified graph is essential to ensure a low running time. When importing the road network with OSMNX, the 'simplify' option should be selected. This simplifies the topology of the imported graph by removing all nodes that are not intersections or dead-ends.

The edge dataset for the road network produced by OSMNX is pre-processed to add two columns *travel_time_mins* and *patrol_beat*. For each edge, the patrol beat within which the road segment is contained is identified using the *geometry* attribute of both the patrol beats and the graph edges. In cases where the geometry of the edge overlaps multiple patrol beats, the beat containing the centroid of the edge (i.e. the 'midpoint' of the shape) is chosen.

Another key data source when building the model environment consists in a shapefile containing the patrol beats of the police force. This dataset should contain the attributes *name*, *precinct* and *geometry* (a polygon). Additionally, the patrol beats are pre-processed to add the *centroid_node* column. This is done by identifying the closest equivalent node to the spatial centroid of each patrol beat polygon (i.e. the 'center' of the shape).

A dataset of CFS incidents simulating reactive demand over the time period is also required to initialise the model environment (see input parameters below). Each incident will be converted to an incident entity in the first step of initialisation. As such, the following columns are required in the dataset (see list of incident attributes in Table 3.4): *call_datetime*, spatial location (*latitude* and *longitude*), *precinct* and *resolution_time*. Incidents are pre-processed to add the *node* column used to embed the incidents in the road network graph. This is done by identifying the closest equivalent node to the spatial location of each incident.

Finally, the dataset of historical crimes is used in the first step of initialisation to update the *density_hist_inc* attribute for the edges of the road network (see details on how the attribute is calculated below). In pre-processing, the column *edge_index* is added to each crime, which indicates the graph edge on which the incident occurred. This is done by finding the closest graph edge to the location of the crime (using the *geometry* of the crime point).

Initialising the model environment

Once its sources have been pre-processed, the entities that make up the model environment are initialised. These entities are the road network graph, the patrol beats and the CFS incidents. This first part of the initialisation process requires that the user inputs a time period for the simulation, as illustrated in Figure 3.6. The inputed time period takes the form of 2 date-times symbolising the start and the end of the simulated time period.

This first part of the initialisation process is composed of the following steps (as summarised in Figure 3.7):

1. **Initialise the road network**: this step comes first in the initialisation process because the graph nodes and edges are required in the subsequent initialisation of other entities such as incidents, agents and patrol beats. The "calculate historical crime density" submodel (see details in Subsection 3.4.2) is executed which calculates the *density_hist_inc* on each edge of the graph.

2. **Initialise the patrol beats**: the values for *name*, *precinct* and *centroid_node* are directly taken from the shapefile of patrol beats. In the "select streets to patrol in beat" submodel (see details in Subsection 3.4.2), the 'hottest' road segments (i.e. those with the highest *density_hist_inc*) are selected to be later patrolled by idle agents during the simulation. If no dataset of historical crimes was provided by the user, the streets to patrol are selected arbitrarily in each beat (see Figure 3.7). Then, a shortest route is planned in the "plan

patrol route in beat" submodel (see details in Subsection 3.4.2) that visits all selected streets to patrol and weighs edges based on deterrence.

3. **Initialise the CFS incidents**: CFS incidents are events that will occur as the simulation runs. Each incident within the inputted time period is initialised with the following attribute values:

   - *call_datetime*, *precinct*, *time_on_scene* and *node* are directly extracted from the pre-processed dataset of CFS incidents (see pre-processing step above)
   - *status* is initialised as 'not yet occurred'
   - *dispatch_time* and *travel_time* are both set to 0 minutes. These values will be incremented automatically during the simulation once the incident has occurred, depending on the *status* of the incident (as explained in Section 3.2.4).

Initialising the dispatcher and the agents

Once the model environment – including the incidents that will occur during the simulation – are initialised, the second initialisation step takes place in which agents are initialised according to a specific deployment configuration as specified by the user. The user of the model specifies the patrol beats that will receive an agent, to a maximum of one agent per beat. A deployment configuration is represented by an array of n binary values where n is the number of patrol beats in the force. The possible values for each patrol beat are: 0 if the beat is un-staffed and 1 if the beat is staffed with an agent, as illustrated in Figure 3.8.

Agents are initialised with the corresponding *patrol_beat* and *precinct* values. Their *pos* attribute is set to the *centroid_node* of the patrol beat, and their *status* is set to 'idle'. Their *route* is an empty list and they have no assigned *incident*. Their *crime_deterrence* is set to 0. Finally, the dispatcher entity is initialised with empty lists for its *incident_queue* and *avail_agents* attributes.

3.4.2 Submodels

"Calculate historical crime density"

This submodel is executed upon initialisation of the road network (see Section 3.4.1). Its purpose is to calculate the level of historical crimes that occurred on each road segment (if a dataset of historical crimes was provided by the user) and add a *density_hist_inc* attribute to

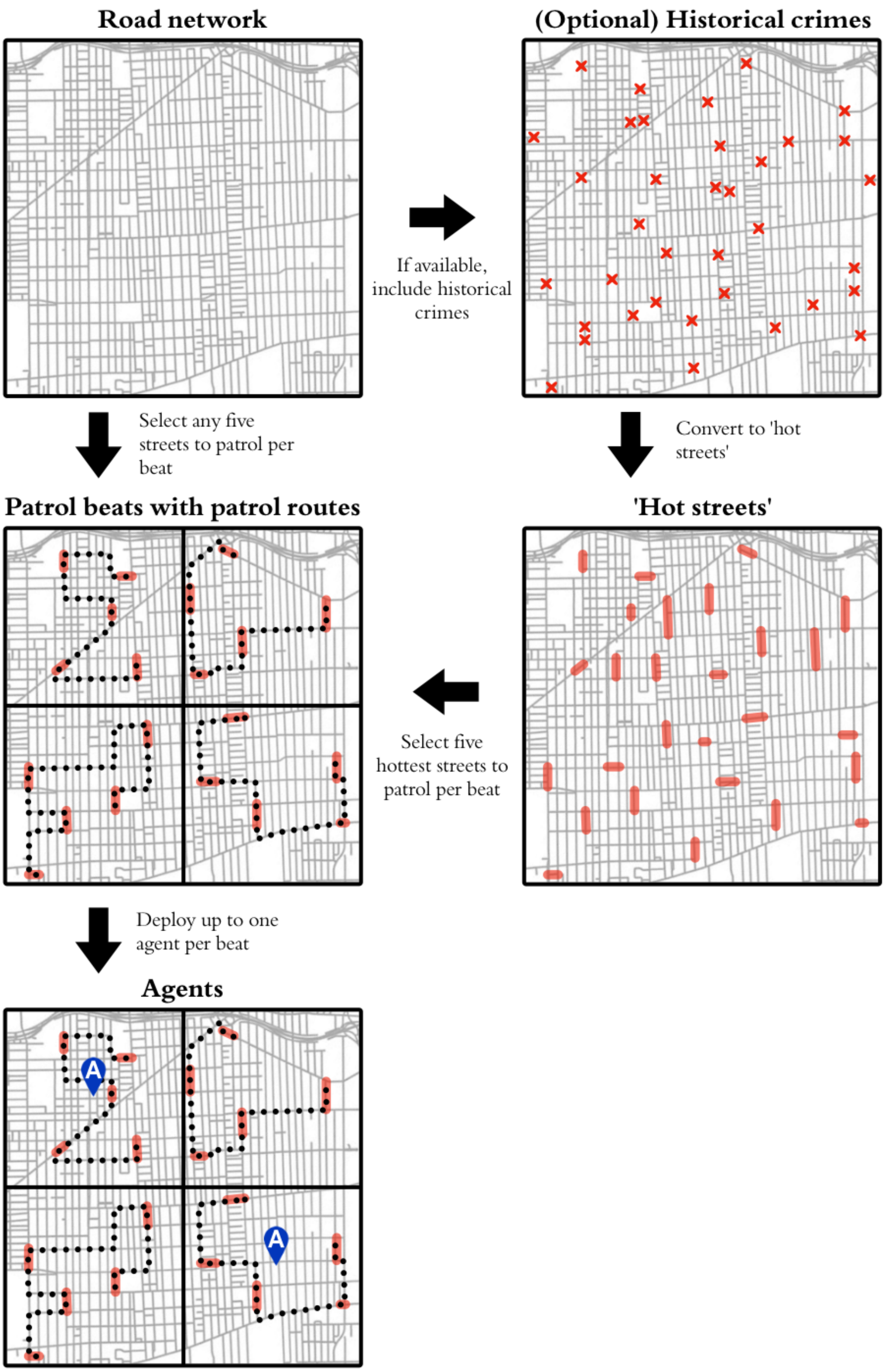

Figure 3.7: Illustration of the layers required to initialise the model environment and the agents within it

the corresponding edges of the road network. As previously mentioned, the road network is represented in the model by a NetworkX graph alongside two datasets describing its nodes and

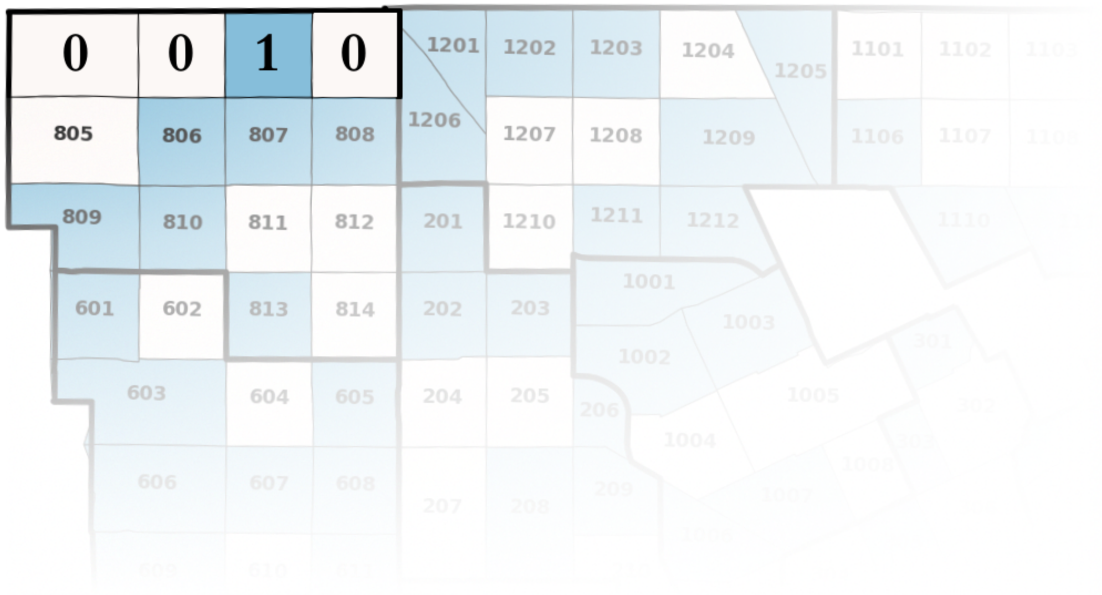

Figure 3.8: Illustration of the encoding of a deployment configuration as an array of binary values in the model. Patrol beats in blue are staffed with an agent.

edges.

The *density_hist_inc* attribute is here calculated as the ratio $\frac{n}{l}$, where: n is the number of historical crimes in the provided dataset that occurred on the road segment and l is the length of the segment (in meters). Dividing by the segment length l accounts for the fact longer road segments are intrinsically more likely to feature more crimes.

The number of historical crimes n on each edge is calculated by summing those crimes that occurred in the dataset on that edge on similar time periods to that inputted by the user (e.g. all weekdays 08:00 to 16:00 when the inputted time period is Monday 08:00 to 16:00). This calculation is performed using the *edge_index* column – the index of the edge on which the crime took place – which was added to the historical crime dataset in pre-processing. The decision to group similar time periods together based on the demand they experience will be detailed in Chapter 5. It is expected that, with a crime dataset spanning at least a year's time, there should be enough historical crimes having occurred during similar time periods. However, in cases where fewer than 500 historical crimes are found on similar time periods in the provided dataset, the edge attribute *density_hist_inc* is instead calculated by counting crimes across the entire dataset, regardless of time periods. If the user does not provide any historical crime dataset, the *density_hist_inc* attribute of all graph edges is set to zero.

The newly calculated *density_hist_inc* attribute is subsequently used in two places in the model. First, it is used in the initialisation of patrol beats when selecting a set of streets to patrol based on historical demand (see submodel "select streets to patrol in beat" below). Second, it is used in the simulation itself by agents who resume patrolling after responding to an incident. These agents calculate the route back to their patrol beat that produces the most crime deterrence using the *density_hist_inc* attribute of the graph edges (see "Patrol" submodel below).

"Select streets to patrol in beat"

This submodel is executed when initialising each patrol beat. The streets to be patrolled by idle agents are selected amongst all graph edges in the beat. As mentioned in Chapter 2, there are many sophisticated techniques to identify crime hotspots in the literature (see Chainey et al., 2008 for a review). However, in the version of the model presented here, 'hot' road segments are identified by simply choosing the 5 segments with the highest historical crime density (*density_hist_inc*). The number of road segments selected may be changed in future versions of the model. If no edge in the beat has a *density_hist_inc* greater than 0 (either because the user did not provide a dataset of historical crimes or because no historical crime took place in this specific beat), 5 road segments are selected arbitrarily. This is done using a random state seed so as to prevent introducing stochasticity in the model. The reader may refer back to Figure 3.7 for an illustration of the process of selecting the streets to patrol in each beat.

"Plan patrol route in beat"

In policing operations, a high-level strategy of hotspot patrolling needs to be turned into detailed patrol routes. This submodel is executed when initialising each patrol beat, after first executing the "select streets to patrol in beat" submodel (see above). Once the road segments to patrol in each beat are selected in the model initialisation, a patrol route is planned that visits all the pre-defined segments to patrol in the beat and weighs edges based on deterrence.

Finding the optimal route between graph nodes is a NP-complete problem known as the travelling salesman problem. There have been different approaches in the literature in designing police patrol routes (see for instance Chawathe, 2007; Chen et al., 2015). In an attempt to keep computing time as low as possible as part of this initial proof of concept, a simple routing solution is implemented following the procedure detailed in Figure 3.9. Starting from the centroid node of the patrol beat, the patrol route is built incrementally by repeatedly finding to the next closest road segment until all streets to patrol have been visited exactly once. Calculating this patrol route ahead of running the model (as opposed to every time an agent resumes its idle behaviour) saves considerable computational time, as routing calculations in dense graphs are computationally expensive.

"Distribute incidents"

This submodel is executed by the dispatcher at each model step. Incidents automatically enter the dispatcher's queue of unallocated incidents (*incident_queue*) when the call takes place (at incident's *call_datetime*). If two incidents occur at exactly the same time, they enter the queue in an arbitrary order (according to their order of occurrence in the CFS dataset). Importantly, incidents remain in the queue until an agent reaches the scene. This means that incidents with an 'allocated unattended' *status* continue to be considered for allocation in this submodel, in case a newly available agent find themselves closer to the incident compared with the already-dispatched agent. Further details about this process of on-the-fly re-dispatching are provided in sub-submodel "find nearest available agent" below.

Keeping the response time to incidents low is a priority for police agencies. In the model, the dispatcher thus seeks to minimise incident response time when distributing incidents to available agents. More specifically, the dispatcher aims to reduce both components of incident response time: namely *dispatch_time* and *travel_time*.

In order to keep incident *dispatch_time* low, and given that all incidents in the queue have the same priority (emergency), the dispatcher processes the incidents in a First In First Out (FIFO) fashion. Incidents are thus processed by the dispatcher in order of occurrence.

Once an incident has been chosen for processing from *incident_queue*, the next step for the dispatcher is to identify the available agents in the precinct of the incident as part of the intra-sector dispatching approach chosen here. If no agents are available in the precinct, no dispatch is made and the incident remains in the queue. If some agents are available in the precinct, the dispatcher executes its "find nearest available agent" sub-submodel, in which it seeks to find the agent whose driving time to the incident is minimal. Figure 3.10 illustrates

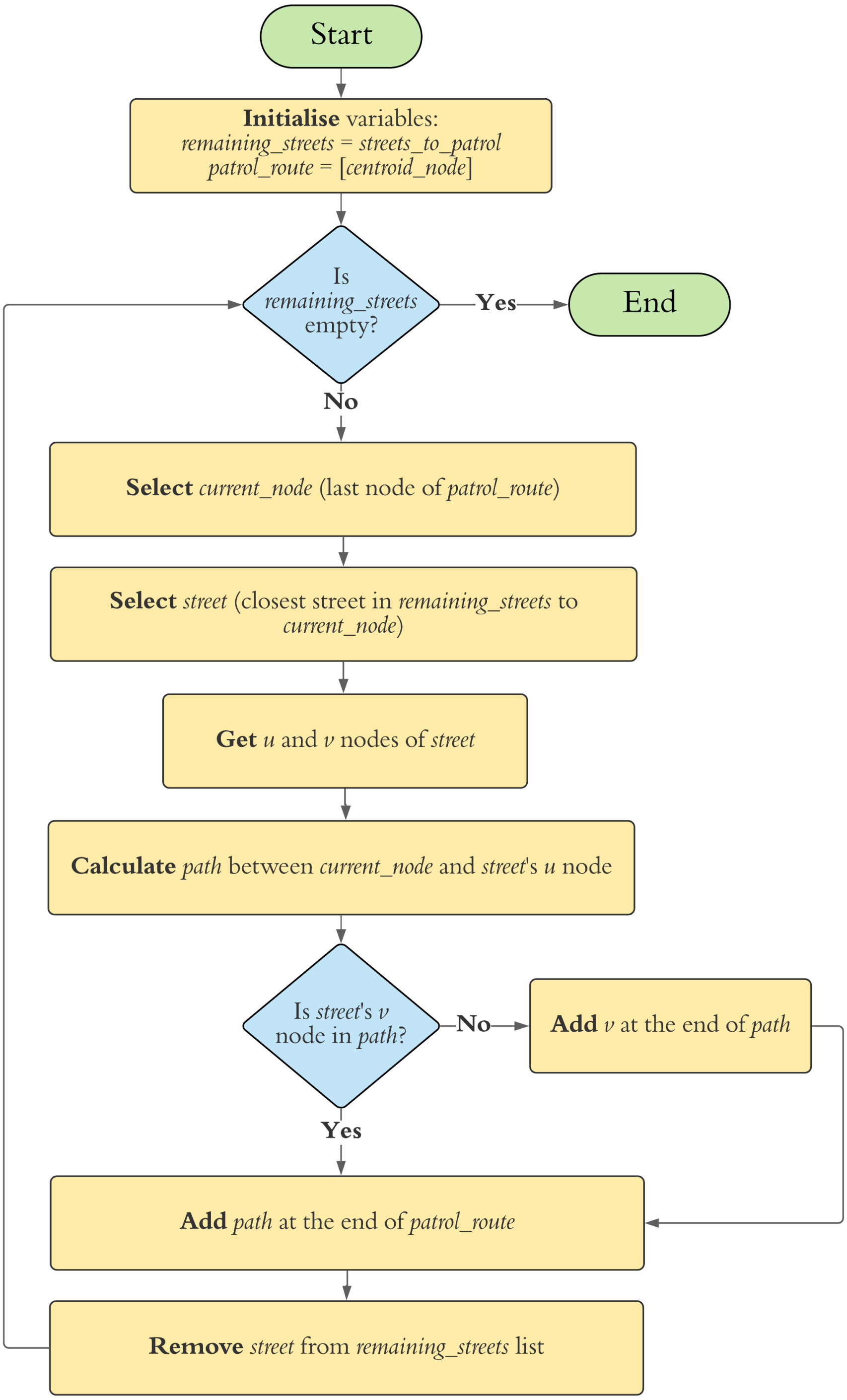

Figure 3.9: Flow diagram explaining the algorithm used in the "plan patrol route in beat" submodel executed upon model initialisation.

the way in which the dispatcher distributes unattended incidents to available agents. Finally, the dispatcher dispatches the chosen closest available agent by executing the "dispatch agent to incident" sub-submodel (see details below).

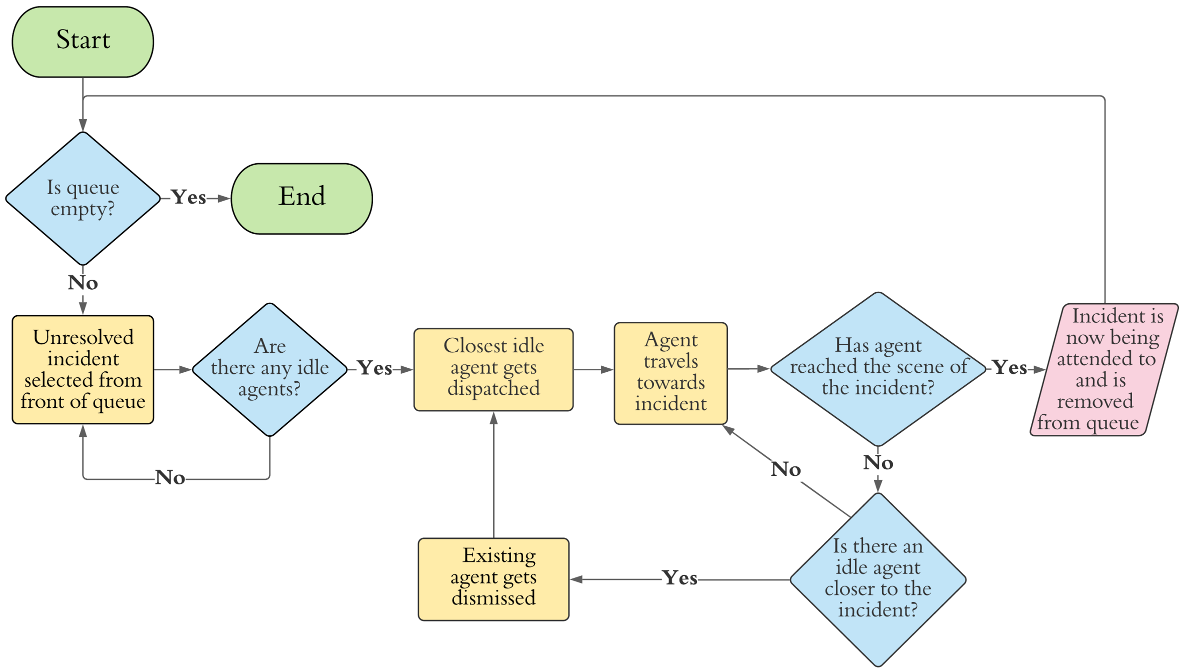

Figure 3.10: Flow diagram illustrating the dispatcher's "distribute incidents" submodel executed at each model step.

"Find nearest available agent (sub-submodel)"

This sub-submodel is executed by the dispatcher for each incident processed in the "distribute incident" submodel (see details above). Its aim is for the dispatcher to select the nearest available agent to dispatch to a given incident with the view to minimising the incident *travel_time*.

In order to identify the nearest agent, the dispatcher first calculates the fastest route to the incident for each candidate agent in the precinct. This fastest route is one that minimises the *travel_time_mins* attribute for each edge along the route. The route takes into consideration the road network infrastructure (one-way roads etc.) in order to simulate realistic travel itineraries for the agents.

The dispatcher then estimates the time it would take each candidate agent along their fastest route to reach the incident. This estimated driving time for a candidate agent is calculated as $\sum_{i=1}^{n} t_i$ where t_i is the *travel_time_mins* for the i^{th} edge of the route and n is the number of edges on the route.

Finally, after estimating the driving time to the incident for all candidate agents, the dispatcher selects the agent with the smallest driving time. If the incident has no agent already assigned to it, then the dispatcher executes its "dispatch agent" submodel in which the selected agent is dispatched to the incident. Figure 3.10A illustrates the process of finding the nearest available agent by the dispatcher.

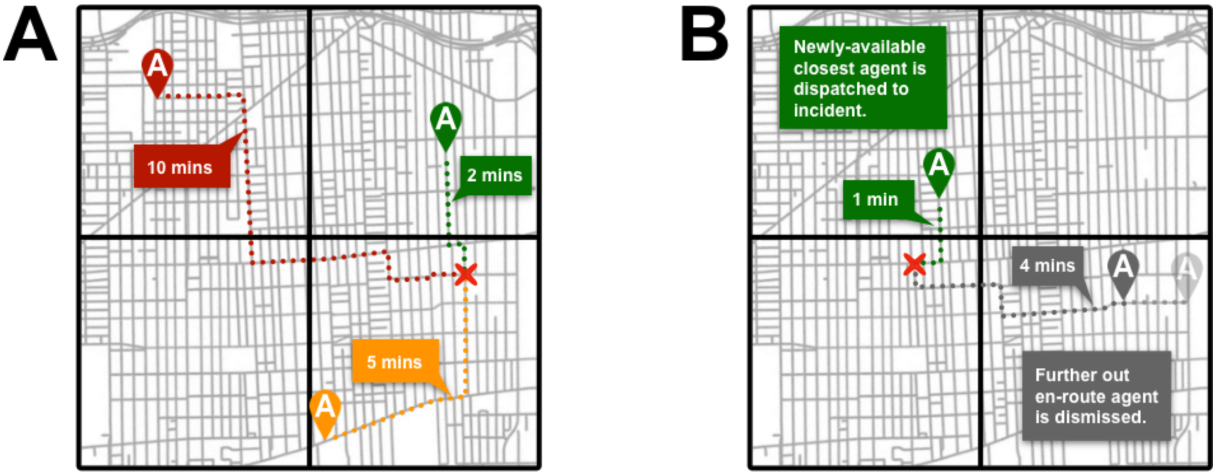

Figure 3.11: Illustrations of the dispatcher's "find nearest available agent" submodel. A: The dispatcher compares the fastest routes of all available agents in the precinct. B: the dispatcher re-dispatches a newly available agent in the precinct to replace the existing en-route agent.

When calculating the agents' fastest routes and the corresponding driving times, the dispatcher assumes that the agents are able to drive at the speed limit allowed on each road segment. In the real world, the speed of responding vehicles may be slower than the speed limit due to a range of environmental factors such as traffic or weather conditions. Conversely, police responding vehicles may also drive faster than the speed limit – providing it is safe to do so – by deploying blue lights and sirens. As such, it seems a reasonable modelling compromise to assume that responding units dealing with emergencies are able to travel on each road segment at the allowed speed limit.

"Dispatch agent to incident (sub-submodel)"

This sub-submodel is executed by the dispatcher for each agent chosen to be dispatched to an incident through the "find nearest available agent" sub-submodel (see details above). Having selected an agent for dispatch, the dispatcher updates the incident's *agent* attribute with that agent's entity. Additionally, the following attributes of the chosen agent are updated by the dispatcher:

- Agent's *status* is changed to 'travelling', which means they become unavailable for further dispatching.
- Agent's *incident* takes up the incident entity

- Agent's *route* takes up the fastest route that was calculated by the dispatcher upon executing the "find nearest available agent" submodel. As previously mentioned, routing calculations bear a computational cost in the model that may increase considerably with the size of the road network. As such, given that the dispatcher already pre-calculates the fastest route for each agent, it is preferable for the dispatcher to pass on that route to the chosen agent instead of the agent re-calculating the fastest route at each step of the simulation.

"Move"

Agents execute the "move" submodel when they are either travelling to an incident (*status* is 'travelling') or patrolling (*status* is 'idle'). In this submodel, agents traverse the road network by moving from node to node along their *route* attribute, which is made of a list of connected graph nodes.

As previously stated, an agent begins each model step with a time allowance equal to the model step time (e.g. one minute). At each model step, moving agents evaluate the furthest node they can reach on their *route* within the model time step, based on the *drive_time_mins* attribute of the graph edges. Once again, this assumes that agents drive at the maximum speed limit allowed on each road segment (see discussion in the "find nearest available agent" sub-submodel). At the end of each step, their *pos* attribute is updated to the furthest node reached during this model step and the nodes that were visited during this time step are removed from their *route*.

In order to simulate realistic driving times, agents in the model 'memorise' the portion of their time allowance that is spare at the end of each time step and add it to their time allowance at the next step. Figure 3.12A illustrates the movement of agents along a hypothetical route, with a one-minute time allowance at each step. This approach also accounts for particularly long edges where the driving time for the edge exceeds the step time (e.g. an edge with a driving time of 2 minutes when the step time is 1 minute) as illustrated in Figure 3.12B. With this approach, agents remain on the same node for the necessary number of simulation steps until they can reach the other side of the graph edge. For instance, an agent would be able to travel across an edge which requires 2 minutes of driving time after 2 one-minute time steps.

Due to the nature of the model environment, the road network is truncated around the periphery of the police force. As a result, segments of one-way roads situated near the periphery are

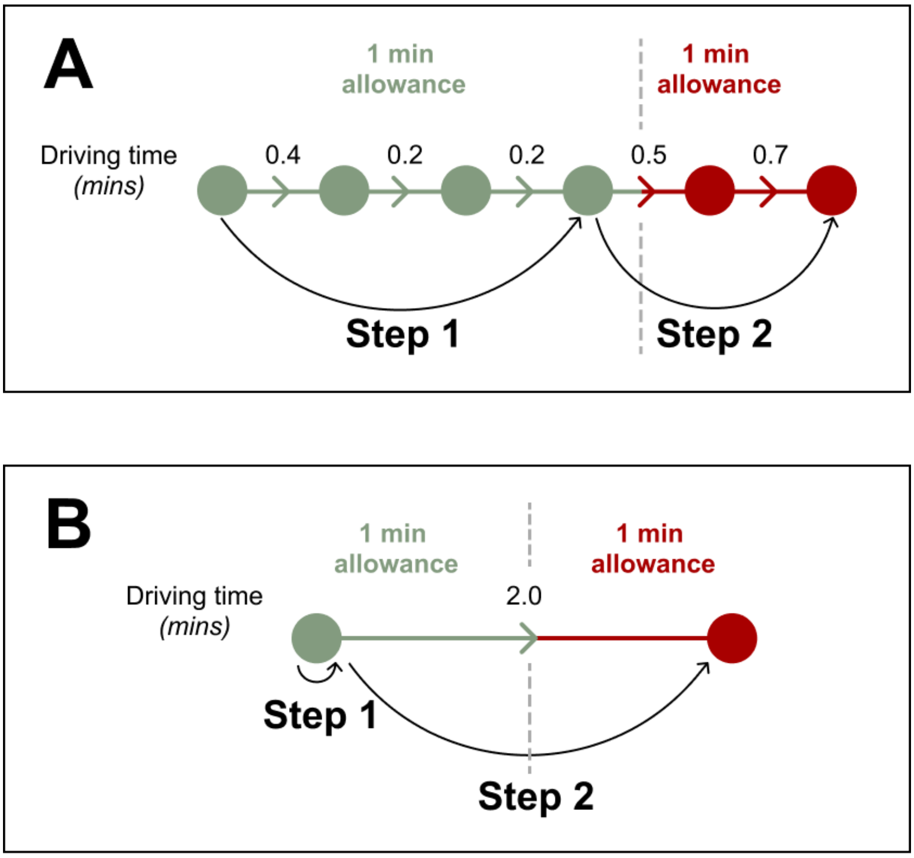

Figure 3.12: Illustration of 2 hypothetical scenarios of the movement of an agent along their *route*. The agent starts each step with a one-minute time allowance. A: Agent travels as far as their time allowance permits and uses their spare time from step 1 to reach the final node in step 2. B: Agent cannot travel across the long edge in one model step. They remain on the same node for step 1 in order to reach the next node in step 2.

either not reachable, or agents are unable to leave them. To circumvent the issue, agents are temporarily allowed to drive up one-way roads until they are able to either reach their destination or regain access to the main road network. Although not possible in the real world, this behaviour remains a rare occurrence and should thus not affect the overall pattern of response time.

When an agent reaches the scene of their assigned incident within a given time step, their *status* is updated to 'at_scene'. They may then proceed to executing the "stay at scene" submodel (see below) in the same step, provided there is spare time in their step time allowance.

"Stay at scene"

This submodel is executed by all agents whose *status* is 'at_scene'. Those are agents that have reached the scene of their assigned incident and are tending to it. Agents remain on the *node* corresponding to the scene of their incident until the incident attribute *time_being_tended* becomes equal to the incident attribute *resolution_time*.

Figure 3.13 provides a flowchart to illustrate the "stay at scene" submodel process from the moment the agent reaches the scene to the incident being resolved. At each model step, agents check whether the incident will be resolved within the time step (i.e. an agent has been tending to it for the duration of *resolution_time*). If so, their *status* is updated to 'idle' and they may then proceed to executing the "patrol" submodel (see below) in the same step, providing there is spare time in their step time allowance. These agents will be available for dispatch at the beginning of the next time step, when the dispatcher executes its "find nearest available agent" sub-submodel. The *status* of resolved incidents is updated to 'resolved'. Conversely, if the incident cannot be resolved within this time step, the agent remains on the incident's *node* for another step.

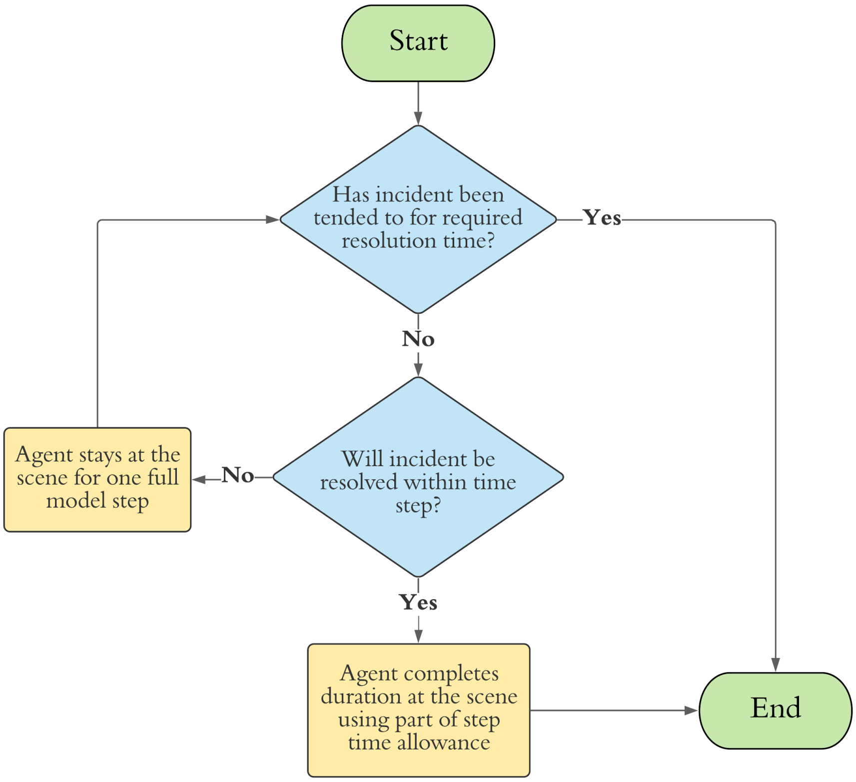

Figure 3.13: Flow diagram of the "stay at scene" submodel from the moment an agent arrives at the scene to the moment the incident is resolved.

"Patrol"

When agents are not responding to an incident, their *status* is 'idle'. As previously explained in Section 3.3, idle agents in the model engage in a patrolling behaviour in order to simulate patrol vehicles attempts to deter crime. Implementing this behaviour in the model allows for the measurement of the potential gain in crime prevention that could be brought by deploying agents to particular patrol beats.

In the model, each 'idle' agent is aware of the *patrol_route* of their designated patrol beat (see submodel "plan patrol route in beat" for details on how the patrol routes are designed). Upon becoming idle, the agent first calculates the shortest path (with edges weighted by crime deterrence using the edge attribute *density_hist_inc*) from their current location to the start of the patrol route of their assigned beat. The agent's *route* is then updated to be a concatenation of (1) the calculated path to the start of the patrol route and (2) the *patrol_route* itself (see Figure 3.14 for illustration).

Figure 3.14: Agent routing to the start of their assigned *patrol_route*

Once their *route* attribute is defined, agents execute the "move" submodel (see details above) in which they move along their route until they reach the last node. When the agent reaches the last node of the *patrol_route*, they have completed a patrol 'round'. They may then start a new round, providing they have spare time on their step time allowance. Similarly to their initial routing to the start of patrol route (as shown in Figure 3.14), agents calculate a path from their current position (i.e. the last node of the route) to the start of the patrol route, as illustrated in Figure 3.15.

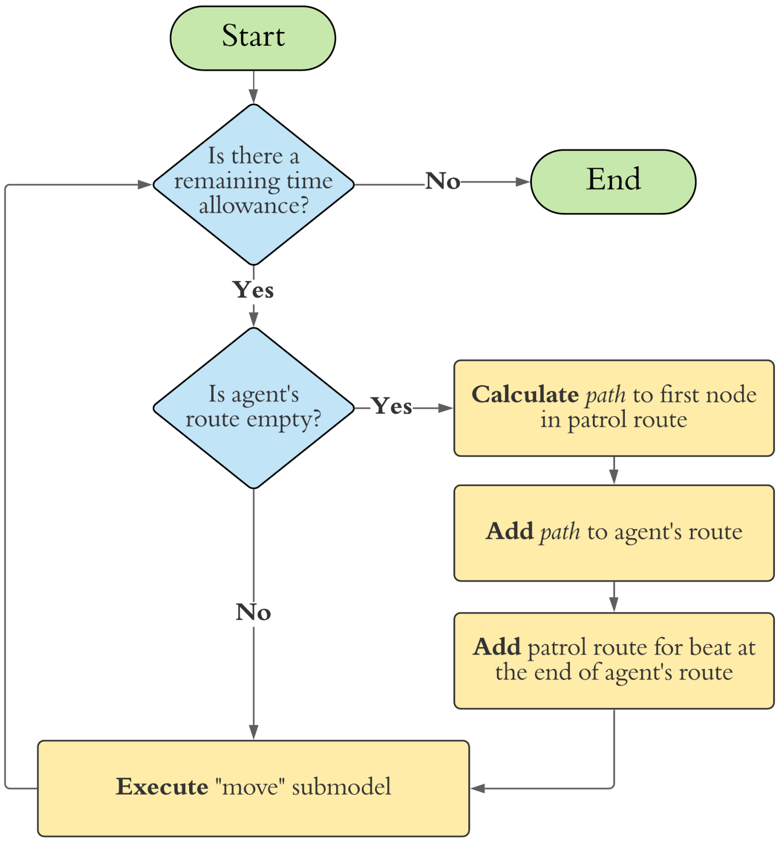

Figure 3.15: Flow diagram illustrating the "patrol" submodel executed by idle agents at each step. Agents start each step with a one-minute time allowance.

3.5 Conclusion

This chapter has specified in details the ABM built in this thesis following the ODD protocol. This ABM was designed to be applied to any police force context, and as such, the description of the model remained generic. The ABM simulates a fleet of motorised patrol units performing various activities throughout their shift, including responding to CFS incidents and patrolling to deter crime. The outcome of the model includes a number of performance metric such as average response time, percentage of 'failed' responses and total crime deterrence score.

The purpose of the ABM is to evaluate various deployment configurations with the view to identifying optimal solutions to the PDOP. In order to ensure that the model is fit for this purpose, it needs to undergo a process of validation which compares the outcomes of the model against equivalent outcomes observed in the real police system emulated by the model. The next chapter provides a validation of the model along with a series of model experiments using the case study of Detroit, Michigan (US).

Chapter 4

Analysis of Detroit Police Department's data

4.1 Introduction

The previous chapter has provided a detailed description of the ABM developed in this thesis. The aim with this ABM is to build a portable modelling platform capable of rapidly simulating patrol activities and dispatch for different police forces. In this thesis, the ABM is applied to the exemplar police force of Detroit Police Department (DPD) for the purpose of illustration and model validation against real world data (see Chapter 5 for model validation).

This chapter introduces the police force of DPD and provides visualisations of its supply and demand levels. In Section 4.2, the city of Detroit is briefly introduced along with its police force. Section 4.3 presents the data sources from Detroit that are used in the ABM and the various pre-processing steps that were performed on the data. Then, spatial and temporal visualisations of the reactive demand from the CFS data (Section 4.4) and of the proactive demand from the reported crime data (Section 4.5) are provided. Finally, Section 4.6 provides visualisations of DPD's existing reactive effectiveness looking at incident response times in the CFS dataset.

4.2 Description of the force

4.2.1 The city of Detroit

Geography

Figure 4.1: The city of Detroit and the two enclosed cities of Hamtramck and Highland Park.

The city of Detroit (Michigan, US) has an area of about 143 square miles (370 km^2) and in 2019, its total population was estimated at 670,031. As shown in Figure 4.1, the city surrounds the two enclave cities of Highland Park and Hamtramck.

History

The relocation of the automobile industry that had been the reason behind Detroit's rapid boom in the 1920s caused a significant population decline and urban decay. This led to a rise in unemployment, poverty and crime, building up to Detroit filing of a municipal bankruptcy case in 2013 – the largest case in US history.

Since its bankruptcy case, Detroit has undergone significant transformation, however these have been mostly directed towards the gentrification of the downtown area. This has led to an increasing divide between Detroit's downtown area and its neighbourhoods, an issue commonly referred to as the Two Detroits.

Crime

Known in the 1970s and 1980s as the "murder capital of America", Detroit has struggled with high crime for decades. In 2019, Detroit's violent crime rate was 1,965.3 per 100,000 (Federal

Bureau of Investigation - Uniform Crime Reporting (UCR) Program, 2019); one of the highest in the United States. While the crime rate remains very high in most of the city, its downtown area has seen a significant decline in crime in recent years, exhibiting even lower crime than national and state averages (Metzger and Booza, 2005).

4.2.2 The police force

The city of Detroit is serviced by Detroit Police Department (DPD). The force is divided into 11 precincts (see Figure 4.2), each subdivided into 131 patrol beats (called scout car areas), as shown in Figure 4.3. Scout car areas are subdivisions of precincts to which officers can be assigned for patrol (see Chapter 2).

Like most Western police agencies, calls are centrally received and, since 2013, dispatching decisions are aided by a Call And Dispatch (CAD) system (see Chapter 2 for details on CAD). Private security forces such as Downtown PD or Wayne State University PD are also in service in precinct 3. These private forces patrol areas around downtown and midtown, alleviating the demand faced by the precinct.

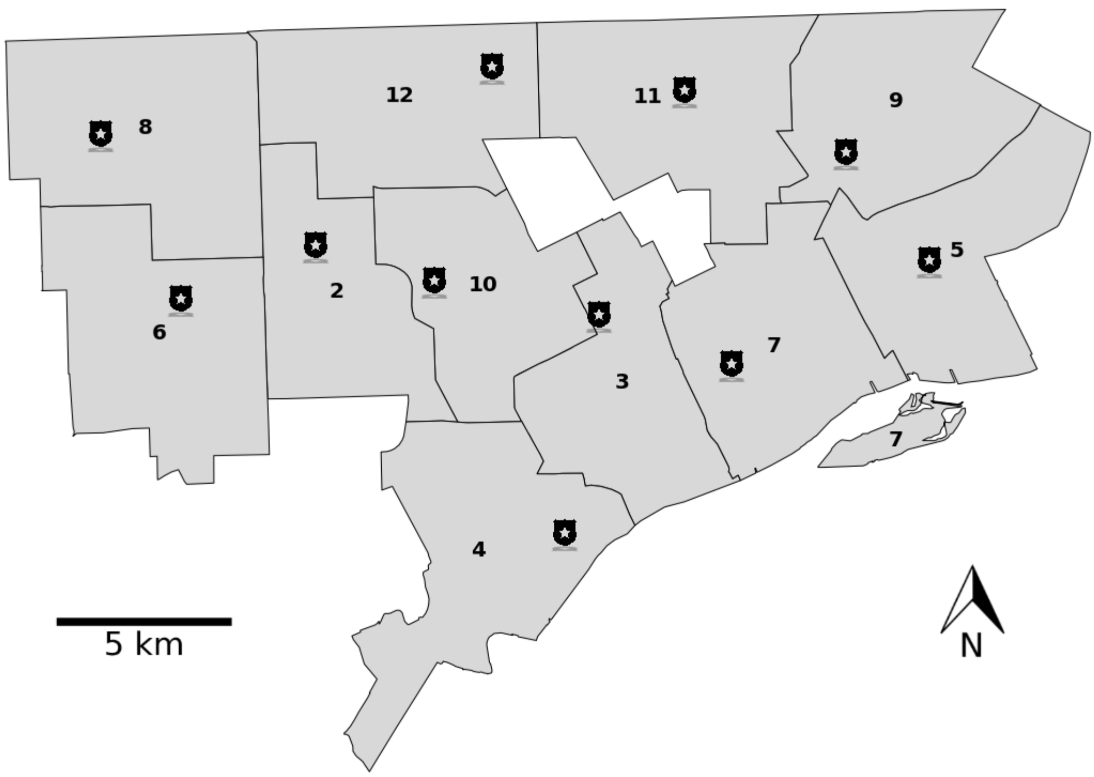

Figure 4.2: DPD's precincts and police stations

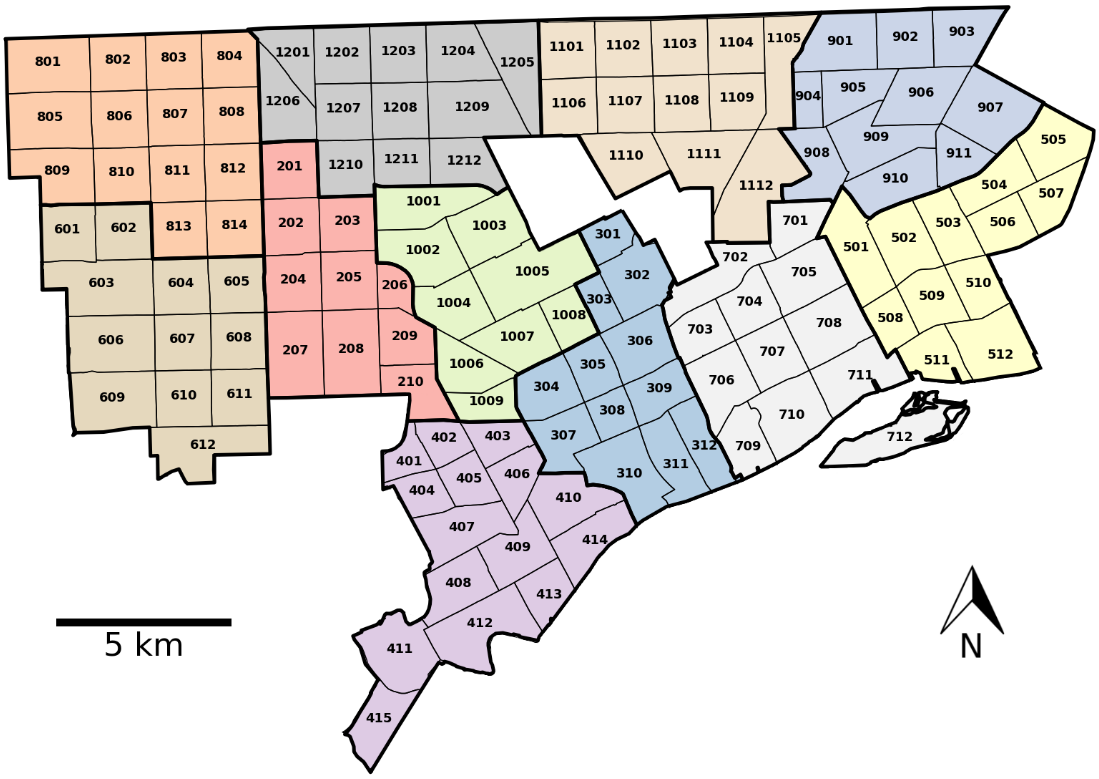

Figure 4.3: DPD's scout car areas

Detroit has lost nearly half its patrol officers since 2000 as officers retired or left for other police departments amid the city's bankruptcy and cuts to pay and benefits. Currently, there are 448 citizens for every officer, as opposed to in the 1970s, when the citizen-to-officer ratio was around 380 (Hunter, 2015). This emphasises the need to deploy available patrols in a cost-effective manner; a problem that is a the core of this research.

4.3 Data pre-processing

4.3.1 The data sources

The city of Detroit (Michigan, US) was chosen as it is one of the few worldwide cities for which all data sources needed to build the model are publicly available online. Having provided an overview of the city of Detroit and its police force, the following section outlines the various datasets used to run and validate of the model for the case study police force.

Road network

The road network for the city of Detroit was downloaded using the python library OSMNX. As previously detailed in Chapter 3, this produces a directed graph comprised of nodes (representing network intersections) and edges (representing road segments). An edge is a line or curve segment connecting two nodes at its ends. In the case of Detroit, the acquired network graph is composed of 59,696 edges and 20,719 nodes. The road network in Detroit is tightly connected, with few long edges between intersections. Indeed, only 0.3% of the edges (131 of them) are so long as to necessitate a driving time longer than 1 minute. Given that the step time in the model built in this thesis is of 1 minute, this means that most nodes in Detroit can be reached from an adjacent node within one model step.

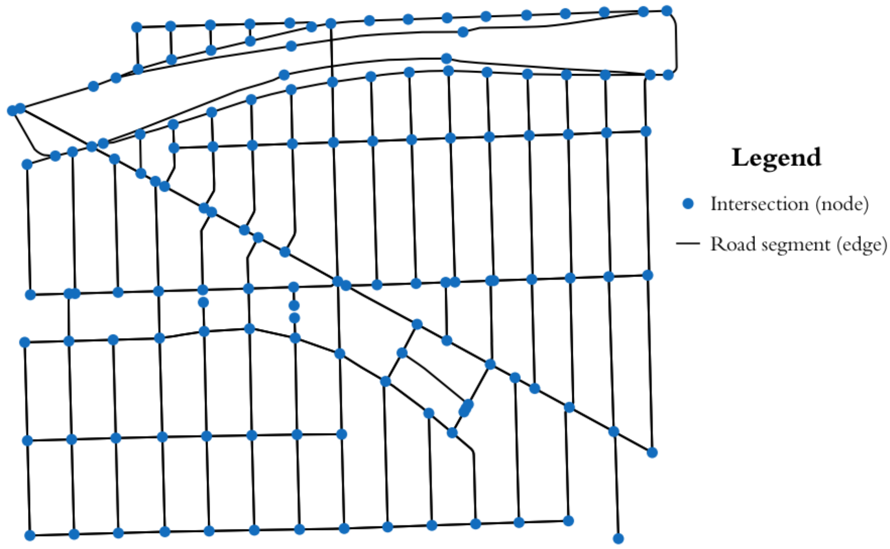

Figure 4.4: A portion of the street network in Detroit, illustrating nodes and edges

Patrol beats

As discussed previously, patrol beats (called scout car areas in Detroit) are used to delineate individual patrolling areas for patrol vehicles. The shapefile representing the boundaries of Detroit's patrol beats was drawn from the City of Detroit Open Portal (https://data.detroitmi.gov/datasets/dpd-scout-car-areas). For each patrol beat, the shapefile contains all the necessary information to describe patrol beats in the ABM (see Chapter 3), in particular its geometry, its name as well as the precinct of which it is a subdivision.

CFS incidents

To simulate reactive demand, the model requires a dataset of historical CFS. A historical CFS dataset spanning a three-year period (2017-2019) was obtained from the City of Detroit Open Data Portal (https://data.detroitmi.gov/datasets/911-calls-for-service). The data contains information about each incident needed to run the ABM, in particular its call datetime, spatial location, priority, as well as dispatch time, travel time and time responders spent at the scene.

The incident's spatial coordinates (lat-lon) had been spatially perturbed by DPD prior to making the dataset public. This process consisted in moving the location of each incident to the nearest road intersection (e.g. 'Promenade Ave & Roseberry St'). This process of perturbation allows for the confidentiality of offenders and victims to be preserved while sharing the data with researchers.

According to the OSM data, the average (median) street in Detroit is 100 meter long. This indicates that Detroit – alike many other US cities – features a well connected grid-like street network and as such, the perturbed locations offer a reasonable proxy for real incident locations. Prior to running the model, the spatially perturbed incident's coordinates were translated into their nearest graph node in order for routing calculation to be performed in the simulation. As the spatial perturbation process had already converted incident locations to the nearest road intersection, the nearest node here corresponds to the node representing that intersection.

Reported crimes

In order to evaluate the crime deterrence score (as a result of agent patrolling) for a given deployment configuration, the model required a dataset of historical reported crimes. Such a dataset was obtained from the City of Detroit Open Data Portal (https://data.detroitmi.gov/datasets/rms-crime-incidents) spanning the same three-year period (2017-2019) as the historical CFS dataset. For each crime, the data contains the date at which the crime was reported and the spatial coordinates at which it occurred. Much like the CFS incident dataset, the crime locations had been spatially perturbed by DPD. As such, these perturbed locations were translated into their nearest graph node in a pre-processing step.

4.3.2 Data preparation

CFS incidents

Since the ABM is concerned with modelling the dispatching of patrol vehicles to emergency incidents specifically, those incidents with a priority code value different from 1 (emergency) were excluded from the dataset. Between January 1, 2017 and December 31, 2019, DPD recorded approximately 238,692 emergency calls (priority 1), representing 9.8% of the CFS for the period (see Table 4.1).

Incident priority	Number of incidents	% of incidents
1	238,692	9.8
2	742,085	30.4
3	1,305,943	53.4
4	130,978	5.4
5	24,746	1.0
P	395	0.01
None	1,182	0.05

Table 4.1: Number and percentage of incidents for each priority type in the un-processed CFS dataset

Furthermore, additional filtering steps were performed in order to remove incidents from the dataset according to the following criteria:

- The incident location coordinates, dispatch time, travel time or time at the scene were missing (14,547 incidents removed).
- The incident took place outside of Detroit's boundaries, or was not resolved by DPD but by Detroit's fire department or by the police department of nearby cities Highland Park or Hamtramck (2,346 incidents removed).
- The incident was initiated by an officer or the call duration was zero minutes (43,741 incidents removed).
- No dispatch was made; i.e. the travel time was zero (12,333 incidents removed). This happens if the incident is resolved during the call or if the call was a hoax for example.
- Response time exceeded 100 minutes (341 incidents in the three-year period). These incidents are likely to be recording mistakes where, for instance, an officer forgot to indicate the incident had been resolved. These outliers have the potential to skew the validation of the model as this type of behaviour is impossible to reproduce in the model.

The final processed dataset contained 165,384 emergency incidents (January 1st, 2017 to December 31st, 2019), down from 238,692 emergency incidents in the initial dataset. This corresponds to an average of 151 daily emergency calls throughout the force or 13.7 daily incidents per precinct.

The dataset came with a *scout_car_area* column composed of a concatenation of the precinct number and the scout car area number. This column had been entered by DPD as free text and as such, trailing white spaces had to be removed. Then, the column was split into two separate columns (*precinct* and *patrol_beat*) for analysis. In addition, using the road network imported in OSMNX (see above), a column was added that contains the closest node to the location of each incident in the dataset.

Reported crimes

Similarly to the CFS dataset the reported crime dataset underwent a process of filtering to remove those crimes that took place outside of the boundaries of Detroit. This resulted in a dataset of 247,018 reported crimes for the three-year time period (January 1st, 2017 to December 31st, 2019).

Reported crimes are used in the model to estimate the crime deterrence score for each agent throughout the simulation, which are summed into a total crime deterrence score for the chosen deployment configuration. In its initialisation step, the model thus needs to count the number of historical crimes that took place on each street segment (graph edge). This is usually achieved by first identifying the graph edge on which each crime took place (*edge_index* column). Then, during the initialisation of the ABM, the density of historical crimes (*density_hist_inc*) on each graph edge is calculated. This is typically done using the $\frac{n}{l}$ formula, where n is the number of historical crimes that occurred on the edge on similar past time periods (e.g. other Monday night shifts), and l is the edge's length.

However, as previously mentioned, the crime locations for Detroit have been spatially perturbed and thus point to the nearest road intersection to the original location of the incident. This road intersection is converted to its equivalent graph node in a pre-processing step. The number of historical crimes is thus provided per node, instead of the required number per edge. To circumvent the issue and still obtain estimated crime density values for the graph edges, the historical crimes on each node were distributed to adjacent edges in proportion to their length.

A *density_hist_inc* value is thus calculated for each edge according to the following formula:

$$\textit{density\_hist\_inc} = \text{Num of crimes on node} \times \frac{\text{Edge length}}{\text{Total length for node's adjacent edges}}$$

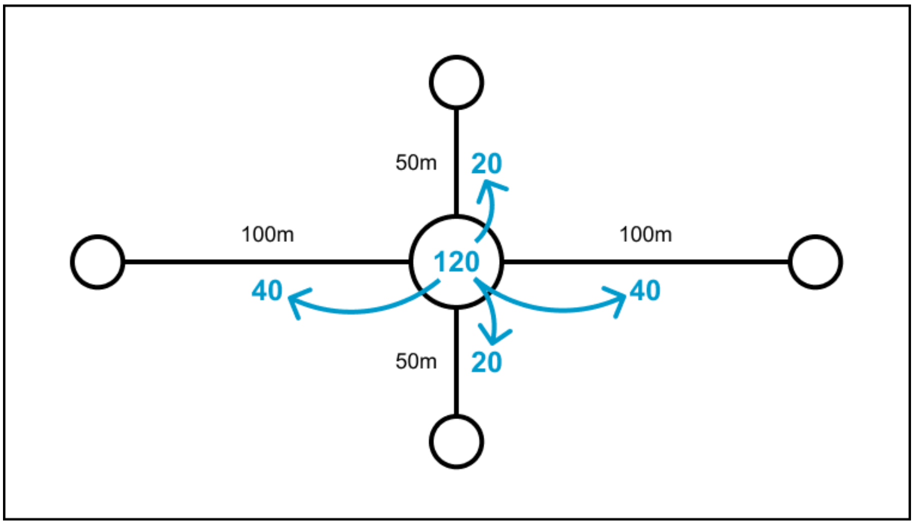

Figure 4.5: Diagram explaining the process of distributing historical crime counts from nodes to adjacent edges.

To illustrate the process by which node-level crime counts are distributed to edges of the graph, consider a node on which 120 historical crimes occurred, and with 4 adjacent edges, two of which are of size 100 meters and 2 of size 50 meters (as shown in Figure 4.5). Through this process, the edges of size 100 meters will get a *density_hist_inc* value of 33 (a third of the 100 crimes on the node) while those of size 50 m will get a value of 16.5 (a sixth of the 100 crimes on the node). This process relies on the assumption that all adjacent edges to a given node experience an equal number of crimes per meter. Longer edges inherit a higher proportion of the crime incidents that took place on the node they are adjacent to than shorter ones.

The calculated graph attribute *density_hist_inc* is used in the initialisation of the model to identify which road segments to patrol and plan patrol routes accordingly (see model ODD in Chapter 3). In each patrol beat, the 5 graph edges with the highest attribute *density_hist_inc* are selected to be patrolled. To illustrate, Figure 4.6 shows the selected streets for an example eight-hour shift in Detroit (16:00 to midnight on Monday, 3rd Sept 2018).

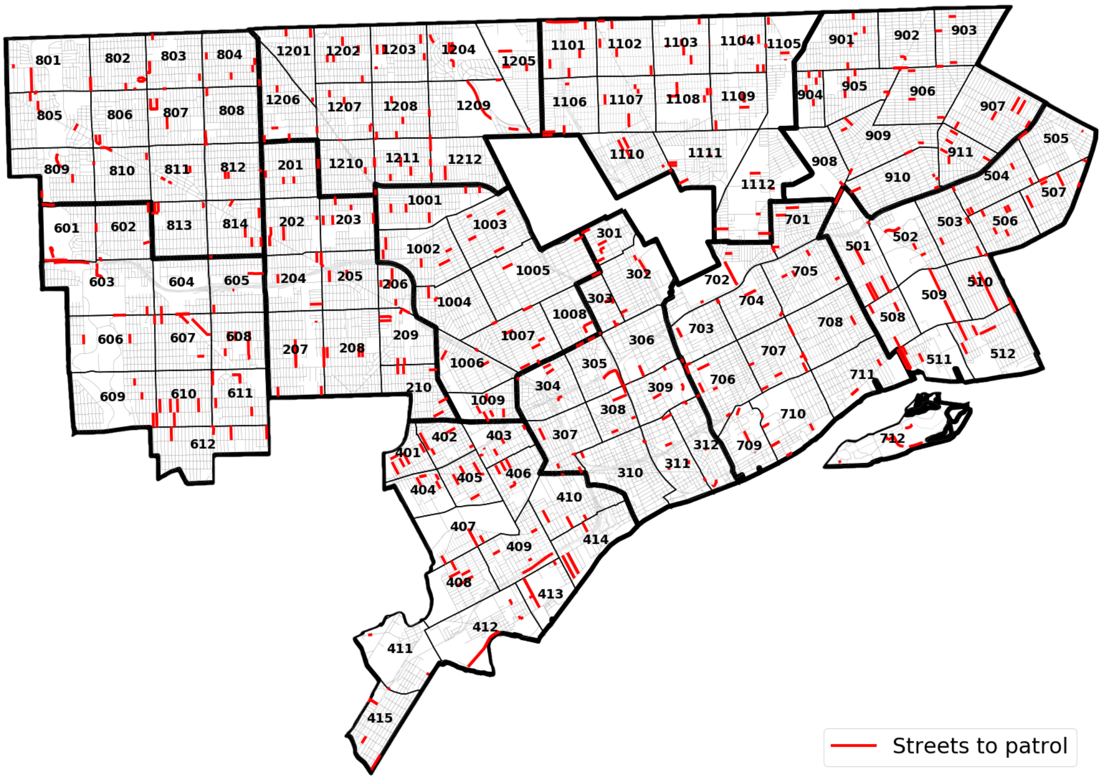

Figure 4.6: 'Hottest' streets to patrol for exemplar time period (16:00 to midnight on Monday, 3rd Sept 2018).

4.4 Reactive demand (CFS) in Detroit

As detailed in Chapter 2, previous research has shown that crime of various types (Brantingham and Brantingham, 1993; Brantingham and Brantingham, 1984; Brantingham et al., 1976; Pyle, 1976; Pyle and Hanten, 1974; Rengert, 1980)) and calls for service (e.g. Boulton et al., 2017; Vaughan et al., 2018) concentrate both spatially and temporally. Having introduced and pre-processed both the CFS and crime datasets, this section provides an exploration and visualisation of the spatial and temporal patterns of reactive demand in Detroit. Understanding the spatial and temporal distribution of reactive demand in the exemplar force chosen here is key to interpreting model outcomes. Indeed, the simulation may show varying levels of reactive effectiveness on different time periods (e.g. Friday night shifts) or in different precincts due to variations in reactive demand.

Call description category	Daily count	% of all calls
Other	33	25.98
FELONIOUS ASSAULT IP	29	22.83
BURGLARY OCCUPIED RESD I/P	16	12.60
SHOTS FIRED IP	15	11.81
DV A/B I/P-J/H	13	10.24
DISTURBANCE	11	8.66
HOLD UP ALARM AND MOW	10	7.87

Table 4.2: Number and percentage of incidents for each call category in the pre-processed CFS dataset (January 1, 2017 - December 31, 2019)

Temporal variations in reactive demand

Existing literature suggests a heterogeneous distribution of crimes by time of the day, day of the week and month of the year (see Chapter 2). Here, we investigate whether this is true for CFS in Detroit. As discussed in Chapter 2, there exist various metrics of reactive demand, from call volume to officer time. In a first instance, reactive demand is here quantified using the volume of calls received.

Emergency incidents in the CFS dataset are classed in 176 different description categories. Table 4.2 summarises the average number of daily emergency calls received for these category. The figures suggest that in-progress felonious assaults are the most common type of emergency calls (about 23% of all calls).

Figure 4.7 displays the hourly call volume received by DPD on weekdays (Monday to Friday) and weekends (Saturdays and Sundays). Generally speaking, the volume of emergency calls fluctuates through the day with a peak reached around 02:00 (between 9 and 10 calls an hour on average) and a dip between 10:00 and 11:00 (about 3 calls an hour on average). The volume of emergency calls thus triples between 10:00 and 02:00 the next day. The difference between weekends and weekdays is most striking between midnight and 11:00, with a longer peak observed on weekends. These results suggest that reactive demand in Detroit fluctuates based on the time of the day and day of the week and is highest on time periods most popular for their night life (Friday and Saturday nights).

In this thesis, days are divided into three 8-hour time periods on which to run simulations independently: (1) *day*: 08:00 - 16:00, (2) *late*: 16:00 - 00:00, (3) *night*: 00:00 - 08:00. These time periods loosely align with the typical design of police shifts, although the latter tend to

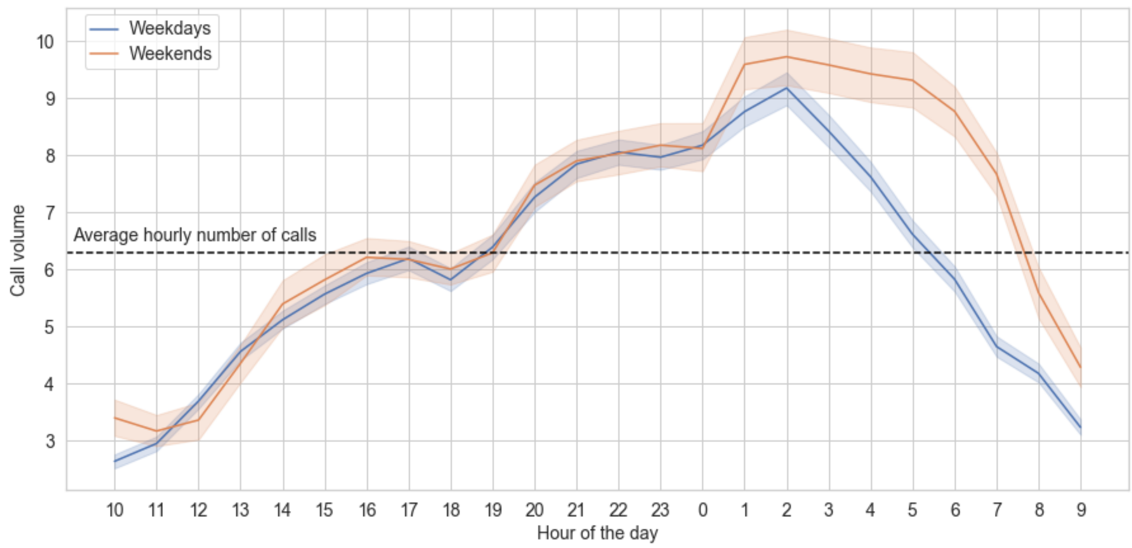

Figure 4.7: Hourly call volume received by DPD on weekdays and weekends (January 1, 2017 - December 31, 2019). The curves show the mean and 95% confidence intervals around the mean (using the bootstrap method).

overlap to ensure continuous servicing.

With these 3 daily shifts in mind, a two-way ANOVA was conducted to determine the effect of both shift and day of week on the volume of emergency calls. A statistically-significant difference was found in the number of calls received between shifts ($f(2)=1430.277$, $p<0.001$) and between days of the week ($f(6)=13.990$, $p<0.001$), as well as between interactions of these terms ($f(12)=5.497$, $p<0.001$). The effect size of these differences was calculated using the eta-squared (η^2) measure. The general rule of thumb given by Cohen (1988) for analysing eta-squared is provided in Table 4.3. The differences between shifts produced a large effect size (η^2=0.456), while those between days of the week and between interactions of the two terms produced a small effect size (η^2=0.013 and η^2=0.010 respectively).

Table 4.3: Quality of effect size based on the value of eta squared

η^2 value	**Quality**
~ 0.01	Small
~ 0.06	Medium
> 0.14	Large

A further Tukey post-hoc test on the 3 shifts revealed significant pairwise differences between:

- *day* and *late* shifts (+23 calls on *late* shifts),
- *day* and *night* shifts (+30 calls on *night* shifts),

Call description category	Time on Scene (mins)		Daily count
	Mean	Std	
Other	73.00	80.00	33
FELONIOUS ASSAULT IP	68.47	78.20	29
BURGLARY OCCUPIED RESD I/P	57.79	62.58	16
SHOTS FIRED IP	48.84	71.12	15
DV A/B I/P-J/H	56.83	67.09	13
DISTURBANCE	29.43	35.95	11
HOLD UP ALARM AND MOW	21.31	22.27	10

Table 4.4: Average and standard deviation for time on scene by call category in the pre-processed CFS dataset (January 1, 2017 - December 31, 2019)

- *late* and *night* shifts (+7 calls on *night* shifts)

With regards to days of the week, a Tukey post-hoc test revealed significant pairwise differences between weekends and weekdays (+5 calls on weekends). In addition, night shifts on weekends also appeared significantly different from those on weekdays.

Taken together, results suggest that, in accordance with the literature, DPD experiences significant differences in terms of emergency call volume based on the shift and the day of the week, with the highest volume experienced on shifts popular for their night life (i.e. night shifts on Saturdays and Sundays).

Spatial variations in reactive demand

While call volume is an informative metric to quantify reactive demand, it fails to account for the amount of time that officers are required to spend at the scene of each incident. Indeed, some categories of incidents require a longer time at the scene than others (see Table 4.4). For instance, an in-progress felonious assault requires on average 68 minutes of officer time at the scene while an average disturbance incident only requires about half that amount of time (29 minutes). As such, an arguably better way of quantifying reactive demand is to consider the officer workload, here defined as the cumulative amount of time that officers are required to spend at the scene of incidents during a given time period. This metric is used in the exploration of spatial patterns that follows.

Figure 4.8 shows the spatial disparities in reactive demand workload between precincts of DPD. For instance, precincts 8 and 9 both require about 20 hours a day of officer time, while precincts 3, 4 or 7 require less than 10 hours a day. Reactive demand workload is thus more than twice as high in precincts 8 and 9 than it is in precincts 3, 4, or 7. Considering these results, one potentially desirable deployment configuration may involve positioning more agents in those precincts 8 and 9 that typically exhibit a higher reactive demand workload. Such a configuration, when simulated in the ABM, is expected to yield a smaller response time than alternative ones with fewer agents deployed to these precincts.

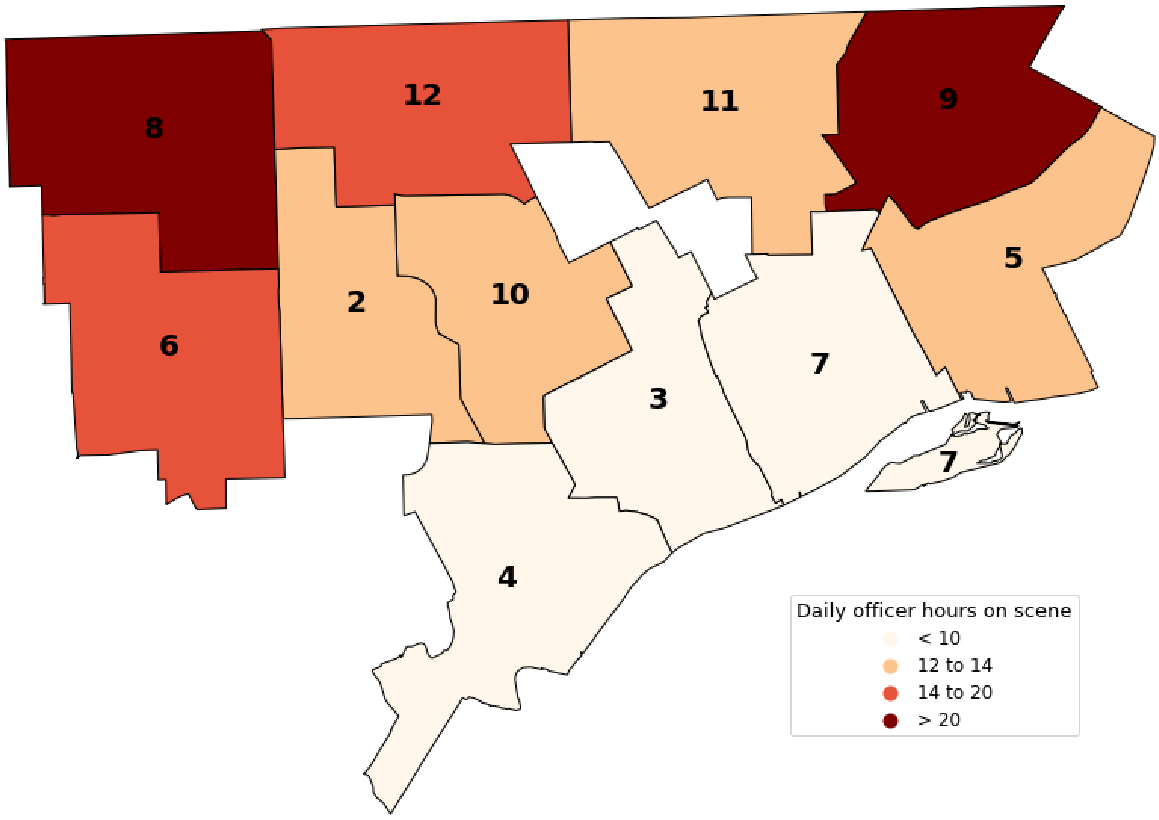

Figure 4.8: Daily number of officer hours spent on scene across DPD's precincts (January 1, 2017 - December 31, 2019).

Summary: reactive demand in Detroit

All in all, the exploration of the CFS dataset suggests that DPD experiences spatial and temporal variations in its reactive demand. With regards to temporal demand, more calls are received on weekend *night* shifts than on other shifts. Reactive demand is also spatially unevenly distributed across the force, with some precincts accounting for twice as much reactive demand workload as others. Although not investigated in this chapter as it is beyond the scope of this thesis, there are likely some interactions between these spatial and temporal patterns. For instance, the downtown area may experience more reactive demand on weekend *night* shifts compared with the rest of the week. Overall, these variations in reactive demand are bound to impact the reactive effectiveness of DPD when responding to calls.

In addition to responding to calls, officers on duty are required to deter crime through patrolling in their designated patrol beat. The next section provides an overview of the proactive demand experienced by DPD, by exploring the spatial and temporal patterns of reported crimes across the force.

4.5 Proactive demand (crime) in Detroit

In the model, the density of historical crimes on patrolled road segments is used to estimate the potential deterrent effect of proactive patrolling. This section provides an exploration of the spatial and temporal patterns of reported crimes in Detroit. Although reported crime does not depict an exact picture of crime (due to high rates of unreported crime in some areas), it provides a useful estimation of the quantity and location of crime across time and space. Table 4.5 shows the percentage of daily crimes for the main categories of reported crime. The most common type of reported crime relates to assaults (18.4% of daily crimes) followed by larceny (16.98% of daily crimes).

Crime category	Daily count	% of all crimes
ASSAULT	39	18.40
LARCENY	36	16.98
OTHER	32	15.09
DAMAGE TO PROPERTY	27	12.74
AGGRAVATED ASSAULT	22	10.38
BURGLARY	21	9.91
STOLEN VEHICLE	18	8.49
FRAUD	17	8.02

Table 4.5: Number and percentage of reported crimes per category in the reported crime dataset (January 1, 2017 - December 31, 2019)

Temporal variations in volume of reported crimes

Figure 4.9 shows the hourly variations in number of reported crimes (from January 1, 2017 to December 31, 2019). It suggests that, much like CFS, the number of reported crimes fluctuates throughout the day from around 4 crimes an hour at 14:00 to about 15 crimes an hour at 08:00. This is likely because victims tend to report crimes that occurred during the night in the next morning. While the difference between weekends and weekdays is subtle, there appears to be more crimes reported between 06:00 and 14:00 on weekends than on weekdays but fewer crimes between 14:00 and 06:00 the next day on weekends compared with weekdays.

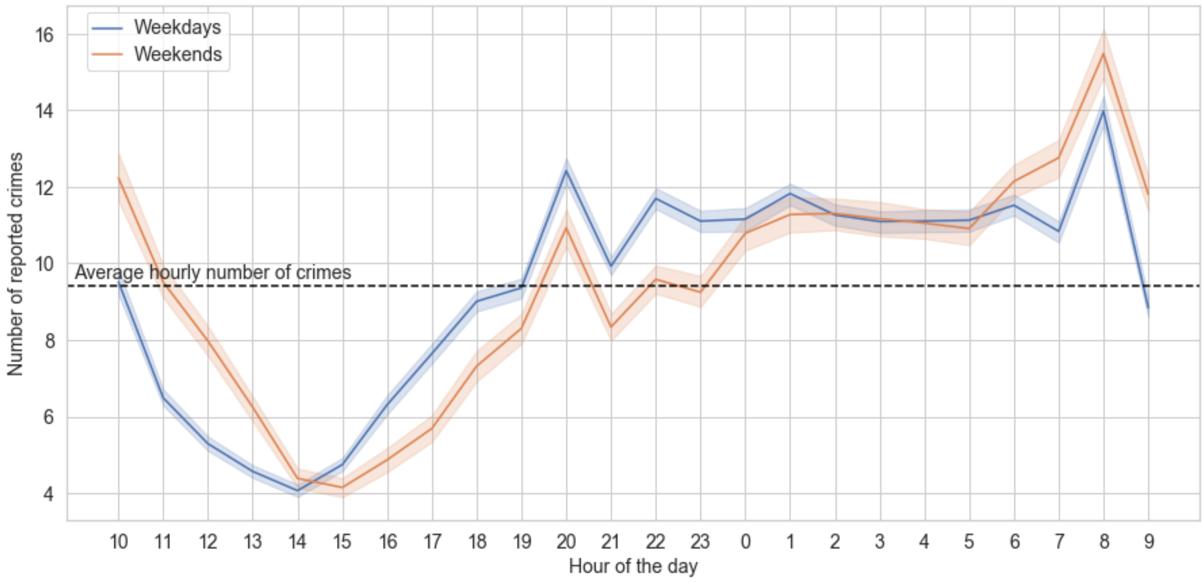

Figure 4.9: Hourly variations in number of reported crimes (January 1, 2017 - December 31, 2019).

Note: the curves show the mean and 95% confidence intervals around the mean (using the bootstrap method).

Spatial variations in volume of reported crimes

Figure 4.11 shows the daily density of reported crime across the scout car areas of DPD (from January 1, 2017 to December 31, 2019). Reported crime appears highest in Detroit's downtown and midtown areas (scout car areas 312, 311 and 309) with a density of 2 crimes per day per km^2. Other hot spots include scout car area 303, and wider areas such as the north east part of precinct 9 (around scout car area 907), the border between precinct 8, 2 and 12 (around scout car area 201) as well as the south east corner of precinct 6 (around scout car areas 607 and 610).

As ever with crime, the available data only describes partial aspects of a much complex picture. For instance, as previously mentioned, private security forces are in service in precinct 3, such as Downtown PD or Wayne State University PD. These private forces patrol areas around downtown and midtown, alleviating the demand faced by the precinct. As such, although these areas appear to receive an abnormally low number of CFS given the high density of reported crimes in the locality, this is likely due to the fact that the calls assigned to these other police departments were excluded from the dataset.

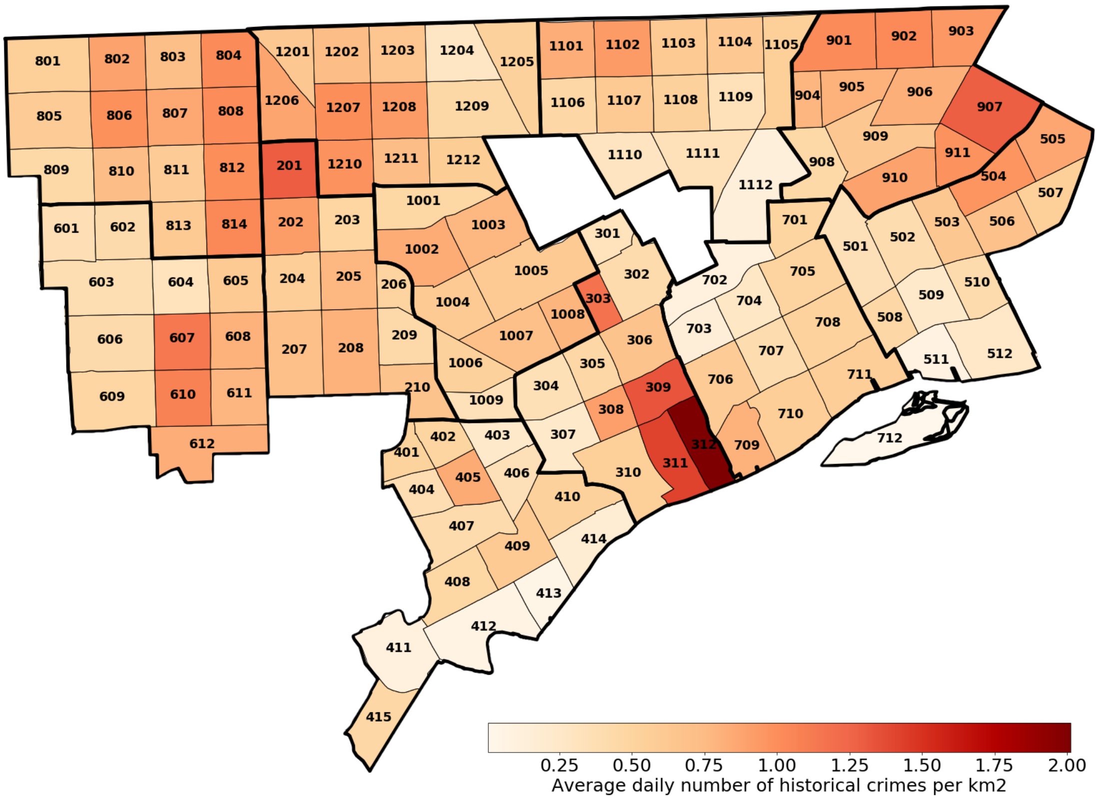

Figure 4.11: Daily number of historical crimes per km^2 across scout car areas (January 1, 2017 - December 31, 2019).

Summary: proactive demand in Detroit

Overall, the proactive demand faced by DPD seems to exhibit spatial and temporal patterns. The number of reported crimes is low on *day* shifts and high on *night* shifts and some scout car areas feature a higher density of crimes than others. These patterns of reported crime suggest that a potentially desirable deployment configuration – one that yields a high proactive effectiveness – may involve the targeted deployment of patrols to those areas with the highest density of historical incidents (scout car areas 312, 311, 309, 303, 907, 201, 607, 610 etc.), and to those times where there is most crime occurring (weekend *night* shifts).

4.6 DPD's reactive effectiveness

This section provides a visualisation of DPD's reactive effectiveness, evaluated based on the incident response times provided in the CFS dataset. Importantly, the patterns of response observed in this section form a benchmark against which the realism of the behaviour of the

ABM agents is validated in the next chapter.

4.6.1 Distribution of incident response times

Figure 4.12 shows the overall distribution of DPD's response times to incidents. As previously mentioned, this is the time between the call coming in and the first responder reaching the scene of the incident. The average response time across the studied time period was 10.2 minutes, the median was 8.1 minutes and the standard deviation was 8.6 minutes. As expected, the distribution is highly positively skewed, due to rare responses with longer response times.

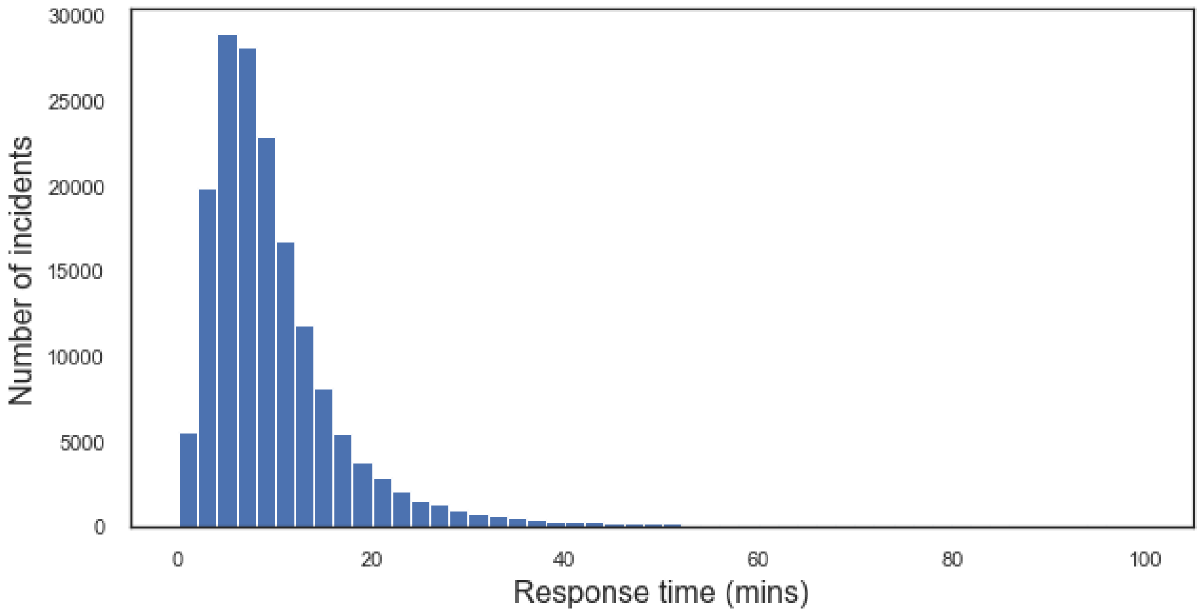

Figure 4.12: Distribution of incident response times at DPD (January 1, 2017 - December 31, 2019)

4.6.2 Temporal variations in reactive effectiveness

Figure 4.13 shows the evolution throughout the day of (1) the average inter-arrival time – i.e. the average time (in minutes) between the arrival of two consecutive calls, (2) response time and (3) time on scene. The inter-arrival time for each incident in the dataset is calculated as the time elapsed (in minutes) since the previous call. Then, these incident inter-arrival times are grouped by hourly period of the day and their values averaged. The resulting average inter-arrival time is shown here to represent reactive demand. Its value decreases from a maximum of 25 minutes at 10:00 to a minimum of 7 minutes at 02:00, when the frequency of calls is the highest. Incident response times remain relatively constant throughout the day (around 10 minutes on average), with two small peaks at 19:00 and 02:00 (around 12 minutes on average).

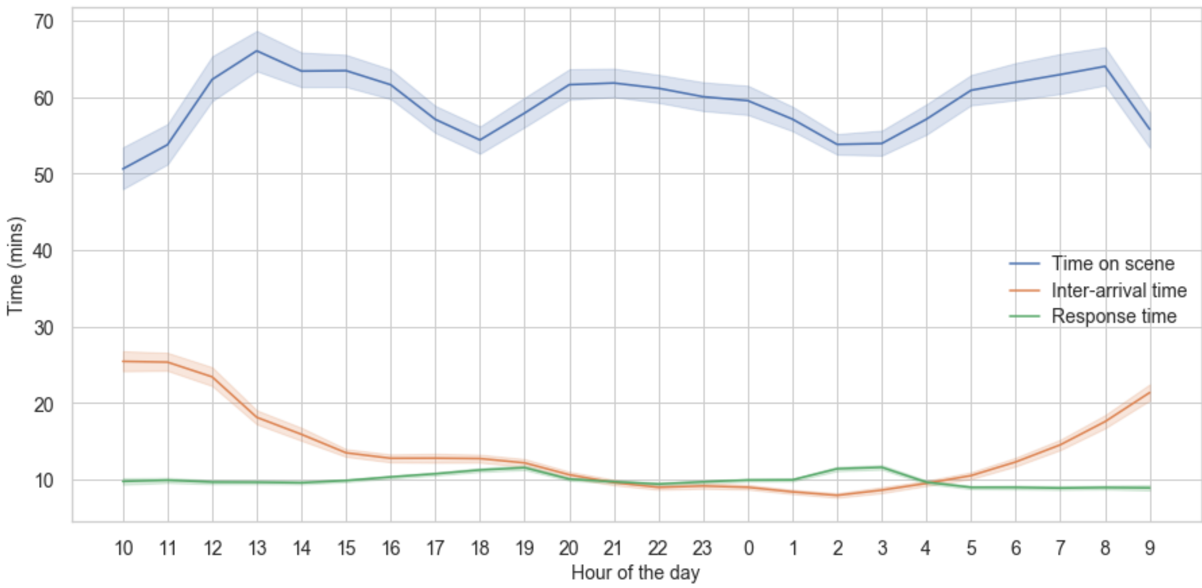

Figure 4.13: Temporal evolution of DPD's average inter-arrival time, response time and time at the scene throughout the day (January 1, 2017 - December 31, 2019).

Note: the curves show the mean and 95% confidence intervals around the mean (using the bootstrap method).

While the latter peak appears to match the peak of reactive demand observed in Figure 4.7, the former does not. Instead, it may result from a variety of causes internal to DPD such as their deployment configuration or the design of their staff rota. Time on scene oscillates throughout the day between 50 and 65 minutes. Overall, these results suggest that, while the volume of emergency calls fluctuates throughout the day, DPD's response to these calls remains relatively constant.

Figure 4.15 provides a more detailed picture by breaking down the response time into its dispatch time and travel time components. Dispatch time exhibits two peaks during the day at 19:00 and between 02:00 and 03:00, which matches the two peaks observed for response time in Figure 4.13. Similar dispatch times can be observed between weekdays and weekends, with the exception of the period between 05:00 and 11:00 in which weekends feature a higher average dispatch time. Travel time, on the other hand, appears to remain relatively constant throughout the day (between 6 and 8 minutes on average), with a slight decrease between 20:00 and 10:00, possibly linked to a more fluid traffic at night. Travel times appear to be relatively similar on weekdays and weekends.

A two-way ANOVA was conducted in order to determine the effect of the shift (*day*, *late* and *night*) and day of week on response times. No statistically significant differences were found in

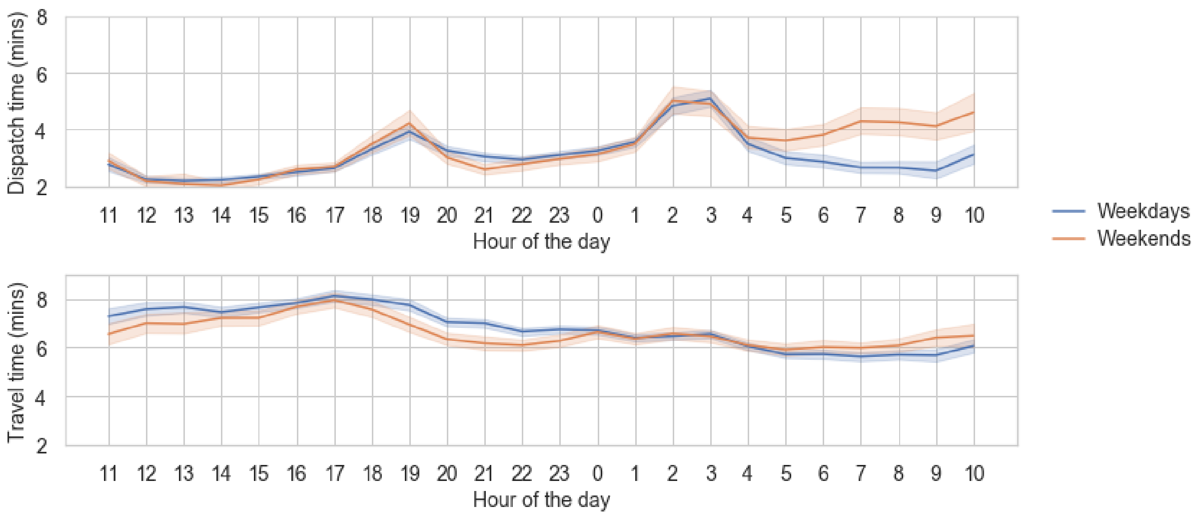

Figure 4.15: Temporal evolution of DPD's dispatch and travel times throughout the day on weekends versus weekdays (January 1, 2017 - December 31, 2019).

Note: the curves show the mean and 95% confidence intervals around the mean (using the bootstrap method).

DPD's response times between days of the week. Although a statistically significant difference in response times was found between shifts ($f(2)=17.056$, $p<0.001$) and for the interactions between shifts and days of week ($f(12)=3.243$, $p<0.001$), the magnitude of these effects was found to be negligible through a Eta-squared calculation. As such, it may be concluded that there is no significant difference in emergency incident response times based on the shift and day of the week. Instead, response times remain consistent with an average of 10.2 minutes. This seems to indicate that DPD's existing deployment is designed to match the variations in reactive demand (for emergency calls): in other words, they likely deploy a higher level of supply on high-demand shifts.

4.6.3 Spatial variations in reactive effectiveness

Having analysed the temporal patterns in DPD's reactive effectiveness, this part now looks into its spatial variations across the force. Figure 4.17 provides a visualisation of the spatial distribution of response time summary statistics (mean, median and standard deviation) on the map of Detroit. While it appears that the mean response time is highest in precinct 8 and 9 – both of which exhibit the most reactive demand workload as seen in Section 4.4, this pattern is less striking when looking at the median. The median for response times ranges from 6.7 minutes (in precinct 3) to 8.8 minutes (in precinct 8 and 12) and the standard deviation is constant across precincts. There thus appears to be little spatial variation in response time across DPD.

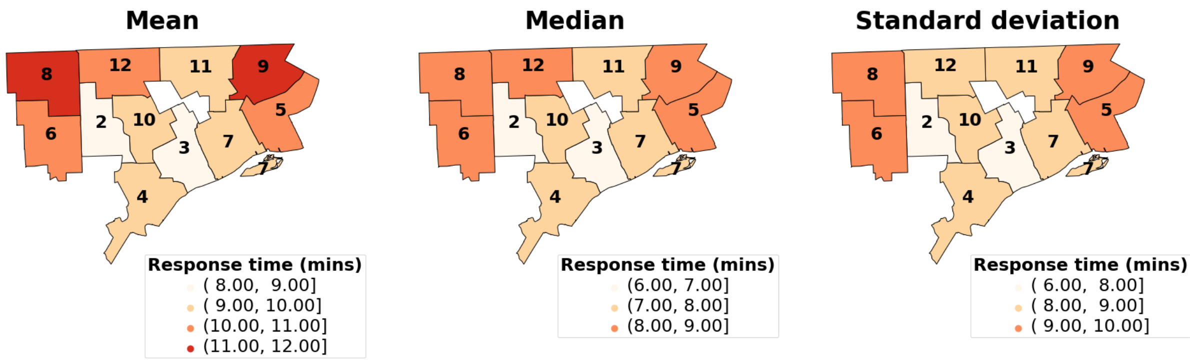

Figure 4.17: Summary statistics for incident response times across DPD's precincts (January 1, 2017 - December 31, 2019).

The statistical significance of the null hypothesis (i.e. no significant difference in the average response time between precincts) was determined with pairwise Kruskal-Wallis non-parametric ANOVAs at $p = 0.01$. This statistical test, which is a non-parametric equivalent of the one-way analysis of variance (ANOVA) was used because the data is not guaranteed to follow a normal distribution. The results indicate a significant difference between precincts (H = 2847.345, p<.01). Given the abundance of statistical power provided by the large number of incidents, analysis of the magnitude of difference in response time replaces conventional post hoc tests of statistical difference. The effect size, was evaluated by calculating the epsilon squared value (Kelley, 1935). Table 4.6 indicates the chosen interpretation of quality of the effect size based on epsilon squared values (Rea and Parker, 1992). The effect size was found to be weak ($\epsilon^2 =$ 0.017). Overall, there appears to be no significant differences in DPD's reactive effectiveness (with regards to emergency calls) across its 11 precincts.

Table 4.6: Quality of effect size based on the value of epsilon squared

ϵ^2 **value**	**Quality**
[0, 0.01)	Negligible
[0.01, 0.04)	Weak
[0.04, 0.36)	Relatively strong
[0.36, 0.64)	Strong
[0.64, 1)	Very strong

4.6.4 Summary DPD's reactive effectiveness

All in all, DPD's response times to emergency calls (for the studied period) appear relatively constant in space and time. This suggests that DPD have designed their deployment strategies so as to provide a relatively uniform response to emergency CFS both across time and across all precincts of the force. The patterns of response times identified in this section are used in Chapter 5 to compare against those produced by the ABM, with the view to validating the model.

4.7 Conclusion

The ABM developed in Chapter 3 is designed to be applied to any police force. For the purpose of demonstration and model validation, the ABM is applied to the case study of Detroit Police Department (Michigan, US), which is chosen for its publicly available datasets. This chapter introduced DPD as a police force (Section 4.2), and listed the data sources used in the model as well as the processing steps undertaken to prepare them for the ABM (Section 4.3). These data sources include the real road network, the boundaries of the patrol beats, the historical CFS incidents and the historical reported crimes for the three-year study period. Then, a description of the spatial and temporal patterns which are present in the data was provided for both the CFS dataset (Section 4.4) and the crime dataset (Section 4.5). Finally, Section 4.6 provided a brief analysis of the existing reactive effectiveness of DPD based on the response times provided in the CFS data.

Taken together, the observations made in this chapter suggest that both reactive and proactive demand faced by DPD are unequally distributed in space and time. Some deployment configurations implemented in the ABM may thus yield better response times on certain shifts and worse ones on others. Similarly, for the same shift, certain deployment configurations may fare better than others because they position agents in strategic patrol beats that are close to arising CFS demand. It may thus be necessary for deployment configurations to be tailored to each shift and day of the week, as will be discussed in the next chapter. Going back to the ABM, the next chapter uses DPD's case study to perform a sensitivity analysis, validate the model and run a range of simulation experiments to showcase the potential of the ABM as a tool to explore and quantify the impact of various deployment decisions.

Chapter 5

Model analysis, validation, and simulation experiments for the case study of Detroit

5.1 Introduction

George Edward Pelham Box famously coined the phrase 'all models are wrong, but some are useful'. While models always fall short of the complexities of the real system that they are emulating, they can still be informative. However, for a model to be usable in an applied context, it needs to achieve a sufficient level of realism. This is typically verified through model validation against real-world data. Amongst all types of models, ABMs are notoriously difficult to validate.

Chapter 3 detailed the design decisions made when building the ABM. This was followed by Chapter 4, which provided an overview of the spatial and temporal patterns of reactive and proactive demand in DPD, the exemplar police force for this thesis. Using the case study of DPD, this chapter conducts a series of analyses and experiments using the ABM, and establishes the validity of the ABM as a representation of patrol activities.

This chapter begins with this section which describes the range of supply and demand values that were tested throughout the analyses of the chapter. This range aims to account for the potential differential impact of supply or demand values on the performance of the system.

Then, a sensitivity analysis is performed in Section 5.2 to assess the sensitivity of the ABM to perturbations in the values of some key chosen parameters. Next, the ABM is validated against real-world data in Section 5.3 using the exemplar police force of DPD. This is first achieved with a simple face validation in which individual agents are followed throughout the simulation. Following this face validation, a population-level validation is conducted by comparing the validation patterns defined in Chapter 3 – i.e. distribution of dispatch and travel times – that were produced by the ABM with those observed in DPD's dataset (see Chapter 4). Finally, a series of simulation experiments are conducted in Section 5.4 in which the ABM is used to explore and quantify the impact of various deployment decisions on the performance of the system.

5.1.1 Modelling low and high demand

As shown in Chapter 4, demand for proactive and reactive policing in Detroit fluctuates between time of day and day of week. It was shown for instance that a Friday or Saturday night shift typically experiences more demand than a Monday afternoon/late shift. The chosen time period on which the ABM is run is thus likely to affect model outcome. As such, running the ABM for a single time period (e.g. one Saturday night shift) risks skewing the interpretation to that particular time period and limit generalisation.

To alleviate the bias that comes with running the model on a single time period whilst also taking into account the variations in demand between shifts, two sets of time periods were created to represent two distinct demand scenarios:

- **Low-demand scenario**: day shifts (from 08:00 to 16:00) on weekdays (Monday to Friday)
- **High-demand scenario**: night shifts (from 00:00 to 08:00) on Saturdays and Sundays

A demand scenario is represented by (1) its 'training set' of 100 randomly selected time periods from the year 2018 and (2) its 'test set' of 100 randomly selected time periods from the year 2019. To illustrate, the 'training set' and 'test set' of the low-demand scenario are both made of 100 randomly selected weekday day shifts (08:00 - 16:00) from 2018 and 2019 respectively, while those of the high-demand scenario are composed of 100 randomly selected 'weekend' night shifts (00:00 - 08:00 on Saturday and Sunday) from the same two years. These four sets (two per scenario) are used throughout the rest of this thesis to evaluate the performance of the system as part of various experiments. The time periods from which the sampling is done are summarised in Table 5.1.

Table 5.1: Time periods sampled to make up the training and test sets for both low-demand and high-demand scenarios

Scenario	**Days**	**Times**	**Set**	**Year**
Low-demand	Monday to Friday	08:00 to 16:00	Training	2018
			Test	2019
High-demand	Saturday, Sunday	00:00 to 08:00	Training	2018
			Test	2019

Running the model for both low and high-demand scenarios is important because a particular deployment configuration may have little effect on model outcome on a low-demand shift but a strong one on a high-demand one (or vice-versa).

5.1.2 Modelling low and high supply

The number of agents in the simulation is also likely to affect the overall performance of the simulated police force (see emergent outcomes detailed in Chapter 3). It is to be expected that the more agents deployed to a precinct, the more available agents may be dispatched to incidents. This ultimately results in shorter dispatch times (to a minimum dispatch time of 1 minute, i.e. the time step for the ABM). Travel times are also affected by the chosen deployment configuration, as an agent deployed to a patrol beat with a high density of historical incidents is more likely to find themselves in the proximity of arising incidents. To account for various levels of supply, the analyses and experiments in this chapter were conducted on several configurations of 10, 20, 30, 40, 50 and 60 agents deployed across the force. In addition, these agents are positioned either at random or in a targeted fashion based on historical demand (details are provided throughout the chapter).

5.2 Sensitivity analysis

It is common for ABMs to undergo a process of calibration, which involves identifying the parameter values that lead to the expected system behaviour (Grimm et al., 2020). Typically, the parameters most suitable for calibration are those to which model results are highly sensitive and for which there is little basis, other than calibration, for selecting values (Grimm et al., 2020).

Most of the model parameter values used in this ABM are based on data from the real world.

For instance, for a given police force, the model environment is composed of (1) the real road network: with information concerning the speed limit on each road segment and oneway roads and (2) the real police administrative areas: including precincts and patrol beats. In addition to the model environment, the ABM also relies on real historical CFS and reported crime data for the chosen police force.

Nonetheless, there are several arbitrarily chosen parameter values, in particular (1) the number of streets to be patrolled in each patrol beat (currently set to 5), and (2) the agent's driving speed on each road segment (currently the maximum speed limit on each road). To identify how sensitive the model is to perturbations in these parameter values, a One-Factor-at-A-Time (OFAT) sensitivity analysis was performed. In an OFAT sensitivity analysis, a nominal set of parameters is selected, then one parameter is varied at a time while keeping all other parameters fixed (ten Broeke et al., 2016). An important use of OFAT is to reveal the form of the relationship between the varied parameter and the output, given that all other parameters have their nominal values. For example, it shows whether the response is linear or nonlinear, or whether there are tipping points where the output responds drastically to a small parameter change. By showing these relationships, OFAT can yield understanding of model mechanisms (ten Broeke et al., 2016).

As mentioned above, the sensitivity analyses were conducted on several configurations of 10, 20, 30, 40, 50 and 60 agents that are randomly deployed across the force. These random configurations provide a benchmark to compare the impact of different numbers of deployed agents on the system. Examples of these random deployment configurations for various numbers of deployed agents in DPD are displayed in Figure 5.1. Later in this chapter, in order to validate the ABM, the model is instead run for deployment configurations which target patrol beats with a high density of historical crimes, as it is more likely to resemble the real configuration implemented by DPD.

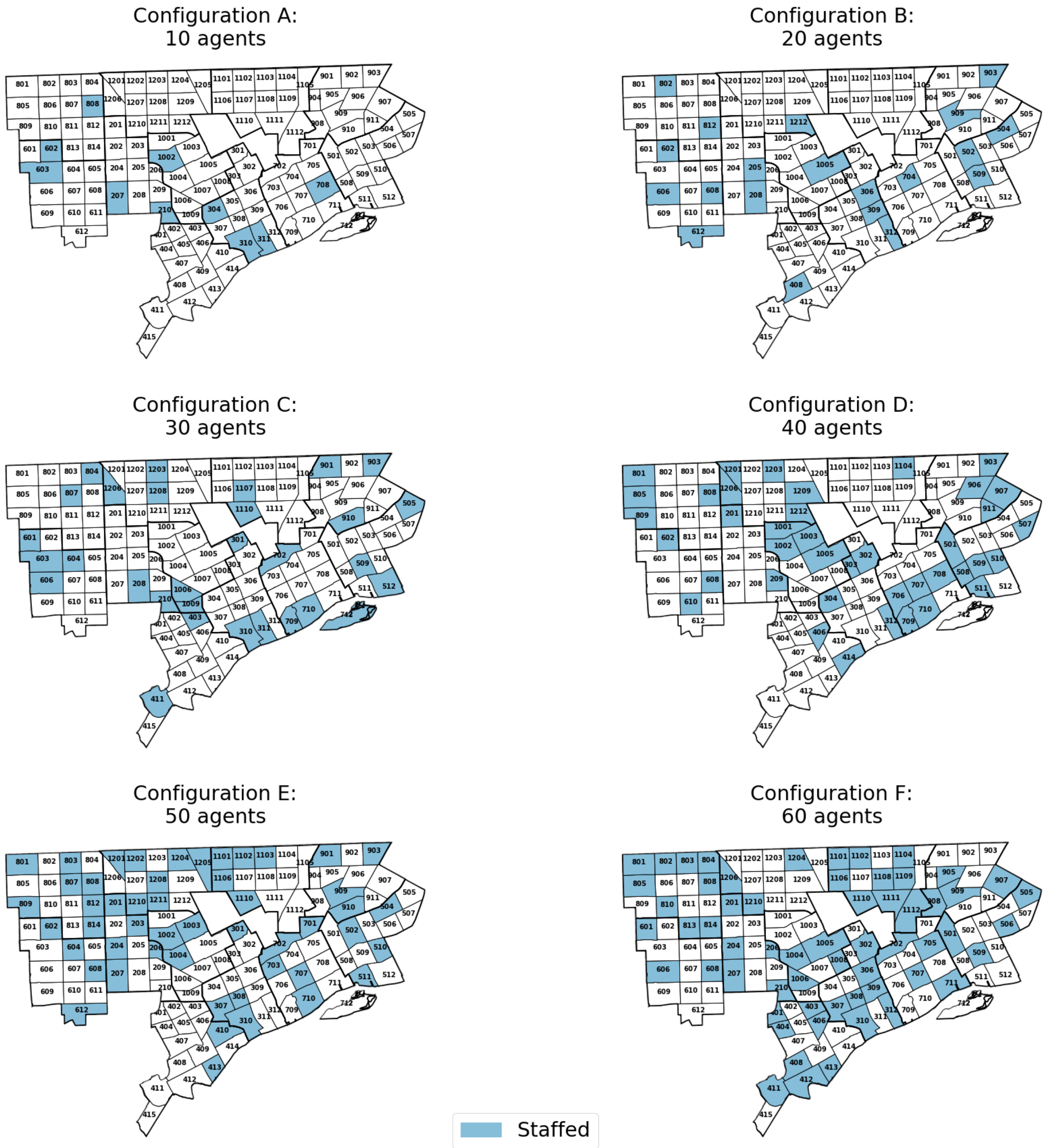

Figure 5.1: Examples of random deployment configurations for various numbers of deployed agents in DPD.

5.2.1 Driving speed of agents

In the model, agents drive along the road network at the maximum speed limit of each road segment. However, in the real world, the speed of responding vehicles may differ from that value based on various external factors such as traffic fluidity. In fluid traffic, vehicles may be able to drive faster than the speed limit as they deploy their blue light and siren. In congested traffic on the other hand, they may have to drive slower than the speed limit.

In order to evaluate how patterns of travel time are affected by variations in the driving speed of the agents, a sensitivity analysis is performed in which 6 modifiers are tested for the driving speed of agents: an increase of 10%, 20%, and 30% as well as a decrease of 10%, 20% and 30% from the speed limit on each road segment. Given that the ABM is deterministic (see ODD in Chapter 3), multiple runs of the model with the same parameters yield identical outcomes. As such, it is not necessary to conduct multiple runs of the model for a given time period. Considering both low and high demand scenarios as well as the aforementioned various supply levels, a total of 1,200 ABM runs were evaluated for each parameter value (see Table 5.2 for a breakdown of the supply and demand values used in this analysis).

Table 5.2: Values used in the sensitivity analysis of the driving speed of agents

Variable	**Number of values**	**Values**
Driving speed modifiers	7	-30%, -20%, -10%, 0%, +10%, +20%, +30%
Demand scenarios	2	Low-demand, high-demand
Time periods	100	From scenario's 'test set' (2019)
Number of agents	6	10, 20, 30, 40, 50, 60
Deployment type	1	Random

The overall distributions for the average response time outputted by the model for each travel speed modifier value are displayed in Figure 5.2. A Kruskal-Wallis test was conducted to determine whether the travel speed of the agents had an effect on the average response time. The results indicate a significant difference between groups for the low-demand scenario (H=311.509, $p<.05$) with a moderate effect size ($\epsilon^2=0.074$) and for the high-demand scenario (H=97.619, $p<.05$) with a weak effect size ($\epsilon^2=0.023$), possibly due to high variability across time periods. These results suggest that, while the driving speed of agents may affect the average response time in a low-demand scenario, the model's sensitivity to the agent's travel speed remains relatively low. As such, the value arbitrarily chosen in the model, which corresponds to the maximum speed limit on each road segment (0% modifier), appears to be a sensible one.

5.2.2 Number of streets to patrol per beat

During the initialisation of the model, 5 streets in each beat are chosen to be visited as part of a patrol route (see details in Chapter 3). This selection is made either at random or based on the streets' density of historical crimes – when a dataset of historical crimes is provided upon model initialisation. This number of streets to patrol in each beat is arbitrary and may affect the overall crime deterrence score, which is an emergent outcome of the model. In order to

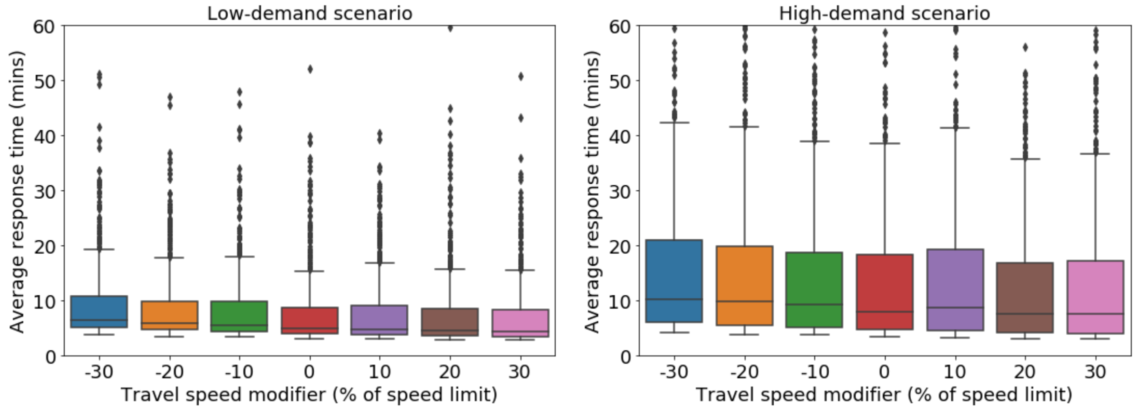

Figure 5.2: Distribution of incident travel times for various agent travel speed modifier values for low and high-demand scenarios.

explore the impact of this number on the overall crime deterrence score, a sensitivity analysis was performed for 2, 5 and 10 streets to patrol per beat. Much like for the previous analysis, a total of 1,200 ABM runs were conducted for each parameter value (see Table 5.3 for a details).

Table 5.3: Values used in the sensitivity analysis of the number of streets to patrol in each beat.

Variable	**Number of values**	**Values**
Number of hot streets	3	2, 5, 10
Demand scenarios	2	Low-demand, high-demand
Time periods	100	From scenario's 'test set' (2019)
Number of agents	6	10, 20, 30, 40, 50, 60
Deployment type	1	Random

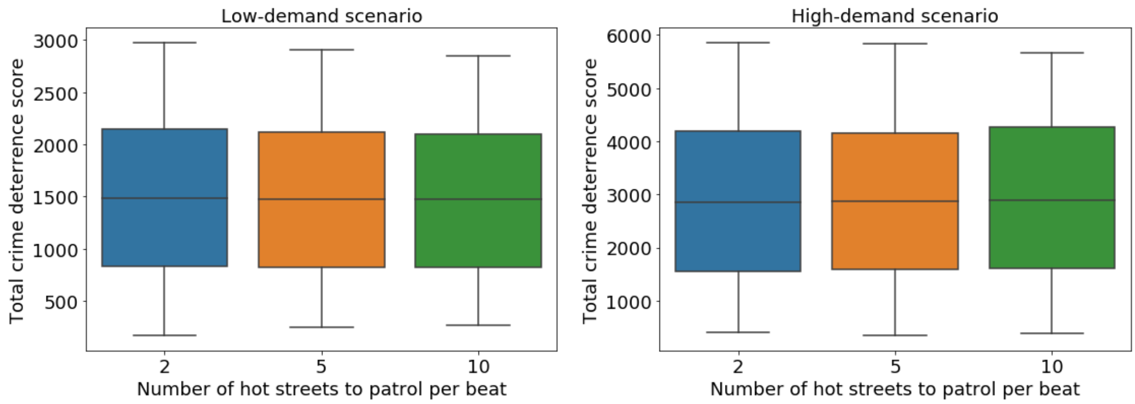

Figure 5.3: Distribution of total crime deterrence score for 2, 5 and 10 streets to patrol per beat for low and high-demand scenarios.

The overall distributions of total deterrence scores that were outputted by the model for each parameter value is displayed in Figure 5.3. A Kruskal-Wallis test was conducted to determine whether the number of hot streets to patrol in each beat had an effect on the total crime deterrence score. The results indicate no significant difference between groups (low-demand scenario: H=0.465, p>.05; high-demand scenario: H=0.265, p>.05). These results suggest that the total crime deterrence score is not affected by the chosen number of streets to patrol in each beat. The model does not appear to be sensitive to the value of this parameter. As such, the value of 5 streets chosen for the model appears to be a reasonable choice.

This result further evidences that the patrol behaviour implemented in this first version of the model is fairly inconsequential in terms of crime deterrence. Indeed, the calculation of the crime deterrence score is here merely a function of agents' idle time. Several improvements on the implemented patrolling behaviour and the deterrence score calculation are suggested in Chapter 8.

5.2.3 Summary: sensitivity analysis

In this section, a One-Factor-at-A-Time (OFAT) sensitivity analysis was performed using DPD's case study for the following 2 parameters: (1) the driving speed of the agents on each road segment and (2) the number of streets to patrol in each beat. Results suggest that, in the case of DPD, the model is not highly sensitive to the values of these parameters. As such, these parameters do not require calibration; instead, the chosen values appear reasonable. In the next section, the model's outputted performance metrics are validated against real data for DPD.

5.3 Model validation

Having established that the model is not overly sensitive to arbitrary parameter values, the next step is to assess whether the model achieves the required level of realism to be useful for its purpose – a process known as validation (North and Macal, 2007).

5.3.1 Challenges in validating a model of the police system

According to Axtell and Epstein (1994), there are 4 levels of model validation. Table 5.4 adapts this framework to the problem of patrol deployment studied in this thesis.

As mentioned in Chapter 2, police patrol activities are complex due to the spatio-temporal interactions of units with their environment. Patrol units engage in an array of different activities throughout their shift. Furthermore, real-world travel time to incidents is dictated by highly unpredictable factors such as the location of patrols when dispatched, the speed at which they

Table 5.4: The 4 levels of validation applied to the patrol deployment problem

Level	**Model type**	**Validation approach**
Level 0	Caricature of reality	Simple visualisations verify that agents behave as expected
Level 1	Qualitative agreement (population level)	The distribution of simulated dispatch and travel times visually matches that of times observed in the data
Level 2	Quantitative agreement (population level)	Statistical tests demonstrate that simulated and real dispatch and travel times are statistically similar
Level 3	Quantitative agreement (individual level)	Longitudinal analysis of individual incidents show that the dispatch and travel time for the incident is similar to that observed in the data.

drive to the incident, the route they take, as well as the status of the traffic lights and the fluidity of the traffic on their route.

Knowing the exact whereabouts of individual patrols at a given time is thus challenging, with past decisions constraining later ones (a concept known as path-dependence). While a few studies have attempted to realistically model the whereabouts of real police vehicles, these have been achieved at a relatively small scale and relied on the availability of GPS data (see Wise and Cheng, 2016). Such data, which track the precise spatial movement of police vehicles, are difficult to obtain for research purposes. One strength of the ABM developed here is its ability to model the patrol activities of any given police force, while requiring minimal external data sources. Although the absence of GPS data prevents a level-three validation based on individual-level comparisons between simulated and observed response time to each incident, it allows the model to be flexibly used on a wider range of contexts.

Importantly, the essence of any model is to simplify reality. For instance, equation-based models sacrifice realism for speed and simplicity. While a major strength of ABMs is their ability to incorporate some of the real world complexity (see discussion in Chapter 2), a level of abstraction is key to focusing on relevant behaviour. As such, modelling the exact whereabouts of police patrols may arguably result in over-fitting the model to the fuzzy movement of these individual units.

A good practice when modelling complex systems is to incrementally introduce complexity in model versions (Townsley and Birks, 2008). The model described in this thesis stands as a first version in which simplifications have been made. As such, the principle aim when building this ABM was to strike a balance between (1) the sufficient inclusion of real data for the model to be meaningful (i.e. usable for testing and comparing deployment configurations) and (2) abstracting away from the complex and superfluous details of real world policing.

Given that the model is to be used solely to compare deployment configuration alternatives against each other (and not against existing police deployment configurations), it is not necessary to produce a model with a level of fidelity that exactly matches that of the real world system. As such, the model is validated by abstracting away from individual responses with a face validation (level 0) and a qualitative validation (level 1) which verifies that the model produces the population-level patterns defined in Chapter 3. As a reminder, those validation patterns concerned the overall distribution of incident travel and dispatch times compared with that observed for incidents in DPD's CFS dataset.

5.3.2 Face validation (level 0)

It is critical for replication efforts to ensure that the implemented model matches the conceptual model – that is, that the code generated as part of this work correctly executes the processes defined in Chapter 3 (North and Macal, 2007). The verification process is necessary for sharing models, as without this step the generated outcomes may merely be the result of some peculiarity of the code (Galán et al., 2009; North and Macal, 2007).

To achieve the first level of validation, an animated visualised simulation of the model applied to DPD was produced with the view to assessing the general realism of the modelled system. In this visualisation, agents can be seen moving along the road network to patrol their designated beat and respond to incidents which appear on the map of DPD.

Furthermore, the model was manually inspected and verified to ensure that the dynamic attributes of model entities are appropriately updated throughout the simulation in response to occurring events. This was achieved by following individual agents, and ensuring that their behaviour was as designed. To give an example taken from an actual run of the simulation applied to DPD, the activities of one simulated agent are detailed in Table 5.5. The agent begins the simulation (08:00) patrolling their patrol beat. They are then dispatched to an incident at 10:19, when the first incident occurs in their precinct (precinct 6). Travelling to the scene of the incident takes them 6 minutes, after which they begin to tend to the incident. After 10 minutes at the scene, the incident is resolved and they route back to their assigned patrol beat where they resume patrolling until they are dispatched to the next incident almost 2 hours later.

The corresponding trail representing the nodes visited by this agent during the time period is

Table 5.5: Evolution of an agent's status throughout the simulated time period (2019-03-19 08:00:00 to 2019-03-19 16:00:00)

Time	Agent status
08:00:00	Patrolling
10:19:00	Travelling
10:25:00	At the scene
10:35:00	Patrolling
12:33:00	Travelling
12:37:00	At the scene
13:00:00	Patrolling
13:42:00	Travelling
13:48:00	At the scene
15:41:00	Patrolling
15:42:00	Travelling
15:48:00	At the scene

shown in Figure 5.4. Upon initialisation, the agent is deployed to patrol the 5 'hot' streets (red segments) of patrol beat 612 (red-coloured area) and is dispatched to respond to incidents (red dots) throughout the precinct. This behaviour is consistent with the rules implemented for the movement of agents.

Knowledge acquired through a Detroit News article (Hunter, 2015) suggests that DPD generally operate in a similar fashion to forces in England and Wales with regards to dispatching. As such – and in the absence of detailed information regarding DPD's dispatching practices – the simulation was demonstrated to Durham Constabulary's call and dispatch team (UK) to review model assumptions and model outputs for a limited number of cases. They confirmed that the behaviour of the model was consistent with the activities of their police officers.

5.3.3 Population-level qualitative validation (level 1)

With a level-one validation, the model's usefulness is evaluated by the degree of qualitative agreement between simulated and observed population-level patterns. The goal here is to evaluate the level of general realism of dispatch and travel times produced by the model by comparing their values with those observed in DPD's CFS dataset for the same incidents over the same time period.

As previously highlighted in Subsection 5.1.2, dispatch and travel times are likely dependent upon the chosen deployment configuration – i.e. how many agents are deployed and to which patrol beats. As such, when validating the model against DPD's real system, it is important to ensure the model is run on a deployment configuration resembling as closely as possible that

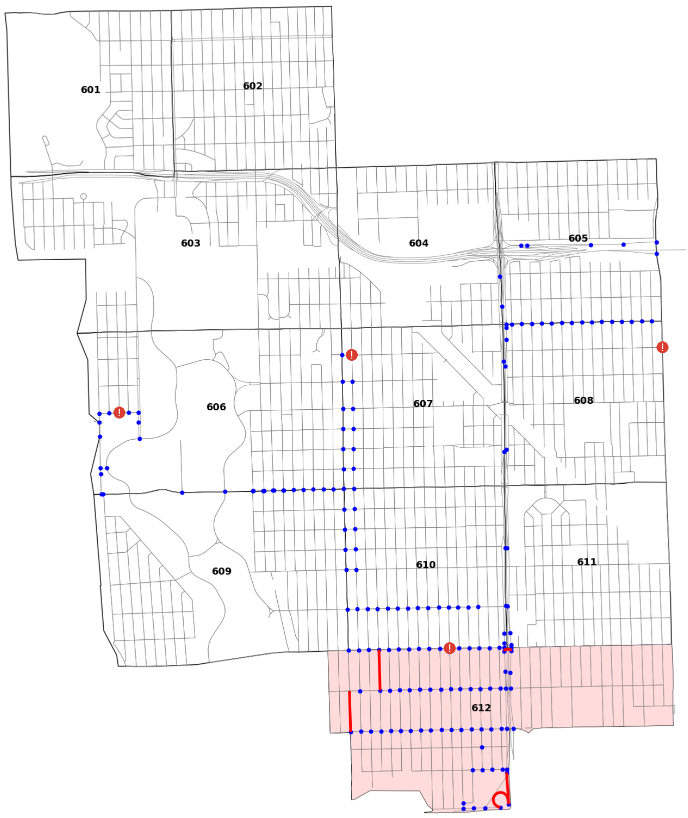

Figure 5.4: Trail of nodes visited by an agent patrolling in patrol beat 612 and responding to incidents in the precinct (precinct 6) throughout the simulated time period (2019-03-19 08:00:00 to 2019-03-19 16:00:00).

which was actually implemented by DPD for the considered time period. This is so that any potential difference in the distribution of simulated and observed dispatch and travel times can be solely attributed to the behaviour of the dispatcher and the agents.

According to the aforementioned Detroit News article (Hunter, 2015), DPD deploys between 2 and 7 squad car units in each precinct and their deployment decisions are based on recent crime trends, suggesting a targeted-patrolling type of behaviour. However, there is no publicly available information that precisely describes the deployment configuration chosen by DPD on a given shift; specifically with regards to the number of agents deployed across the force and their assigned patrol beat.

To cover a range of possible configurations, six configurations were implemented representing the deployment of 10, 20, 30, 40, 50 and 60 agents. To better mimic the deployment configuration implemented by DPD, the agents in these deployments were deployed to targeted patrol beats based on historical crime, in contrast with the random deployments used in the sensitivity analysis in Section 5.2.

The design of these targeted deployments were based on the number of historical crimes that took place in each patrol beat during the 100 time periods that make up each demand scenario's training set (2018). To illustrate, the number of historical crimes across patrol beats for both demand scenarios is shown in Figure 5.5. The spatial patterns of historical crime demand observed in these maps align with results from Section 4.5 of Chapter 4 which identified a high volume of reported crimes in patrol beats around the downtown area. However, the results of Chapter 4 related to the entire dataset of historical crimes spanning three years, while the ones shown here focus on the year 2018 (training set).

The training set (2018) is used here as it corresponds to the year prior to that chosen for the testing set (2019). This aims to represent how targeted deployments are usually made by police officers based on historical demand. First, crimes from time periods in the testing set are used to identify a targeted configuration based on historical demand. Then, the performance of each of those targeted configurations is evaluated using the ABM.

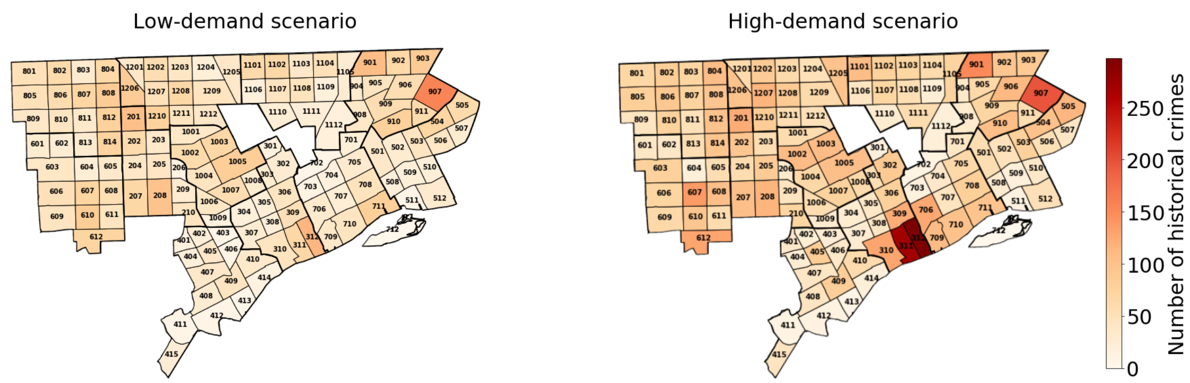

Figure 5.5: Number of historical crimes in the 'training set' time periods (2018) across the patrol beats of DPD. In targeted deployments, the n 'hottest' beats are staffed with an agent (where n is the number of deployed agents in a given configuration).

Having counted the number of historical crimes in each demand scenario's training set (2018) for each patrol beat, all beats were then ranked and the top n beats were staffed with an agent, where n is the number of deployed agents in a given configuration. The resulting targeted deployments configurations are displayed in Figures 5.6 (low-demand scenario) and 5.7 (high-demand scenario). While it may be the case that none of these configurations exactly equate to that implemented by DPD on a given time period, it is nonetheless informative for validation purposes to compare the system's performance for each of these configurations with DPD's performance as observed in the CFS dataset.

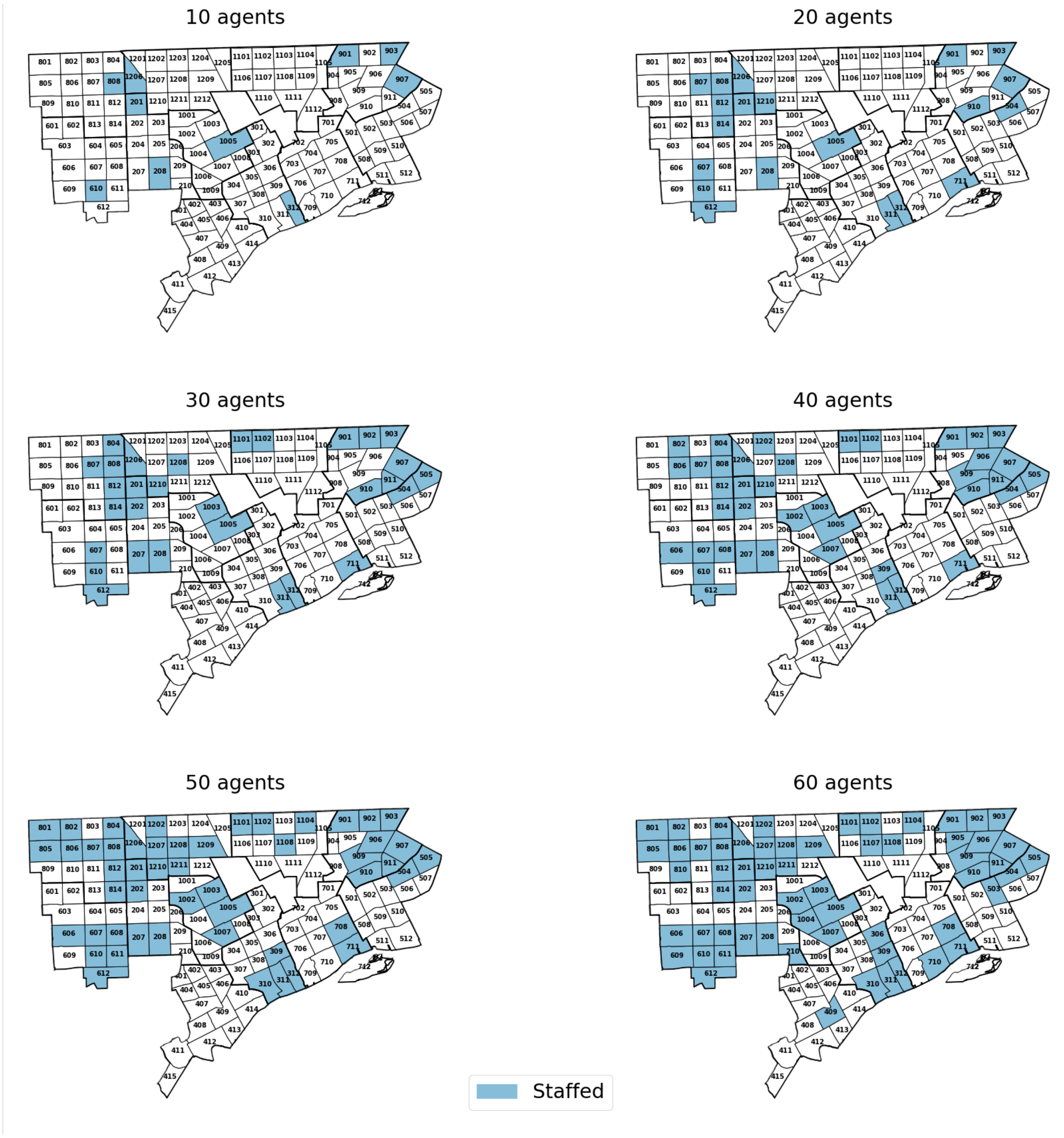

Figure 5.6: Targeted deployment configurations for a given number of deployed agents (based on historical crimes under a low-demand scenario)

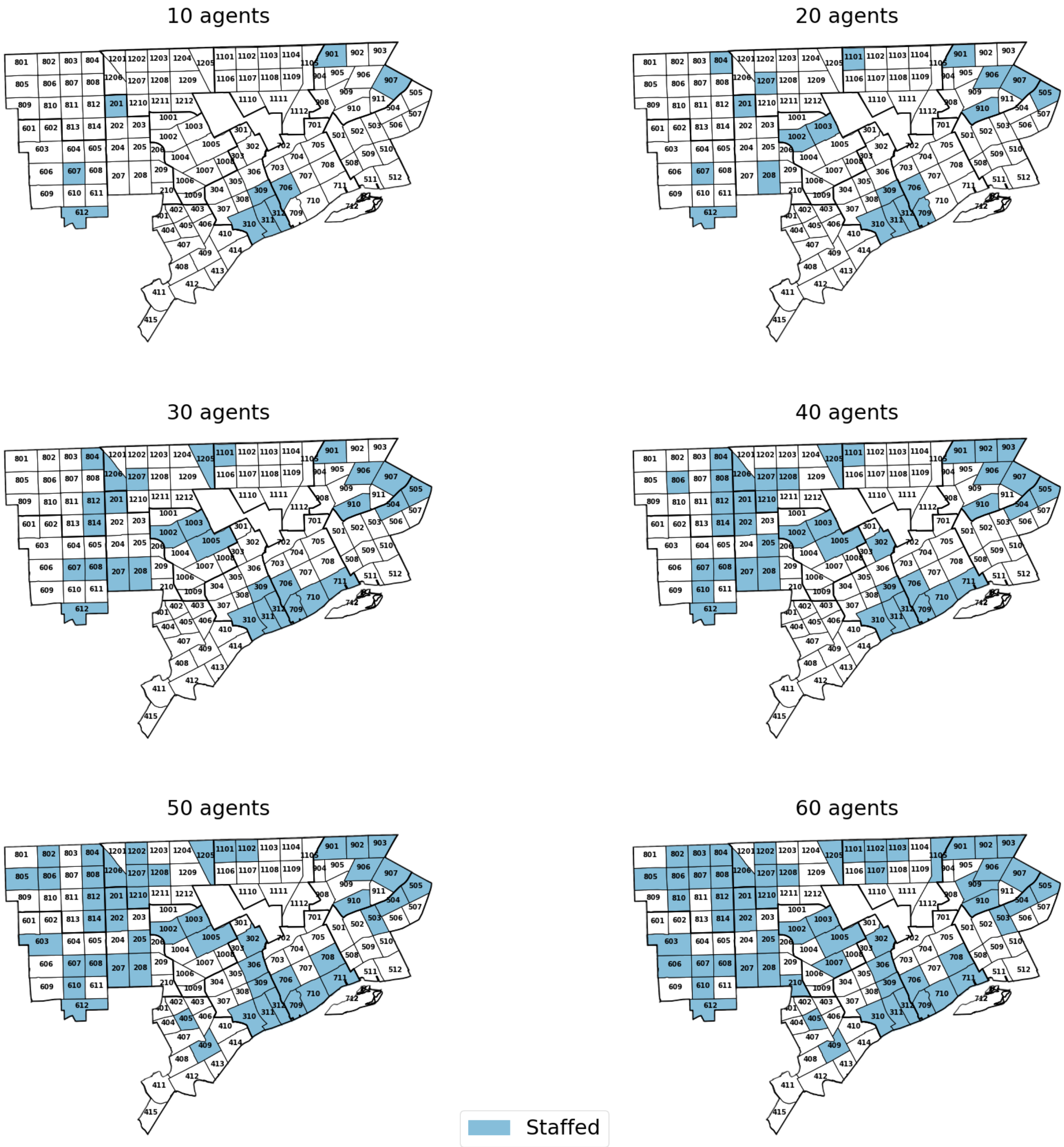

Figure 5.7: Targeted deployment configurations for a given number of deployed agents (based on historical crimes under a high-demand scenario)

For each demand scenario, all six deployment configurations were evaluated, totalling 600 ABM runs for each demand scenario (see Table 5.6 for a summary of the range of values used for model validation).

Validation pattern 1: distribution of dispatch times

The first validation pattern of interest concerns the distributions of observed versus ABM-generated dispatch times. Figure 5.8 shows the distribution of dispatch times observed in

Table 5.6: Values used in the population-level qualitative validation of dispatch and travel times.

Variable	**Number of values**	**Values**
Deployment type	1	Targeted based on historical reported crimes
Demand scenarios	2	Low-demand, high-demand
Time periods	100	From scenario's 'test set' (2019)
Number of agents	6	10, 20, 30, 40, 50, 60

DPD's dataset as well as those generated by the model for the various targeted configurations under the low-demand scenario (left-hand side) and the high-demand scenario (right-hand side).

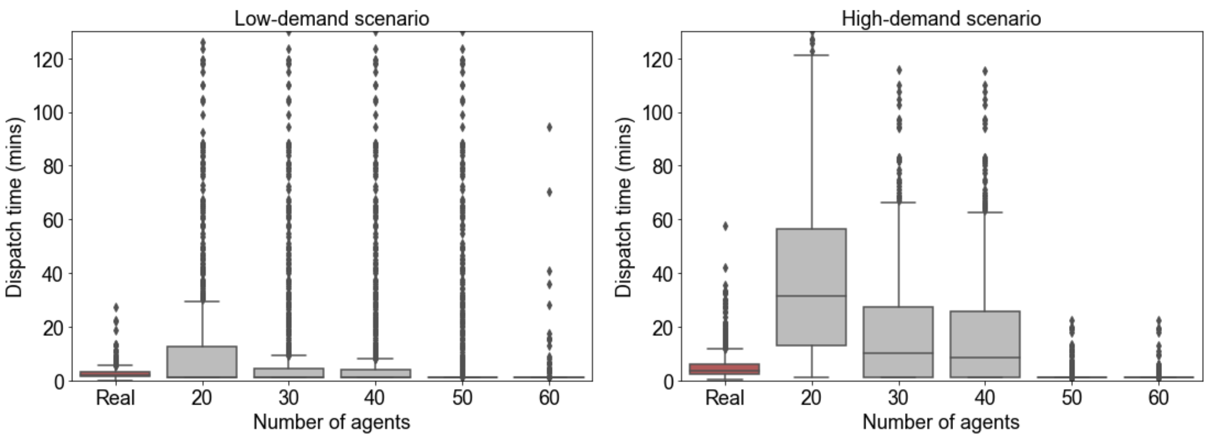

Figure 5.8: Comparison of the distribution of observed (real) versus ABM-generated incident dispatch times.

In the low-demand scenario, the distribution of DPD's observed dispatch times appears to resemble the distribution of dispatch times produced by targeted deployment configurations of 30 and 40 agents. In high-demand time periods however, it is the targeted deployment of 50 agents that appears to best match the distribution of DPD's actual dispatch times. These numbers are in accordance with DPD's statement indicating that they typically deploy between 2 and 7 agents per precinct, hence an average of 50 agents across all 11 precincts. Interestingly, it appears that deploying 40 agents instead of 30 produces very little improvement in dispatch time under both low and high-demand scenarios.

The 10-agent configuration was excluded from the graph as it is too unrealistic and distorted the scale of the y axis. In both scenarios, this deployment appeared to create a considerable backlog of unassigned incidents. With these 10-agent deployments the dispatcher's FIFO behaviour yielded an median dispatch times of 50 minutes in the low-demand scenario and 130 minutes in the high-demand one. Although this deployment configuration is likely too different from DPD's actual one, its outcome makes sense given the low number of agents dispatched. Indeed, dispatch times are expected to increase drastically when resources are stretched – i.e. there are fewer agents deployed than are needed to meet the CFS demand.

In contrast, for configurations with enough agents to service the force, the median generated dispatch time was closer to 1 minute (above 50 agents in both scenarios). In other words, with targeted deployments of 50 agents, there is always at least one available agent in the force to be dispatched immediately to an incoming incident.

Overall, when applied to the specific context of DPD, the model is able to produce realistic looking dispatch times for targeted deployments of 30 agents in a low-demand scenario and 50 agents in a high-demand scenario. These results validate the behaviour of the dispatcher, in particular its FIFO approach to dispatching agents to arising incidents.

Validation pattern 2: distribution of travel times

The second population-level validation pattern relates to the distribution of travel times. Figure 5.9 shows the distribution of observed travel times to incidents as well as that of travel times generated by the model for the various targeted configurations under the low-demand scenario (left-hand side) and the high-demand scenario (right-hand side). The figure shows a general trend by which the more agents are deployed, the quicker they reach the incident they are dispatched to. This pattern is expected, as more agents mean that the 'closest available agent' chosen for dispatch is more likely be in closer proximity to the incident than in a configuration with fewer agents. Interestingly, there is no clear improvement in travel times between a targeted deployment of 10 versus 20 agents under both the low-demand and the high-demand scenario.

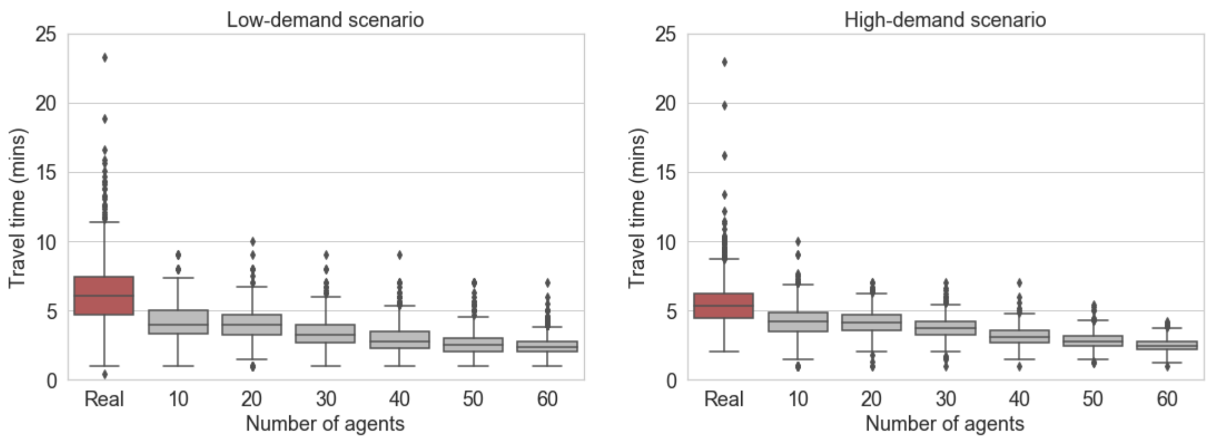

Figure 5.9: Comparison of the distribution of observed (real) versus ABM-generated incident travel times.

Once dispatched to an incident, DPD's response cars appear to take slightly longer to arrive at the scene than the agents in the model do. Furthermore, observed travel times tend to fluctuate more than generated ones. This can be explained by the unpredictable nature of the following real-world factors:

- the speed at which responding vehicles drive: model agents always drive at the maximum speed limit on each road segment whereas real response cars may drive at any speed;
- the flow of traffic: the model assumes a completely fluid traffic but in the real world, traffic congestion may impede the driving speed of response cars;
- the behaviour of responders: model agents start driving towards the incident instantaneously when dispatched whereas it may take some time for real officers to get in their car and start driving to the incident.

Overall, the small differences between observed and generated travel times appear reasonable when considering the unpredictable nature of the real-world system. Furthermore, these small differences are unlikely to prevent the model from fulfilling its purpose in comparing deployment configurations against each other. These results thus validate the travelling behaviour of the agents when responding to incidents, in the context of Detroit.

5.3.4 Summary on model validation

This section attempted to validate the model, in particular the behaviour of the dispatcher and agents. Results from the population-level qualitative validation (level 1) applied to DPD showed that the model appears to be able to broadly reproduce the dispatch and response times observed in DPD's dataset. Indeed, the distribution of generated dispatch and travel times for a realistic deployment configuration (30 or 40 agents in low-demand scenario, 50 agents in high-demand) resembles that observed in the data.

Given the complexity of the real police system, it is impossible for any model – as a necessary simplification of the real world – to be perfectly realistic. Furthermore, the relative lack of data against which to validate the model means that the validation performed in this section can only confirm the realism of the model in a broad sense.

Nonetheless, the purpose of the model is not to compare its outcome to the real police system but instead to compare the outcomes of various deployment configurations against each other.

As such, this section showed that the level of realism of the model is sufficient for its intended purpose. The next section presents the results of a series of simulation experiments for the case study of Detroit conducted using the ABM.

5.4 Simulation experiments for model analysis

Section 5.2 demonstrated that the model is not overly sensitive to parameters and Section 5.3 validated that the dispatcher and agents in the model behave in a sufficiently realistic manner. The ABM developed in this thesis is thus fit for its purpose, that is, to evaluate the performance of the studied police system under particular deployment configurations with the view to comparing configurations. In doing so, the ABM represents a powerful tool in exploring the impact of a given deployment configurations on system performance, ultimately providing a form of evidence-based approach to police deployment decisions making, as discussed in Chapter 2.

As defined in Chapter 2, the PDOP is concerned with identifying where to send available resources to meet both reactive and proactive demand at a minimal cost. This problem can be further broken down into specific questions to be explored such as:

1. What is the performance gain (if any) of deploying additional cars for a particular demand scenario (e.g. a Friday night)?
2. What are the consequences of a given deployment configuration on reactive effectiveness (as measured here by average response time and rate of 'failed' responses) versus proactive effectiveness (as measured here by the total crime deterrence score)?
3. What is the performance gain (if any) of deploying resources based on historical demand instead of randomly?

This section explores ways in which the model can provide answers to the above questions. Through the conducted simulation experiments, it is possible to assess whether and how performance is affected by various deployment configurations.

5.4.1 Experiment setup

The simulation experiments were conducted as per the OFAT principle, in a similar fashion to the sensitivity analysis (see Section 5.2). By turning certain mechanisms on and off, one can gain a better understanding on how model outputs emerge. For a number of agents n (where n takes the sequential values 10, 20, 30, 40, 50, and 60), the outcomes of two deployment configurations were compared: (1) a targeted deployment configuration based on historical CFS demand and (2) a random deployment configuration. The latter was used to provide a benchmark against which to evaluate the benefits, if any, of a targeted deployment.

In the model validation analysis (see Section 5.3 above), the targeted deployments were based on the number of historical reported crimes because the aim was to mimic the design choices made by DPD. In comparison, a different approach is chosen for these experiments, one that uses historical CFS instead of historical crimes to design the targeted deployments. This is so that the performance of the targeted configurations, as evaluated in these experiments, may be directly related and compared to that of the optimum deployment configuration(s) suggested by the single-objective GA in Chapter 7. Indeed, this GA converges towards an optimum configuration based solely on average response times to CFS (which it attempts to minimise). With the GA, a minimal average response time is achieved through a configuration in which agents are placed within short driving time to arising CFS. This equates to deploying agents based on historical CFS demand (assuming that past demand is a good predictor for future demand), as opposed to historical crimes.

The design of targeted deployments for a given number of available agents (n) is based on the number of historical CFS incidents that took place in each patrol beat during the 100 time periods that make up each scenario's training set (2018). To illustrate, Figure 5.10 shows the number of historical CFS incidents in the training set of each demand scenario.

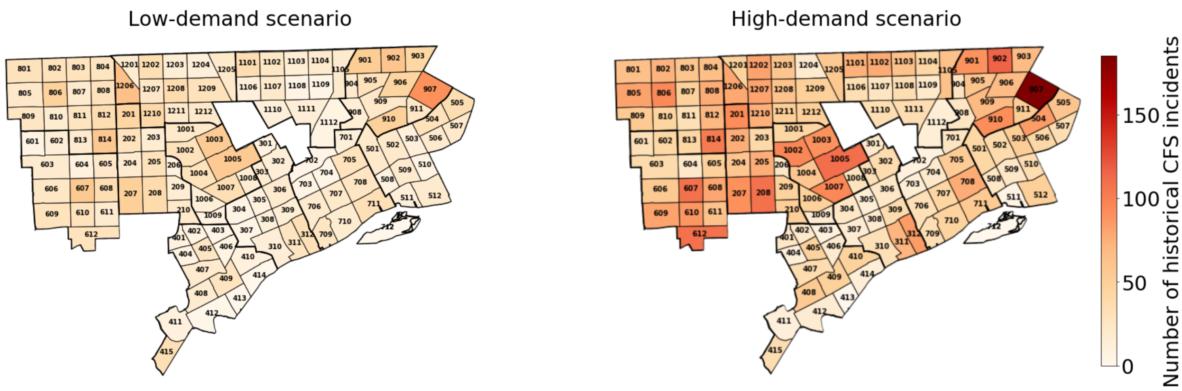

Figure 5.10: Number of historical CFS incidents in the 'training set' time periods (2018) across the patrol beats of DPD. In targeted deployments, the n 'hottest' beats are staffed with an agent (where n is the number of deployed agents in a given configuration).

In a similar fashion to Section 5.3, all patrol beats are ranked based on their number of historical CFS incidents from each scenario's training set and the top n beats are staffed with an agent. The resulting targeted deployments for various numbers of deployed agents are displayed in Figures 5.11 (low-demand scenario) and 5.12 (high-demand scenario).

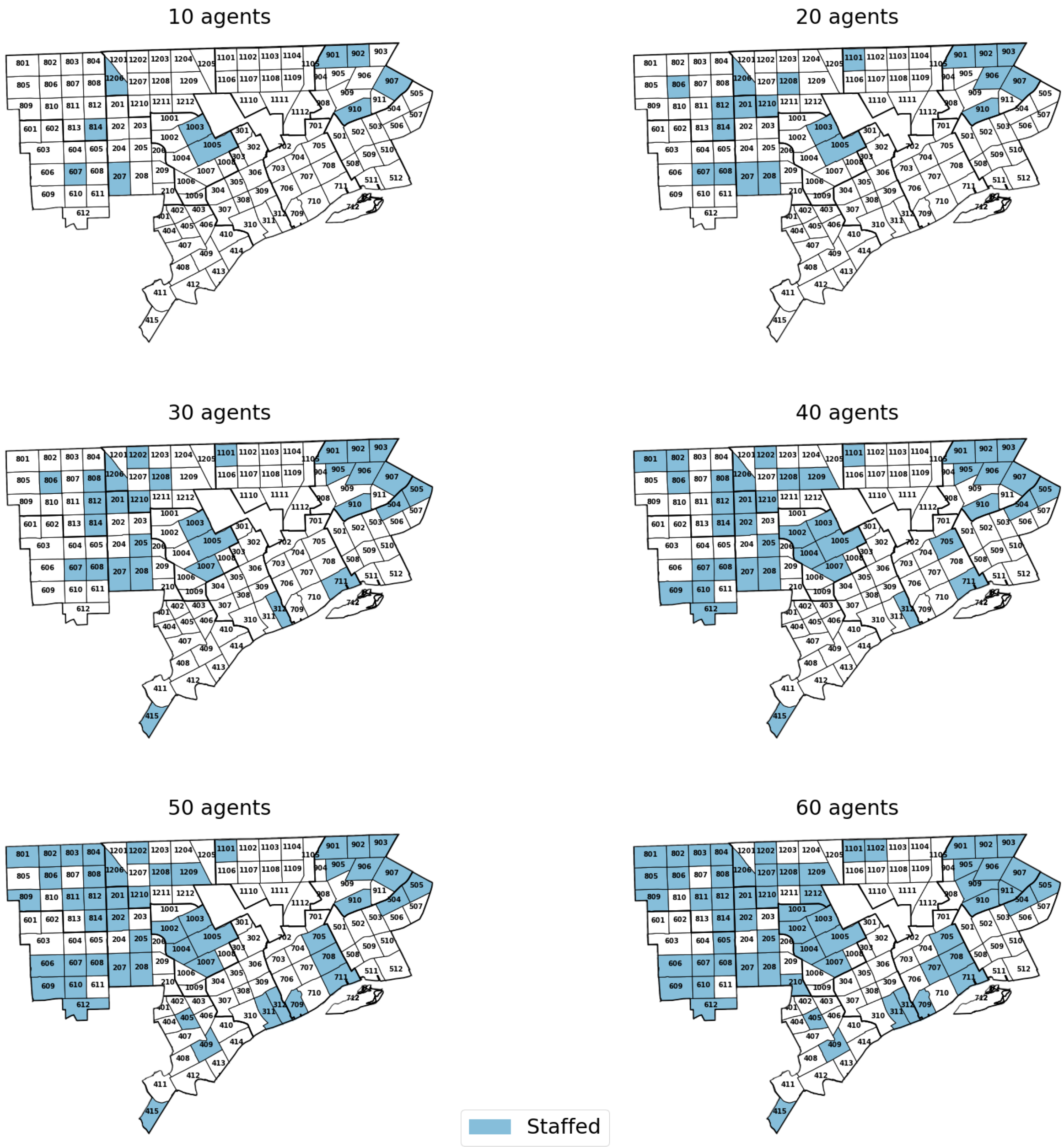

Figure 5.11: Targeted deployment configurations for a given number of deployed agents (based on historical CFS incidents under a low-demand scenario)

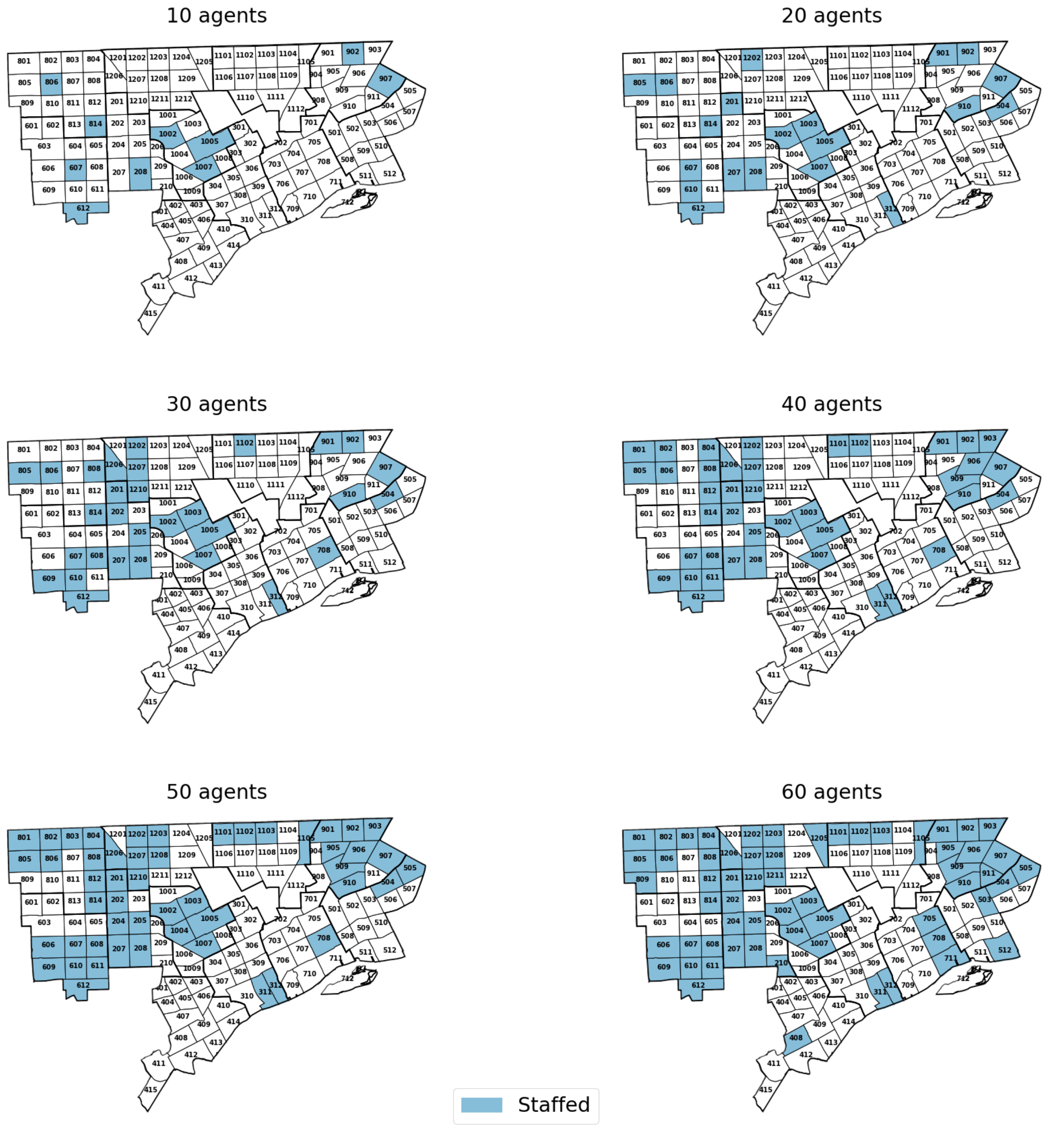

Figure 5.12: Targeted deployment configurations for a given number of deployed agents (based on historical CFS incidents under a high-demand scenario)

As previously mentioned in Chapter 4 the spatial distribution of CFS in Detroit differs from that of reported crimes. In particular, while patrol beats around the downtown area exhibit the highest number of historical reported crimes, they do not generate a high volume of CFS. As a result, the targeted deployments based on historical CFS demand (reactive demand) differ quite significantly from those based on historical crimes (proactive demand) (see Figure 5.5 in Section 5.3 above for comparison).

In contrast to targeted deployment configurations, the random configurations are generated by sampling n random patrol beats to be staffed. A new random configuration is created and evaluated for each run to account for the stochasticity that comes with randomly choosing configurations.

Each configuration (targeted or random, with n agents) was evaluated on the 100 time periods that make up each demand scenario's 'test set' (2019). A total of 1,200 ABM runs were thus executed for each demand scenario (see Table 5.7 for a summary of the range of values used in this simulation experiment). For each configuration, the performance of the system was evaluated on the three performance metrics outputted by the ABM: (1) average response time (2) rate of 'failed' responses and (3) total crime deterrence score.

Table 5.7: Values used in the simulation experiment

Variable	**Number of values**	**Values**
Number of agents	6	10, 20, 30, 40, 50, 60
Deployment types	2	Random, targeted based on historical CFS
Demand scenarios	2	Low-demand, high-demand
Time periods	100	From scenario's 'test set' (2019)

The statistical significance of the null hypothesis (i.e. no significant difference between number of agents or between deployment types for a given number of agents) was determined with a Kruskal-Wallis test at p=0.01 and the effect size of each test was evaluated by calculating the epsilon squared value (refer to Table 4.3 for equivalence between epsilon squared values and effect quality).

5.4.2 Results

The first question explored in these experiments concerns the impact of the total number of deployed agents on system performance (response time, rate of 'failed responses' and crime deterrence score). The second question explored relates to the importance of the placement of these agents – i.e. in which way does deploying a given number of agents to targeted patrol beats affect the system performance compared with a random deployment? The mean and standard deviation for all 3 performance metrics are shown in Table 5.8. In what follows, the impact of deployment configurations on each metric is considered individually.

Table 5.8: Mean and standard deviation of average response time, percentage of ‘failed’ responses and total crime deterrence by number of agents and type of deployment for low and high demand scenarios.

Number of agents	**Deployment type**	**Average response time (mins)**				**Percent. ‘failed’ responses (%)**				**Total deterrence score**			
		Low-demand		**High-demand**		**Low-demand**		**High-demand**		**Low-demand**		**High-demand**	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
10	Random	14.17	8.19	31.47	16.11	8.02	6.04	17.33	7.68	402.82	62.27	725.23	130.68
	Targeted	16.16	9.51	18.88	10.86	10.13	6.41	12.63	6.49	390.52	46.04	718.74	134.32
20	Random	10.67	6.90	19.15	11.50	7.09	6.30	15.38	7.70	837.19	83.37	1554.12	190.69
	Targeted	5.18	1.95	13.35	6.83	1.40	2.66	9.58	5.24	832.69	63.92	1592.37	190.64
30	Random	7.56	5.06	11.86	6.36	4.43	4.94	10.28	6.09	1263.93	110.25	2425.48	231.81
	Targeted	6.67	2.78	11.13	5.60	3.56	3.61	8.59	4.50	1334.55	79.02	2470.95	218.43
40	Random	5.54	3.11	8.58	5.41	1.93	3.08	5.68	5.03	1696.92	109.44	3316.87	217.91
	Targeted	5.64	2.28	7.31	3.49	2.68	3.34	4.60	3.56	1763.77	98.81	3380.65	222.29
50	Random	4.26	1.14	5.74	3.01	0.55	1.66	2.98	3.53	2129.93	131.04	4178.57	262.27
	Targeted	4.30	1.44	5.27	2.18	1.13	2.40	2.49	2.75	2239.68	95.26	4229.28	214.95
60	Random	3.94	1.13	4.54	2.14	0.58	1.94	1.32	2.88	2576.11	140.99	5024.84	309.68
	Targeted	3.74	0.88	4.82	1.99	0.46	1.74	2.04	2.13	2712.32	116.45	5063.92	247.88

NOTE: $n = 100$ per combination of number of agents, deployment type and demand scenario.

Average response time

Figure 5.13 shows the distribution of average response times across number of agents and type of configuration (random versus targeted) under both demand scenarios. The results suggests that, overall, the average response time decreases as the number of deployed agents increases. Furthermore, the higher the number of deployed agents, the more consistent the response times. These trends, which were anticipated, can be observed in both low-demand and high-demand scenarios. In the low-demand scenario, the most striking improvement of the targeted deployment over a random one appears to be achieved with targeted deployments of 20 agents.

A Kruskal-Wallis test was conducted to determine whether the number of deployed agents had an effect on the average response time. The results indicate a significant difference between numbers of agents for both scenarios (low-demand scenario: H = 710.621, p<.01; high-demand scenario: H = 735.261, p<.01). The observed effect size was strong for both scenarios (low-demand scenario: $\epsilon^2 = 0.593$; high-demand scenario $\epsilon^2 = 0.613$).

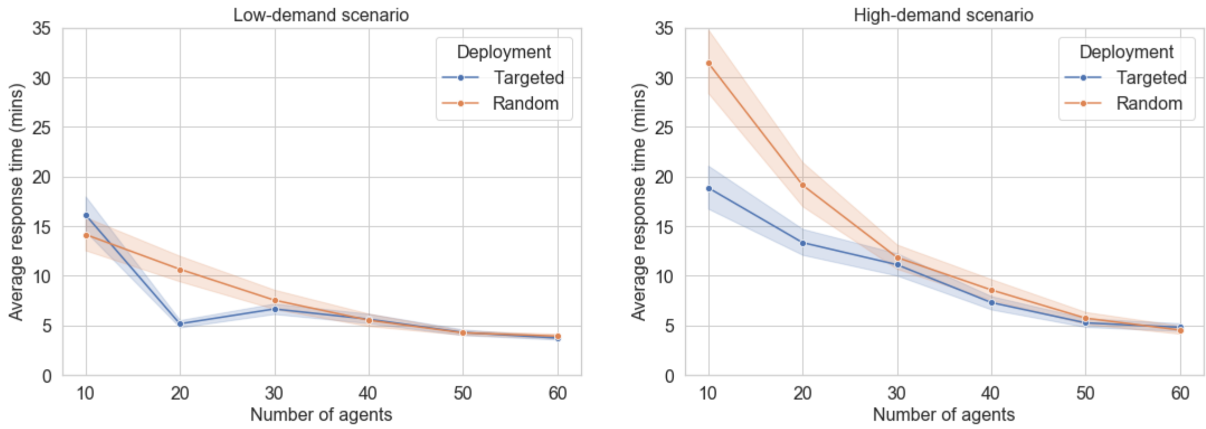

Figure 5.13: Average response time by number of deployed agents for random versus targeted deployments and both low-demand (top row) and high-demand (bottom row) scenarios. The curves show the mean and 95% confidence intervals around the mean (using the bootstrap method).

Additional pairwise Kruskal-Wallis tests were conducted to determine whether, for a given number of deployed agents, the chosen deployment type (i.e. targeted versus random) had an effect on the average response time. Results are displayed in Table 5.9 (low-demand scenarios) and Table 5.10 (high-demand scenarios). In the low-demand scenario, the only noteworthy difference in average response time between targeted and random deployments was observed for configurations of 20 agents. Responses were 51% faster on average with a targeted deployment of 20 agents compared with a random one for the same number of agents. For deployments involving any other number of agents, the targeted deployment did not bring any considerable improvement in response time. In the high-demand scenario, the targeted deployment yielded significantly faster response times for configurations of 10 agents (40% faster responses on average) and 20 agents (30% faster responses on average).

Table 5.9: Mean difference in response time between targeted and random deployment in low-demand scenarios.

Num. of agents	**Mean response time (mins)**			**% change**	**P-value**	**Effect size**
	Random	**Targeted**	**Diff.**			
10	14.17	16.16	+1.99	+14.07 %	0.11	Weak
20	10.67	5.18	**−5.49**	**−51.47 %**	**<0.01**	Strong
30	7.56	6.67	+0.89	−11.79 %	0.62	Negligible
40	5.54	5.64	+0.10	+1.83 %	0.18	Negligible
50	4.26	4.30	+0.04	+0.90 %	<0.01	Weak
60	3.94	3.74	−0.19	−4.93 %	<0.01	Moderate

Note: the values in bold are those showing a sizeable and statistically significant difference in mean response time with a moderate to strong effect size.

Table 5.10: Mean difference in response time between targeted and random deployment in high-demand scenarios.

Num. of agents	**Mean response time (mins)**			**% change**	**P-value**	**Effect size**
	Random	**Targeted**	**Diff.**			
10	31.47	18.88	**−12.59**	**−40.01 %**	**<0.01**	Relatively strong
20	19.15	13.35	**−5.79**	**−30.27 %**	**<0.01**	Moderate
30	11.86	11.13	−0.73	−6.16 %	0.59	Negligible
40	8.58	7.31	−1.27	−14.78 %	0.14	Weak
50	5.74	5.27	−0.47	−8.26 %	0.02	Weak
60	4.54	4.82	+0.28	+6.18 %	0.19	Negligible

Note: the values in bold are those showing a sizeable and statistically significant difference in mean response time with a moderate to strong effect size.

These results suggest that a targeted deployment may be particularly beneficial for reducing the average response time in Detroit when resources are stretched – e.g. under high-demand scenarios when only 10 to 20 agents are deployed. In the high-demand scenarios tested in the model, targeted deployments have the potential to yield responses that are up to 40% faster than with a random equivalent configuration. In deployments where the number of agents is sufficient to meet demand, on the other hand, a targeted deployment does not yield faster responses than a random one.

Percentage of 'failed' responses

Figure 5.14 shows the distribution of the percentage of 'failed' responses across number of agents and type of configuration (random versus targeted) under both demand scenarios. As previously explained, a response is considered 'failed' if its response time is greater than 15 minutes. The figure suggests that, similarly to the related average response time, the percentage of 'failed' responses tends to decrease when the number of deployed agents increases.

Results from the Kruskal-Wallis test indicate a significant difference in percentage of 'failed' responses across numbers of deployed agents (low-demand scenario: H = 424.201, $p<.01$; high-demand scenario: H = 642.880, $p<.01$). The effect size was relatively strong for the low-demand scenario ($\epsilon^2 = 0.354$) and strong for the high-demand scenario ($\epsilon^2 = 0.536$).

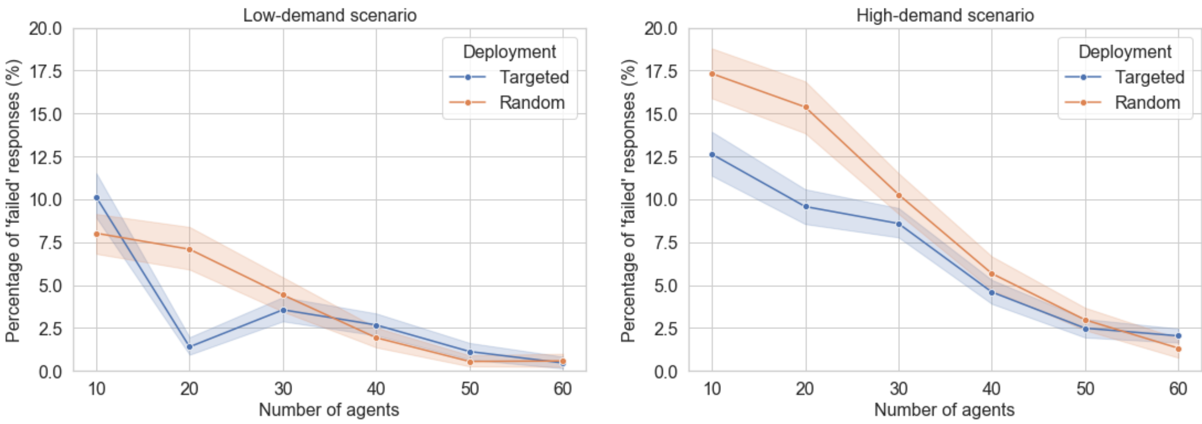

Figure 5.14: Percentage of 'failed' responses by number of deployed agents for random versus targeted deployments and both low-demand (top row) and high-demand (bottom row) scenarios. The curves show the mean and 95% confidence intervals around the mean (using the bootstrap method).

Additional pairwise Kruskal-Wallis tests were conducted to determine whether, for a given number of deployed agents, the chosen deployment type (i.e. targeted versus random) had an effect on the percentage of 'failed' responses. Results are displayed in Table 5.11 (low-demand scenarios) and Table 5.12 (high-demand scenarios).

Results are similar to those observed for the average response time (see above). Here too, the only noteworthy difference in percentage of 'failed' responses between targeted and random deployments under the low-demand scenario was observed for deployments of 20 agents. For this configuration, the number of 'failed' responses dropped by almost 6%. For deployments involving any other number of agents, the targeted deployment did not bring any considerable reduction in percentage of 'failed' responses compared with the random one.

In the high-demand scenario, the targeted deployment lead to significantly fewer 'failed' responses for configurations of 10 agents (5% fewer) and 20 agents (6% fewer). As previously shown for the average response time, these results suggest that a targeted deployment is particularly beneficial for reducing the average response time in Detroit when resources are stretched (high-demand scenarios with 10 to 20 agents).

Table 5.11: Mean difference in percentage of 'failed' responses between targeted and random deployment in low-demand scenarios.

Num. of agents	**Percent. failed responses (%)**			**P-value**	**Effect size**
	Random	**Targeted**	**Diff.**		
10	8.02	10.13	+2.10	0.02	Weak
20	7.09	1.40	**−5.69**	**<0.01**	Relatively strong
30	4.43	3.56	−0.87	0.48	Negligible
40	1.93	2.68	+0.74	0.03	Weak
50	0.55	1.13	+0.58	0.04	Weak
60	0.58	0.46	−0.12	0.64	Negligible

Note: the values in bold are those showing a sizeable and statistically significant difference in mean response time with a moderate to strong effect size.

Table 5.12: Mean difference in percentage of 'failed' responses between targeted and random deployment in high-demand scenarios.

Num. of agents	**Percent. failed responses (%)**			**P-value**	**Effect size**
	Random	**Targeted**	**Diff.**		
10	17.33	12.63	**−4.69**	**<0.01**	Moderate
20	15.38	9.58	**−5.80**	**<0.01**	Moderate
30	10.28	8.59	−1.69	0.08	Weak
40	5.68	4.60	−1.08	0.32	Negligible
50	2.98	2.49	−0.49	0.69	Negligible
60	1.32	2.04	+0.72	<0.01	Moderate

Note: the values in bold are those showing a sizeable and statistically significant difference in mean response time with a moderate to strong effect size.

Crime deterrence score

Figure 5.15 shows the distribution of the total crime deterrence score across number of agents and type of configuration (random versus targeted) under both demand scenarios. The results suggest that, for both demand scenarios, the total crime deterrence increases with the number of deployed agents. Results from the Kruskal-Wallis test indicate a significant difference in crime deterrence across numbers of deployed agents (low-demand scenario: H=1163.284, p<.01; high-demand scenario: H=1162.838, p<.01). The effect size is very strong for both scenarios (low-demand scenario: ϵ^2 =0.970; high-demand scenario: ϵ^2 =0.970).

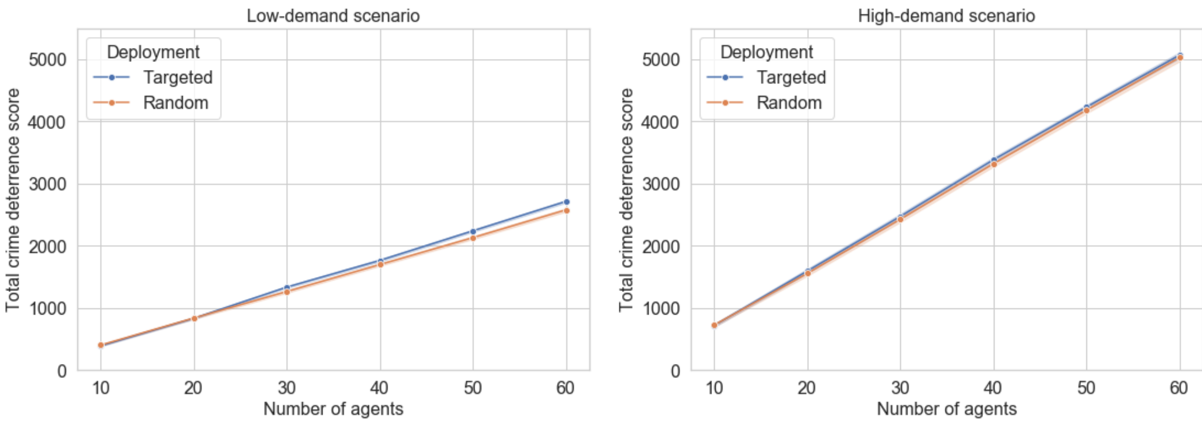

Figure 5.15: Total crime deterrence score by number of deployed agents for random versus targeted deployments and both low-demand (top row) and high-demand (bottom row) scenarios.

The observed increase in crime deterrence when the number of agents increases for a given demand scenario is directly linked to the increase in agent patrolling time, as shown in Figure 5.16. As more agents are deployed, each agent is less likely to be dispatched to incidents and thus may spend more time patrolling their designated beat. As previously explained, the patrolling behaviour presently employed in this iteration of the model is merely the direct outcome of agents' idle time. Subsequent versions may enhance this behaviour to generate more realistic evaluations of the crime deterrence carried out by agents during their patrols.

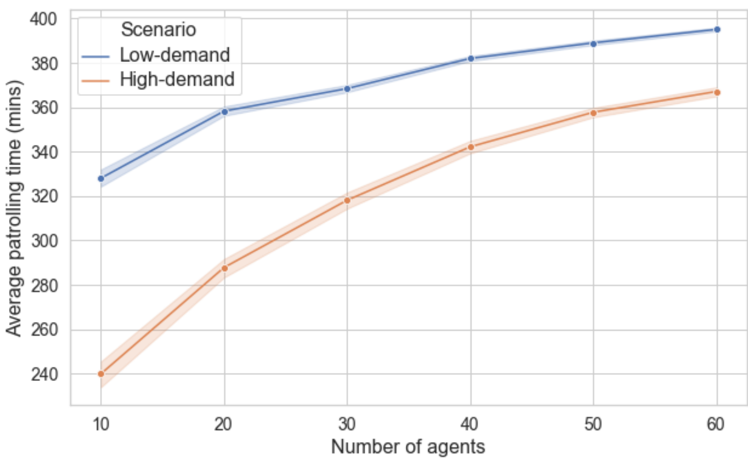

Figure 5.16: Average agent patrolling time under a low-demand and high-demand scenario. The curves show the mean and 95% confidence intervals around the mean (using the bootstrap method).

A noteworthy result here relates to the fact that the total deterrence score is greater in high-demand scenarios than it is in low-demand ones. This is because, although agents spend less time patrolling in a high-demand scenario than in a low-demand one, their preventative potential is greater according to the deterrence score defined in this thesis. Indeed, the number of historical crimes on the streets patrolled by agents is greater in high-demand scenarios (e.g. Saturday 00:00 to 08:00) than in low-demand ones (e.g. Monday 08:00 to 16:00).

Additional pairwise Kruskal-Wallis tests were conducted to determine whether, for a given number of deployed agents, the chosen deployment type (i.e. targeted versus random) had an effect on the total deterrence score. Results are displayed in Table 5.13 (low-demand scenarios) and Table 5.14 (high-demand scenarios).

Table 5.13: Mean difference in total deterrence score between targeted and random deployment in low-demand scenarios.

Num. of agents	**Total deterrence score**			**% change**	**P-value**	**Effect size**
	Random	**Targeted**	**Diff.**			
10	402.82	390.52	−12.30	−3.05 %	0.26	Negligible
20	837.19	832.69	−4.50	−0.54 %	0.81	Negligible
30	1263.93	1334.55	**+70.62**	**+5.59 %**	**<0.01**	Moderate
40	1696.92	1763.77	**+66.85**	**+3.94 %**	**<0.01**	Moderate
50	2129.93	2239.68	**+109.75**	**+5.15 %**	**<0.01**	Relatively strong
60	2576.11	2712.32	**+136.21**	**+5.29 %**	**<0.01**	Relatively strong

Note: the values in bold are those showing a sizeable and statistically significant difference in mean response time with a moderate to strong effect size.

Table 5.14: Mean difference in total deterrence score between targeted and random deployment in high-demand scenarios.

Num. of agents	**Total deterrence score**			**% change**	**P-value**	**Effect size**
	Random	**Targeted**	**Diff.**			
10	725.23	718.74	−6.49	−0.89 %	0.63	Negligible
20	1554.12	1592.37	+38.25	+2.46 %	0.09	Weak
30	2425.48	2470.95	+45.47	+1.87 %	0.17	Negligible
40	3316.87	3380.65	+63.78	+1.92 %	0.05	Weak
50	4178.57	4229.28	+50.71	+1.21 %	0.14	Weak
60	5024.84	5063.92	+39.08	+0.78 %	0.42	Negligible

Note: the values in bold are those showing a sizeable and statistically significant difference in mean response time with a moderate to strong effect size.

Under the low-demand scenarios tested in the model, a significant difference in crime deterrence score between targeted and random deployments was observed with deployments of 30, 40, 50 and 60 agents. In these scenarios, the targeted deployment yielded 4 to 5% more crime deterrence than random ones. Under the high-demand scenarios however, there was no noteworthy difference between the two types of deployment.

These results suggest that a targeted deployment is particularly beneficial for increasing the time that agents are able to spend on patrol in low-demand scenarios (e.g. weekday day shift) and when supply is high (30 agents or more). However, the benefits of a targeted deployment on agent idle time appear to be less tangible when the demand is high (e.g. weekend late shift) and/or when supply is stretched, presumably because agents spend more time responding to incidents in such situations. It is worth reminding the reader that the targeted deployment configurations evaluated here are based on the number of historical CFS incidents in beats, instead of the number of historical crimes in those beats. Higher crime deterrence scores are expected to be achieved with targeted deployments based on historical crime. However, as previously mentioned in this chapter, it is the targeted configurations based on historical CFS that are studied in this experiment, so that they can be compared with those identified by the GA in Chapter 7.

Summary

Overall the number of deployed agents has a significant impact on the performance of the system, both in terms of proactive and reactive response. In the scenarios tested in the model, increasing the number of deployed agents yielded (1) a reduction in average response time, (2) a smaller percentage of 'failed' responses and (3) more 'idle' time for agents to patrol hot streets, thus deterring more crime.

Additionally, a deployment type which targets patrol beats based on historical CFS demand may bring benefits compared to a random deployment. When resources are stretched, the targeted deployment implemented in the model yields a better average response time and a smaller percentage of 'failed' responses. On the other hand, when supply is sufficient to meet CFS demand, a targeted deployment does not bring significant improvements in reactive effectiveness, yet may lead to more crime deterrence as agents have more 'idle' time to patrol.

5.5 Summary: model analysis, validation and simulation experiments

This chapter proposed a series of analyses and experiments using the ABM built in this thesis applied to the exemplar police force of DPD. Section 5.1 began by introducing the manner in which various levels of demand and supply were tested in the model. With regards to demand, the CFS data for DPD was split into a training and test set for two distinct demand scenarios (low versus high). In terms of supply, various deployment configurations were tested throughout the chapter featuring different numbers of agents positioned either randomly or in a targeted fashion based on historical demand. Then, a sensitivity analysis was proposed in Section 5.2, followed by a validation of the ABM in Section 5.3. The purpose of the model is to explore the potential impact of various deployment configurations on system performance. As such, a series of simulation experiments were conducted on DPD's system in Section 5.4 to answer questions concerning (1) the total number of agents deployed and (2) the specific positioning of these agents across the patrol beats of the force.

The ABM built in this this thesis constitutes a safe environment in which various deployment configurations can be explored and their potential real-world consequences may be anticipated. Simulation experiments such as those conducted in this chapter can thus provide policy makers with insights regarding deployment decisions. For instance, such experiments can help quantify the performance gain or loss of deploying an additional car on a particular shift, or that of deploying agents to particular patrol beats.

One particular question of relevance to police deployment decisions concerns identifying the optimal configuration for a particular shift, which is an optimisation problem (see the definition of PDOP in Chapter 2). For instance, the police may wish to find the deployment configuration that minimises average response time and/or maximises crime deterrence through patrolling. Although very informative, the simulation experiments conducted using the ABM in this chapter are unable to answer optimisation questions on their own as they can only be conducted on a small number of configurations, e.g. 10, 20, 30, 40, 50, 60 agents with a small number of different deployment types (e.g. random versus targeted). To answer such an optimisation question, the search for solutions needs to be automated to cover a wide range of possible configurations in order to ensure finding the optimal one. The next chapter will describe how genetic algorithms can help automate the search for solutions to the PDOP.

Chapter 6

Designing Genetic Algorithms for the PDOP

6.1 Introduction

The ABM developed in this thesis simulates the activities of patrols across the force with the view to evaluating the impact of a given deployment configuration (model input) on system performance (model outcome). The goal of the ABM is to evaluate multiple deployment configurations in order to identify optimal solutions to the PDOP. Chapter 5 explored ways in which this ABM tool can be utilised to better estimate the performance of various deployment configurations, using the exemplar police force of Detroit. Although insightful, these ABM experiments are somewhat limited in their ability to answer the PDOP as there are too many configurations to be tested. In itself, the ABM is a descriptive tool rather than a prescriptive one.

This chapter brings forth the value of the methodology chosen here to explore the PDOP: a simulation-based optimisation approach using the ABM as an evaluating tool within a Genetic Algorithm (GA) which automates the search for solutions. Specifically, this chapter focuses on detailing the general design of the GA framework developed in this thesis, while the next chapter will present results obtained from applying this framework to the case study of Detroit. The code for the GAs is available at https://github.com/mednche/police-deployment-optimisation/src/GA.

First, Section 6.2 introduces the need to combine the ABM with a metaheuristic search algorithm that is able to scan the parameter space in an efficient manner. In the specific simulation-based optimisation approach explored in this thesis, the search for optimal solutions to the patrol deployment problem is guided by a GA. After presenting the key concepts of GAs as well as their advantages and limitations in Section 6.3, Section 6.4 presents the design decisions that were made in this thesis to build GAs for the PDOP. Finally, Section 6.5 provides an overview of the logistical decisions made to run the GAs including ways to monitor model performance and prevent over-fitting.

6.2 The need for metaheuristics

The concept of search space in optimisation problems was introduced Chapter 2. The size of a problem is grounded in (1) the dimensionality of the problem at hand (i.e. the number of parameters) and (2) the number of different values for the problem parameters (Eiben and Smith, 2015). Depending on the size of the chosen police force, the search space for the PDOP may become very large. In the case study of Detroit, the force is composed of 131 patrol beats which can each take up one of two values (0: unstaffed or 1: staffed). As such, there are a total of 2^{131} possible solutions to the PDOP in Detroit.

Roughly speaking, a problem is considered easy if there exists a fast solver for it, and hard otherwise. The running-time of an algorithm is the number of elementary steps, or operations, it takes to terminate. Assuming it takes one minute to run a single ABM for a candidate solution, finding the exact optimal solution to the PDOP in Detroit would take approximately $1.4e^{45}$ years. It is thus clearly impossible to exhaustively explore all deployment configurations when searching for solutions to the PDOP in the context of Detroit.

In such a scenario, metaheuristic search optimisation algorithms prove to be an efficient tool to provide usable near-optimal solutions in a short amount of time. Although metaheuristic algorithms are not mathematically guaranteed to find the optimal solution, they are able to scan the search space in an efficient manner, trading completeness for speed. They provide an approximate solution where classical optimisation techniques fail to find any exact solution due to their iterative approach. The following section will introduce genetic algorithms and outline their their main components.

6.3 Genetic Algorithms

The approach chosen in this thesis uses a specific type of metaheuristics called Genetic Algorithms (GAs). First developed by John Holland and David E. Goldberg in the 1970s (Goldberg and Holland, 1979; Holland, 1975), the GA is a search optimisation technique inspired by the principles of genetics and natural selection. While this section provides a generic introduction to GAs, further details about how the concept translates to solving the PDOP will be provided in the following section (Section 6.4).

This section begins with a brief introduction to the process of natural evolution and how GAs relate this phenomenon to the solving of optimisation problems. Then, a detailed description of the key components of GAs is provided. Finally, this section ends with a list of the advantages and limitations of GAs as a search optimisation technique.

6.3.1 Natural selection and GAs

The process of natural evolution

The concept of natural evolution explains how an individual's ability to survive in its environment (fitness) is determined by both its phenotype (traits) and genotype (DNA). An offspring inherits traits from both parents as well as new traits that the parents may not have, due to mutation and cross-overs in their own genotype. These traits may increase an offspring's fitness, yielding a higher probability of survival and passing the traits on to the next generation. Over time, the best individuals survive and reproduce and so evolution progresses, with the average fitness of the population improving.

Each individual is a dual entity: its phenotypic (macroscopic) properties are encoded at the genotypic (microscopic) level. At the macroscopic level, individuals exhibit a combination of behavioural and physical features called phenotypic traits (e.g. big ears, white fur etc.) which determine their fitness by directly affecting their response to the environment. During the reproduction stage, small random variations in phenotypic traits lead to the production of new combinations of traits.

At the genotypic (microscopic) level, natural evolution is governed by molecular genetics. Genes are arranged in several chromosomes. During the reproduction stage, these genes may be altered by two independent random phenomena: (1) small localised mutations and (2) cross-overs (or recombinations) whereby whole sections of DNA are swapped at once. These alterations of the genes result in an alteration in the phenotype, which gets evaluated by the environment as part of the 'survival of the fittest' process.

Collectively, parent selection and gene alteration are the two driving forces of natural evolution. New individuals are created which exhibit some of their parents' traits as well as new traits, and their fitness is evaluated by the environment.

The metaphor of GAs

The fundamental metaphor of GAs relates this powerful natural evolution to a particular style of problem solving – that of trial-and-error (Eiben and Smith, 2015), also known as 'generate-and-test'. In a GA, an analogy is made between an individual and a candidate solution. Each solution is composed of a set of model parameters (or features) that equates to an individual's genes. Unlike in nature, GAs typically represent all the genes of a given individual on a single chromosome. The evaluation function provides a heuristic estimate of solution quality, and the search process is driven by selection and variation operators. Table 6.1 provides a summary of the equivalence of terms between natural evolution and GAs. A concrete example, applied to the PDOP, will be later provided in Section 6.4 of this chapter.

Table 6.1: Equivalence between genetic algorithms and natural evolution

Natural evolution	**Genetic algorithms**
Evolution	Trial-and-error problem solving
Individual	Candidate solution
Gene	Solution parameter
Fitness	Solution quality

6.3.2 Components of a GA

Generally speaking, a GA features a population of individuals (i.e. the candidate solutions), all of which are evaluated at each generation through the evaluation function and assigned a fitness value. 'Fitter' individuals are given a higher chance to mate and seed the next generation. During the mating phase, random crossovers and mutations give rise to new individuals (offspring) to be tested. Thus, as the generations pass, there is a change in the constitution of the population. The individuals in the population become fitter and fitter until a stopping criterion is reached. The flowchart presented in Figure 6.1 provides a summary of the general scheme of a GA, which is now described in details.

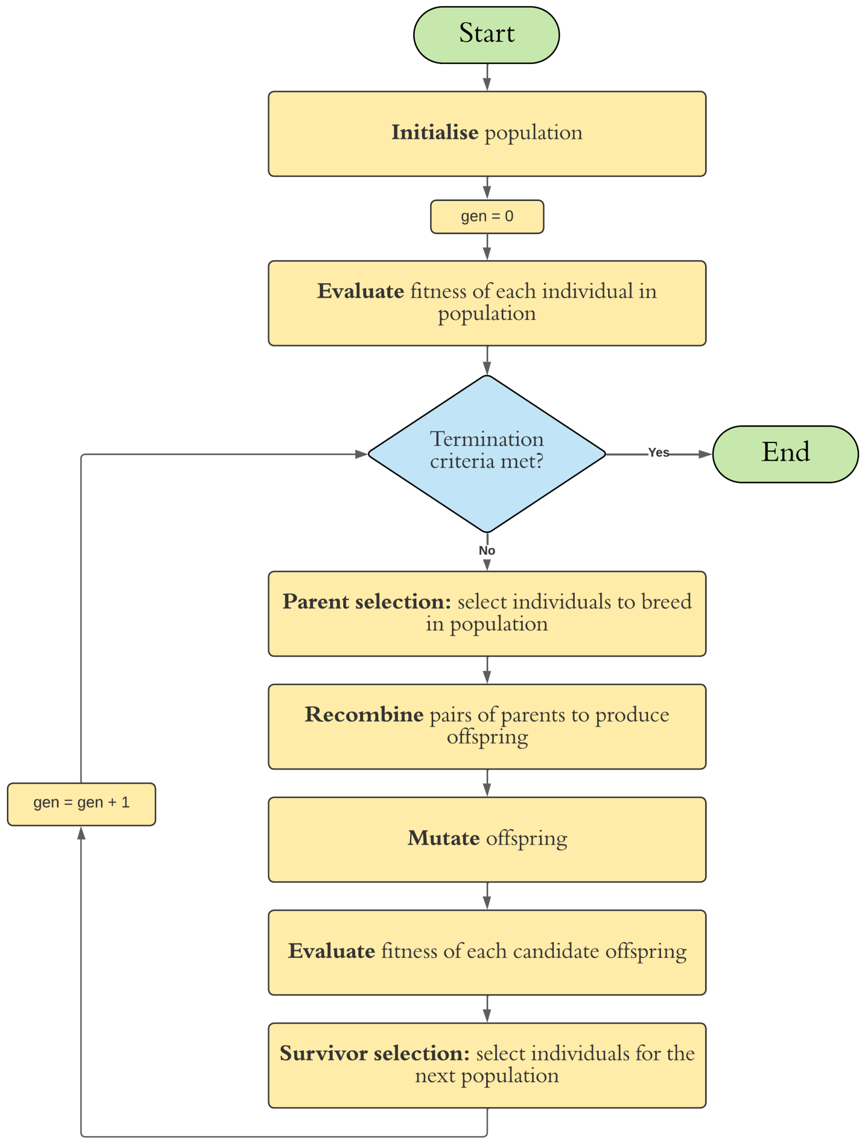

Figure 6.1: Flowchart of the general scheme of a GA.

Population

In a GA, the population is composed of individuals representing possible candidate solutions. As previously explained in Chapter 2, in the case of the PDOP specifically, candidate solutions represent deployment configurations (i.e. which beats are staffed by a patrol unit).

Upon initialisation, the population is seeded with μ randomly generated individuals, and this size typically remains constant throughout the generations. Population diversity is measured by the number of different individuals it contains. Technically speaking, it is the population itself that changes and adapts over time, rather than the individuals which are static objects. In other words, while individuals form the units of selection, the population forms the unit of evolution.

Evaluation function

The role of the evaluation function is to calculate an individual's fitness – i.e, the quality of a solution – based on the objective function(s) of the problem. The resulting fitness determines an individual's chance of survival and of multiplying. The evaluation function is commonly called the fitness function in GAs. This sometimes causes a counter-intuitive terminology when the original problem requires minimisation (e.g. average response time), because the term fitness is usually associated with maximisation.

Parent selection

The role of the parent selection mechanism is to choose individuals to seed the next generation based on their fitness. This, together with the survivor selection mechanism (see below), is responsible for pushing quality improvements in the population. Parent selection is typically probabilistic, thus high-quality individuals have more chance of becoming parents than those with low quality. Nevertheless, low-quality individuals are still given a small chance to breed in order to prevent the search from becoming too greedy and thus getting stuck in a local fitness optimum.

Variations (mutations and crossovers)

The role of variation operators is to create new individuals from old ones. From the generate-and-test search perspective, variation operators perform the 'generate' step. Similarly to the process of natural evolution, there are two types of variation operators: crossovers and mutations.

Crossovers are applied between two or more parents, producing one or more offspring. Crossovers are stochastic: the choices of what parts of each parent are combined, and how this is done, depend on random decisions. This is the primary mechanism by which solutions evolve. With crossovers, offspring inherit a number of genes from each parent. Mutations may then be applied to an offspring to create a slightly modified mutant. Like crossovers, a mutation operator is always stochastic: its output depends on the outcomes of a series of random choices.

Population replacement and survivor selection

A cycle of population replacement unfolds as follows. Each generation begins with a population of size μ, from which a mating pool of parents is selected. Every member of the pool is a copy of an individual in the population, but more copies of the 'better' parents tend to be more represented, as the process is probabilistic. Next, λ offspring are created from the mating pool by the application of variation operators, and are evaluated. Amongst the offspring there may be copies of some parent individuals that survived crossover and mutation without being modified (depending on the probabilities of mutation and crossover chosen for the GA).

The survivor selection mechanism – often called the replacement strategy – takes place after the evaluation of the offspring generated from the selected parents. At the end of each model generation, the whole population is replaced by a new population of μ individuals. As such, a choice needs to be made with regards to which individuals are allowed in to the next generation. In contrast to parent selection, which is typically stochastic, survivor selection is often deterministic.

Replacement strategies can be categorised according to whether they discriminate on the basis of the fitness or the age of individuals. A wide number of strategies based on fitness or age have been proposed for choosing which μ of the (μ parents and λ offspring) should go forward to the next generation. Amongst them, the most common strategies are:

- (μ, λ) selection: when $\lambda \geq \mu$, the λ offspring undergo a fitness-based selection and the μ fittest individuals are chosen for the next population. When $\lambda = \mu$, all the offspring replace the parents without selection.
- $(\mu + \lambda)$ selection: the λ offspring and μ parents are merged and ranked according to their fitness, then the top μ are kept to form the next generation. This scheme prevents accidentally losing some of the best individuals through the generations.

Termination criteria

The GA process is iterated over several generational steps until a termination criterion is met. Typical termination criteria include:

- the maximally allowed CPU time elapses;
- the fixed number of generational steps is reached;

- the fitness improvement remains under a threshold value for a given period of time (i.e., for a number of generational steps);
- the population diversity drops under a given threshold.

6.3.3 Advantages and limitations of GAs

Much like any computational method, GAs present both advantages and limitations, which are here summarised in Table 6.2. GAs are computationally expensive for some problems due to the repeated calculation of fitness values. As such, they are not best suited for simple problems. Furthermore, GA hyper-parameters (e.g. population size, mutation rate etc.) are notoriously difficult to tune (Eiben and Smith, 2015) and the stochastic nature of metaheuristics means there is no guarantee of the quality of the solution(s) as they may not always reach optimality.

Table 6.2: Advantages and limitations of GAs

Category	**Advantages**	**Limitations**
Implementation	• Concept easy to understand • Parallel distribution possible • Works with ABM	• Difficult to tune hyper-parameters
Exploration of the search space	• Efficient search • Balanced exploration versus exploitation	• Computational cost of repeat fitness calculation • No guarantee of optimality

Despite these limitations, GAs provide a number of advantages compared with other techniques. First, they offer considerable flexibility in their design and their concept is intuitive and easy to grasp. Additionally, they can be combined with other techniques such as ABM (see for instance Choi et al., 2001; Stonedahl and Wilensky, 2011) which is the chosen approach in this thesis.

Second, GAs are particularly efficient in searching complex and large search spaces (Choi et al., 2001; Srinivas and Deb, 1994, see for instance), allowing them to quickly rule out large parts of the search space and identify solutions in reasonable time. As such, they are amongst the best suited techniques for complex nonlinear optimisation problems (Mangla et al., 2021, see e.g.), as well as combinatorial optimisation problems (Stonedahl and Wilensky, 2011, see e.g.). Additionally, GAs are parallel in nature: the fitness of each individual in the population can be evaluated independently. as such, advances in modern computing – e.g. multiprocessing techniques – can be harnessed to considerably reduce computing time.

Third, GAs are noted for their robust solution quality when searching large search spaces. As defined in Chapter 2, the PDOP is a multimodal problem which also features multiple objectives. A multimodal problem has multiple solutions or multiple optima, which makes the optimisation problem more complex. Multi-objective optimisation problems involve the optimisation of multiple objective functions, and there may be multiple optimal solutions representing trade-offs between the objectives. When a problem is both multimodal and multi-objective, finding the set of Pareto-optimal solutions can be challenging, and special techniques such as GAs are often employed (Deb, 2001; Goldberg, 1989; Holland, 1992; Michalewicz, 1996; Pham and Karaboga, 2000; Srinivas and Deb, 1994), in order to locate the global optimum or to identify a number of high-fitness solutions corresponding to various local optima. This is because their search strategy based on random choice allows them to balance exploration of the feasible domain and exploitation of 'good' solutions.

6.3.4 Summary: Genetic Algorithms

This section presented the generic framework that forms the common basis for GAs. In broad terms, this framework always involves a population of candidate solutions that are manipulated by selection, re-combination, and mutation operators. Although the general principles of GAs are common to all GA applications, there is no generic GA, and the user has to custom-design the algorithm for the problem at hand. The next section provides a detailed description of the design decisions that were made for the GAs implemented in this thesis to be specifically applied to the PDOP.

6.4 Designing GAs for the PDOP

This thesis explores two variants of the PDOP, each requiring a custom-designed GA. In the first variant of the problem, there is only one objective: to minimise the average incident response time. With this GA variant, there is no optimisation of the number of agents itself. As such, the single-objective PDOP aims to optimise the response time given a current level of staffing.

In the second GA variant, multiple objectives are considered simultaneously (e.g. average response time, percentage of 'failed' responses, total deterrence score) and the GA seeks to identify a number of 'good' solutions that provide a trade-off between these objectives (see Chapter 7 for details). Both GA variants built in this thesis share the same foundational design, varying only on some aspects related to the specific objective(s) of the problem. As such, the remainder of this section details the design decisions and parameter values that are shared across both GAs. The specific differences will be detailed in the next chapter (Chapter 7).

6.4.1 Problem representation

As previously explained in Chapter 2, the approach chosen in this thesis is that of a simulation-based optimisation in which a GA utilises the ABM as a tool to evaluate the fitness of its individuals. At each generational step, the GA is responsible for (1) generating new individuals, (2) evaluating the fitness of individuals using the ABM, (3) selecting parents based on their fitness, (4) applying variation operators to create a population of offspring and (5) selecting survivors among the offspring to create the new population.

Since the GA relies on the ABM for fitness evaluation, the individuals need to be encoded in the GA in the same manner they are inside the ABM itself. As such, the individuals, which here represent deployment configurations are encoded as array of n binary values (where n is the number of patrol beats in the police force). As explained for the ABM in Chapter 3, a binary values in the array indicates whether the corresponding patrol beat is staffed with an agent (1) or not (0).

The fitness of each individual is calculated using a fitness function which is composed of the objective function(s) of the problem. Fitness functions are thus typically problem specific and as such, they differ between the two GA variations implemented in this thesis. For instance, the single-objective GA only utilises the average response time outputed by the ABM to calculate fitness. The fitness function of the multi-objective GA, on the other hand, combines multiple performance weighted metrics. Details on how fitnesses are calculated in these fitness functions are provided in the next chapter (see Chapter 7). A summary of the representation of the PDOP chosen in the GAs developed in this thesis is provided in Table 6.3.

Table 6.3: Summary of the GA representation of the PDOP.

GA term	**Police term**	**Encoding**
Individual	Deployment configuration	Vector
Feature	Staffing indicator	Binary
Fitness	Performance metric(s)	Vector

6.4.2 GA parameters

The GAs were built with the DEAP Python library (Fortin et al., 2012) and were run using multiprocessing on ARC4, part of the High Performance Computing (HPC) facilities at the University of Leeds, UK. This HPC facility offers 40 CPU nodes with a maximum available CPU run time of 48 hours. Although the design of a GA is viewed by some as an optimisation problem itself (Eiben and Smith, 2015), the values of the hyper-parameters were not tuned in this early version of the algorithm. Instead, much of the design decisions were inspired by recommendations in the literature (Eiben and Smith, 2015, in particular), or constrained by the computational limitations of the platform used to conduct the research.

Population

According to the literature, the population size μ in GAs tends to vary between 1 and 100 individuals, depending on the computational resources available (Eiben and Smith, 2015). Here, a population of size $\mu = 40$ was chosen in order to match the number of nodes available on the HPC facility (40 nodes). Upon seeding the initial population, each generated individual represents a deployment configuration with a random number of agents. In the case of Detroit, this number of agents is randomly sampled between 1 and 60 for each individual as per information gathered in the Detroit News article mentioned in Chapter 5 (Hunter, 2015). Then, k bits of the individual's sequence are sampled at random and their value set to 1, while the remaining bits are set to 0.

Constraining the number of agents

Evidently, in the real world, resources are not unlimited and although minimising response time is a priority for police agencies, they have to do so with a finite amount of resources, which represents a fixed constraint. The PDOP considered in this thesis is thus a Constrained Optimisation Problem (COP). The most common way to handle such a constraint within a GA is to add a penalty function which gives a fitness disadvantage to individuals that violate it (Eiben and Smith, 2015). In the case of Detroit, 'unfeasible' individuals are those featuring a number of agents outside of the arbitrary range $[1, 60]$. A penalty would thus be applied to a candidate individuals featuring 65 agents, for example. In order to save on computational time, 'unfeasible' individuals do not go through the process of evaluation – which relies on the running of an ABM. Instead, their fitness is immediately set to a very poor value (see Chapter

7 for details) and are thus unlikely to be selected by the GA for the next generation.

This penalty handling approach is used in both the single-objective and multi-objective GA variants to avoid exploring unfeasible solutions which cannot be implemented in the real world. However, the value of the penalty differs between the two variants and will be detailed in the next chapter (see Chapter 7 for details).

Tournament parent selection

Although several methods exist for selecting parents, the tournament selection is perhaps the most widely used selection operator in GAs due to its conceptual simplicity and ease of control (Eiben and Smith, 2015). The tournament selection involves running several 'tournaments', each opposing k individuals chosen at random from the population. The winner of each tournament – i.e. the individual with the best fitness – is selected for breeding.

In practice, λ tournaments are conducted, leading to the selection of λ parents from the initial population of μ individuals. The larger the tournament size k, the greater the chance that the resulting breeding population will contain individuals of above-average fitness, and the less that it will consist entirely of low-fitness individuals. Hence a higher value of k tends to create a higher selection pressure.

In the single-objective GA developed in this thesis, it is this tournament selection method that is chosen for the parent selection. As per the literature (Eiben and Smith, 2015), the tournament size is set to $k = 3$ and the selection is conducted with replacement. As such, each tournament is independent, and the same parent individual may be present in multiple copies in the resulting breeding pool. In the multi-objective GA, parent selection uses a modified tournament operator (see Chapter 7 for details).

Bit flip mutation

Although a few other schemes have been occasionally used, the most common mutation operator for binary encoded individuals – and the one used in this thesis – is the 'bit flip' (Eiben and Smith, 2015). In this mutation process, each individual in the population has a probability $mutpb$ to undergo a process of mutation. Then, within individuals themselves, each feature is allowed to flip (i.e. from 1 to 0 or 0 to 1) with a small probability p_m, as illustrated in Figure 6.2. The actual number of values changed in an individual is thus not fixed, but depends on the sequence of random numbers drawn. For instance, for a mutated individual with an encoding of l features, on average $l \times p_m$ features will be mutated.

Figure 6.2: Bitwise mutation for binary encodings

A high p_m value tends to increase population diversity, leading to a better coverage of the search space (Eiben and Smith, 2015). However, when p_m becomes too high, the GA is reduced to a random search (Eiben and Smith, 2015). As such, a higher p_m is often coupled with a more aggressive selection process to ensure the best solutions are not lost.

GA studies typically recommend a mutation rate p_m of between $1/l$ (where l is the number of features) and $1/\mu$ (where μ is the population size) (Eiben and Smith, 2015), with values smaller than 0.5 typically used. Here p_m (the probability of each bit/feature to be flipped) is set to 0.1 and *mutpb* (the probability of each individual to undergo mutation) is set to $1/\mu = 0.025$.

Two-point crossover

There are typically two types of crossovers (as shown in Figures 6.3): one-point and two-point crossovers. In one-point crossovers, a point on both parents' chromosomes is picked randomly in the range $[1, l-1]$ (where l is the number of features), and designated a 'crossover point'. The tails to the right of that point are swapped between the two parents. This results in two offspring, each carrying some genetic information from both parents (Eiben and Smith, 2015). In two-point crossovers, two crossover points are picked randomly from the parent chromosomes and the chromosome sections in between the two points are swapped amongst the parents (Eiben and Smith, 2015).

Early GA literature recommends a crossover probability p_c of between 0.5 and 0.9 (Eiben and Smith, 2015). As such, the GAs developed here feature two-point crossovers with an arbitrary crossover probability $p_c = 0.9$.

Survivor selection and population replacement

The survivor selection and population replacement approach differ between the single-objective and the multi-objective GA variants, although in both cases, $\lambda = \mu = 40$. More details about

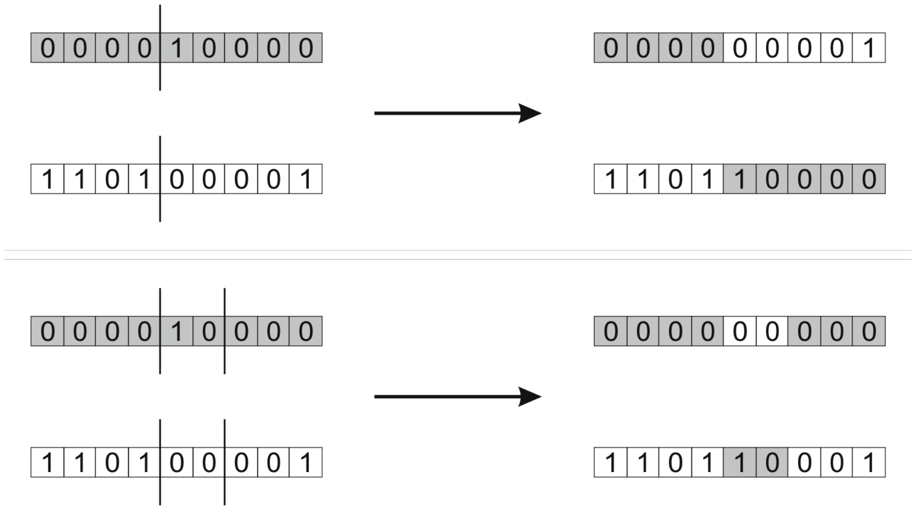

Figure 6.3: One-point crossover (top) and two-point crossover (bottom) for binary encodings

this parameter can be found in the next chapter (see Chapter 7).

6.5 Running the GAs

The previous section described the problem representation and parameters chosen for the design of the two GA variants built in this thesis. This section presents information related to the running of the GA itself. This includes the data sources required to run the GA, the metrics used to measure the performance of the GA learning and the method implemented to prevent over-fitting.

6.5.1 Data

As previously mentioned, the GA relies on the ABM to evaluate the fitness of the individuals and as such, it necessitates the same data sources as the ABM itself. Those data sources, first described in Chapter 3, are related to (1) the model environment (precincts, patrol beats and road network), (2) the CFS incidents used to simulate a particular demand scenario and (3) the reported crimes used to quantify the crime-deterring effect of agents patrolling. Note that the latter is not needed in the single-objective variant of the problem because crime deterrence is not amongst the problem's objectives. Omitting this dataset when running the single-objective GA greatly speeds up the ABM initialisation step. As mentioned in Chapter 3, when the user does not provide a historical crime dataset, the density of historical incidents on all road segments is null ($density\_hist\_inc = 0$). As a result, upon initialisation, the patrol route in each beat is planned by arbitrarily sampling 5 streets. Idle agents thus patrol these streets but this action does not generate any crime deterrence (since the density of historical crime is null).

Much like in the ABM experiments conducted in Chapter 5, the GAs were run separately for both a low-demand and a high-demand scenario. Each scenario is composed of a pair of training and test sets, each containing 100 time periods from the year 2018 and 2019 respectively (see Section 5.2 in Chapter 5 for details on how these sets are produced).

The training sets are used by the GAs to evaluate the fitness of each individual at each generational step throughout the learning. The test sets, on the other hand, are used at the end of the training process to provide a fair evaluation of all those individuals which remained in the final population. Indeed, some of these individuals may have been selected during a particular generation because they yielded a good fitness when evaluated against the particular subset of time periods that was sampled at that generation. Evaluating individuals on the test set allows the level of generalisability of these solutions to previously-unseen data to be assessed. To this end, it is crucial that the test-set time periods be different from those of the training set. Although CFS demand in 2019 may differ from that of 2018, separating the two years for training and test sets arguably better reflects the manner in which the model would be used in real-world policing. Indeed, police agencies would typically train the model on historical data before using what the model has learned about past demand to make deployment decisions.

6.5.2 Measuring GA performance

Broadly speaking, there are two basic performance measures for a GA: (1) its effectiveness (i.e. solution quality) and (2) its efficiency (i.e. algorithm speed). The performance measure used in this thesis to monitor how well the GAs have learned is based solely on effectiveness, due to the limitations imposed on CPU running time by the HPC facility. In other words, the performance of the GA is here defined as the fitness of the best individual at termination, where the termination criteria is reached after a predetermined number of generations.

Monitoring population diversity throughout the learning process is another useful performance assessment tool for GAs. As the generations pass, population diversity tends to drop as a result of the same fittest individuals breeding with one another. This is a commonly observed phenomenon which results in many individuals in the population bearing identical features. It is important to prevent a premature convergence – i.e. a drop in diversity happening early in the learning process – so as to ensure the exploration of a wider search space. This is typically achieved by carefully selecting of the selection and variation operators (refer to the chosen values in Section 6.4).

As the population diversity naturally drops throughout the learning process, the number of identical copies of the same individuals increases in the population. To minimise the number of ABM runs – thus speeding the algorithm, only unique individuals in the population were evaluated with an ABM and the resulting fitnesses were then passed on to other identical copies for these individuals. This is made possible by the deterministic nature of the ABM developed in this thesis.

6.5.3 Preventing over-fitting

One of the most important goals of any machine learning approach is to find solutions that perform well not only on the cases used for learning but also on cases never seen before (Mitchell, 1997). This is known as generalisation. When the solution performs well on the training cases but poorly on the new (test) cases, the model is described as 'over-fitting'. This indicates that the underlying relationships of the whole data were not learned, and instead a set of relationships existing only in the training cases were learned, yet these have no correspondence over the whole possible set of cases. Over-fitting in GAs and the related field of Genetic Programming (GP) is a common issue in the literature (see a review in O'Neill et al., 2010).

Liu and Khoshgoftaar (2004) introduced a method called Random Sampling Technique (RST) for improving model generalisation in GP and reduce running time. With RST, the training set is never entirely used in the search process. Instead, at each generation, a random subset of the training data is chosen and evolution is performed taking into account the fitness of the solutions on this subset only.

When using the RST approach, two parameters are specified: (1) the Random Sampling Subset (RSS) which represents the percentage of training set used to evaluate solutions and (2) the Random Subset Reset (RSR) which represents the frequency at which to re-sample a new subset of the data (e.g. at every generation or every 5 generations etc.).

Experiments in GP conducted using the RST suggest that using a small and frequently changing subset of the training data (low RSS and low RSR) is most effective in reducing over-fitting and improving generalisation (Langdon, 2011; Silva and Gonçalves, 2011). The idea is that, while using more learning cases (a bigger training sample size) intuitively facilitates the learning process, it also aggravates the over-fitting issue. These studies found that having a low RSS value does not tend to damage the training fitness as long as the changes on the random subset occur often enough (low RSR value). In particular, even using only a single training instance and changing it every generation was shown to be able to achieve the same learning outcomes (Gonçalves et al., 2012; Langdon, 2011; Silva and Gonçalves, 2011).

With RST, only those individuals that perform well on various different subsets will remain in the population. As a result, it is expected that, since these surviving individuals perform reasonably well on different subsets, they have captured the underlying relationships of the data instead of over-fitting it.

If the results from these studies can translate to the field of GA and be applied to the context of the PDOP, then this would mean that the overall CPU time for training the GAs could be dramatically reduced by the use of a smaller training subset at each generation. In Chapter 7 that follows, an experiment is conducted for the single-objective GA which confirms that a low-RSS-low-RSR setup does not indeed deteriorate the quality of the final solutions. The implication of the results from this experiment are discussed in Chapter 7.

6.6 Summary: designing GAs for the police deployment optimisation problem

This chapter started by justifying the need for a metaheuristic algorithm to automate the search for optimal solutions to the PDOP (see Section 6.2). More specifically, it is a type of metaheuristics called the Genetic Algorithm that is chosen in this thesis for its flexible implementation and ability to be combined with an ABM. Section 6.3 introduced the concepts of natural evolution and how GAs relate this phenomenon to the solving of optimisation problems. Then, Section 6.4 listed the design choices that were made in this thesis to apply GAs to the police deployment optimisation problem. Finally, Section 6.5 presented the logistical considerations involved when running the two GA variants developed in this thesis. These considerations included ways to monitor model performance and prevent over-fitting. The next chapter details results from running the single-objective and multi-objective GA variants for the exemplar city of Detroit.

Chapter 7

Applying the GAs to finding solutions to the PDOP in Detroit

7.1 Introduction

The chapter begins in Section 7.2 by describing results from a single-objective GA concerned with a simplified version of the PDOP featuring only one objective: to minimise the average incident response time. The section details the specific implementation of this single-objective GA and its results applied to the city of Detroit.

In the real world, however, police agencies are faced with more objectives than merely minimising response time. Importantly, they are concerned with meeting both reactive and proactive demand while keeping the cost of operation as low as possible. Section 7.3 explores the use of a multi-objective GA to solve a version of the PDOP which features multiple conflicting objectives, including maximising crime deterrence score, minimising number of deployed agents, minimising percentage of 'failed' responses in addition to minimising the average response time. The section highlights the parameters of the multi-objective GA and presents the results from its implementation on the case study of Detroit.

7.2 Single-objective GA applied to Detroit

In the first instance, the PDOP is simplified as a single-objective optimisation problem solely concerned with finding the deployment configuration(s) that minimise the average response time to incidents. This section first details the GA parameters that are specific to this variant of the PDOP. Then, an experiment is conducted to demonstrate the applicability of the Random Sampling Technique (RST) technique introduced in Chapter 6 to the GA developed in this thesis in order to prevent over-fitting. Finally, the section highlights the results of the learning process of the single-objective GA and presents the optimal deployment configuration that were identified for the case study of Detroit.

7.2.1 Parameters of the single-objective GA

Fitness function

As discussed previously in Chapter 6, at each generation, the GA evaluates all individuals using a fitness function. In the single-objective variant of the PDOP, the fitness of an individual (representing a deployment configuration) is defined as the average response time to incidents across all k time periods considered for evaluation (where k is the Random Sampling Subset (RSS) introduced in the previous chapter). The choice of value for k used in this study is justified in the RSS-tuning experiment below. The average response time is estimated by running the ABM detailed in Chapter 3.

A penalty is placed on individuals which feature a number of agents outside of the predefined range, as previously explained in Chapter 6. For Detroit, this range is arbitrarily set to $[1, 60]$, as previously mentioned. As part of the penalty, 'unfeasible' individuals are given an arbitrary fitness value of 1000 (representing an average response time of 1000 minutes), to significantly lower their probability of making it into the next population.

As previously mentioned, there is no need to provide the ABM with a dataset of historical crimes in this instance, as the deterrence score is not a metric of interest in the single-objective variant of the GA. When provided, this dataset is normally used upon the initialisation of the ABM to determine the hottest streets to visit as part of the patrol route of each beat (see Chapter 3). In the absence of historical crime dataset, the model instead initialises the patrol routes of each beat to visit a set number of randomly selected streets. Omitting this dataset greatly speeds up the initialisation of the ABM, ultimately leading to much faster generations of the GA.

Parent and survivor selection

In the GAs developed in this thesis, a simple setup is chosen in which $\mu = \lambda = 40$ (where μ is the population size and λ is the number of offspring). In the single-objective GA, at each generation, μ parents are selected through repeat tournaments, each involving $k = 3$ individuals. Because the tournaments are conducted with replacement, highly fit individuals in the population have a higher chance of being present in multiple copies among the selected pool of parents. The parents then breed to create λ offspring; a process which involves the application of mutation and crossover variations.

Finally, through the (μ, λ) generational population replacement (see description in Chapter 6), all λ offspring enter the next population without survivor selection. Parents thus do not remain in the next population and are entirely replaced by the offspring. However, as previously mentioned in Chapter 6, there may be individuals that are identical to their parents in the next generation, provided that no recombination or mutation has taken place.

Summary

Table 7.1 provides a summary of the parameters used in the single-objective GA applied to the case study of the PDOP in Detroit. Much of the values for these parameters have been previously discussed in Chapter 6 as they are shared between both GA variants.

Table 7.1: Parameters of the single-objective GA for the case study of Detroit

Parameter	**Value**
Runs	1
Population size μ	40
Generations	40
Parent selection	Tournament (rank based) of size $k = 3$ with replacement
Crossover operator	Two-point crossover between two parents, $p_c = 0.9$
Mutation operator	Bit flip, $p_m = 0.1$, $mutpb = 0.025$.
Offspring size λ	40
Survivor selection	None
Replacement	(μ, λ) with $\lambda = \mu$: the parents are replaced by all offspring at each generation
Penalty handling	Fitness set to 1000 for individuals with a number of deployed agents outside of the predefined range (e.g. $[1, 60]$ for Detroit)

7.2.2 RSS-tuning experiment

Experiment setup

As discussed in Chapter 6, studies from the field of Genetic Programming suggest that a low Random Sampling Subset (RSS) – as low as $RSS = 1$ – may prevent the GA from over-fitting without hindering the learning of the GA, as long as it is combined with a frequent re-sampling (every generation for instance).

An experiment is conducted to confirm the validity of these results when translated to a GA (instead of GP), specifically in the case study of the PDOP in Detroit. In the context of the PDOP, the RSS parameter represents the number of randomly sampled time periods on which each individual is evaluated in the GA at every generation. Each time period corresponds to one simulation run through the ABM.

The following values were tested for the RSS parameter: 1, 2, 5 and 10 time periods from the low-demand scenario training set. In a similar fashion to Gonçalves and Silva, 2013, for each RSS value, 4 runs of the GA were executed. The GA parameters used in this experiment are for the most part identical to those described in Table 7.1 above. However, in this experiment, the GA was only trained for 20 generations instead of 40 (due to constraints of computational power).

These experiments are highly computationally expensive. As such, an exhaustive experimentation with higher RSS values (e.g. 15, 20 etc. time periods) would be impractical given the limited computational power available. For the same reason, the experiment was not repeated for the high-demand scenario.

Performance measure

As illustrated in Figure 7.1, for each run, a generation proceeds according to the following three steps. First the GA evaluates the fitness of all individuals (deployment configurations) in the population using k randomly sampled time periods from the training set (where k is the RSS value). In other words, the ABM is run k times to simulate a given deployment configuration over k distinct time periods from the low or high demand scenario. Then, the average response time across all k time periods is calculated as the fitness of the individual. Second, the individual with the best fitness – i.e. the deployment configuration yielding the lowest average response

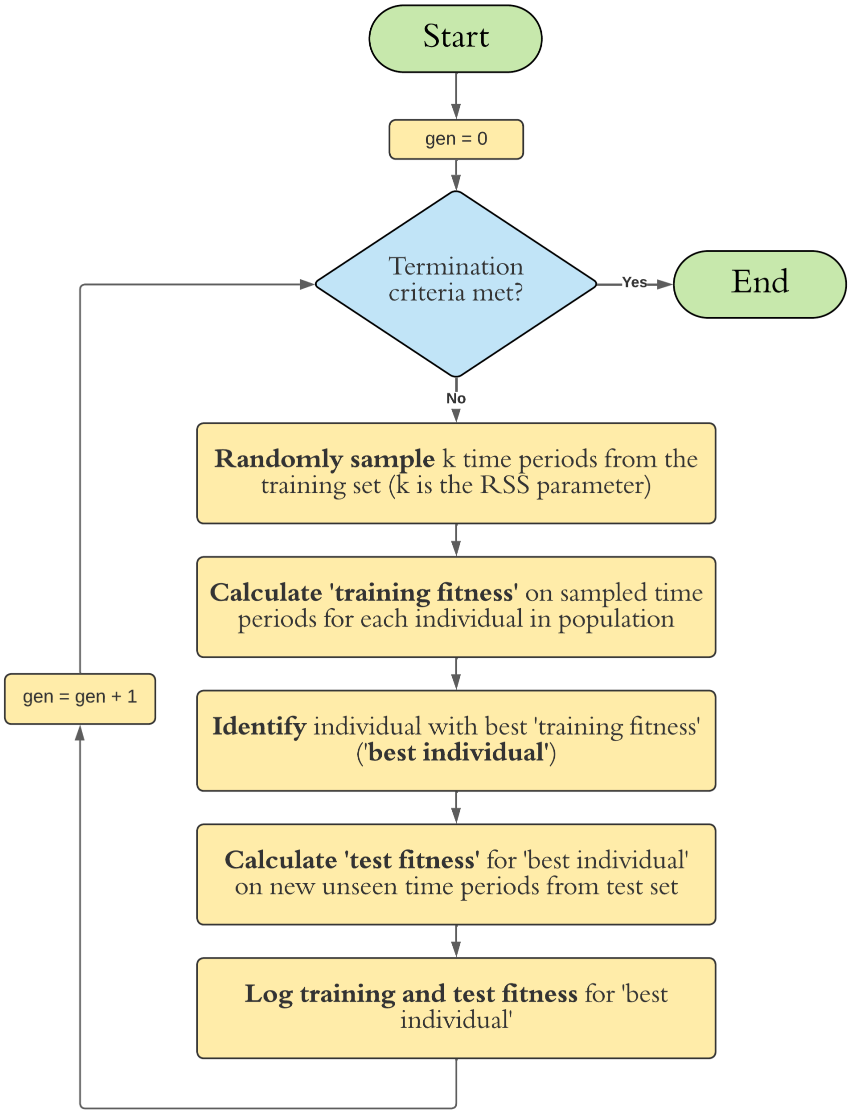

Figure 7.1: Diagram detailing one run of the RSS tuning experiment for a given RSS value

time – is selected and the corresponding fitness is logged as 'training fitness'. Third, this best individual is evaluated on 20 random unseen time periods from the test set (2019) and the resulting fitness is logged as 'test fitness'.

Because the training fitness is calculated as the average response time across only a few time periods (RSS is 1, 2, 5 or 10), it is expected to fluctuate more between generations than the test fitness, which is calculated as the average response time across $n = 20$ time periods.

At the end of the 4 runs, the median for both training and test fitnesses are calculated across individuals and runs. The median was chosen over the mean in all the evolution plots shown in the next section as it is more robust to outliers.

Experiment results

Figure 7.2 displays the evolution of training and test fitness for the best individual selected at each generation. The graphs show that all four values of the RSS parameter produce an overall constant gap between the training and test fitness values, with only minor variations. A widening gap between training and test fitness usually indicates that over-fitting is taking place. As such, the results from this experiment suggest that no over-fitting occurs for values of the RSS parameter ranging from 1 to 10.

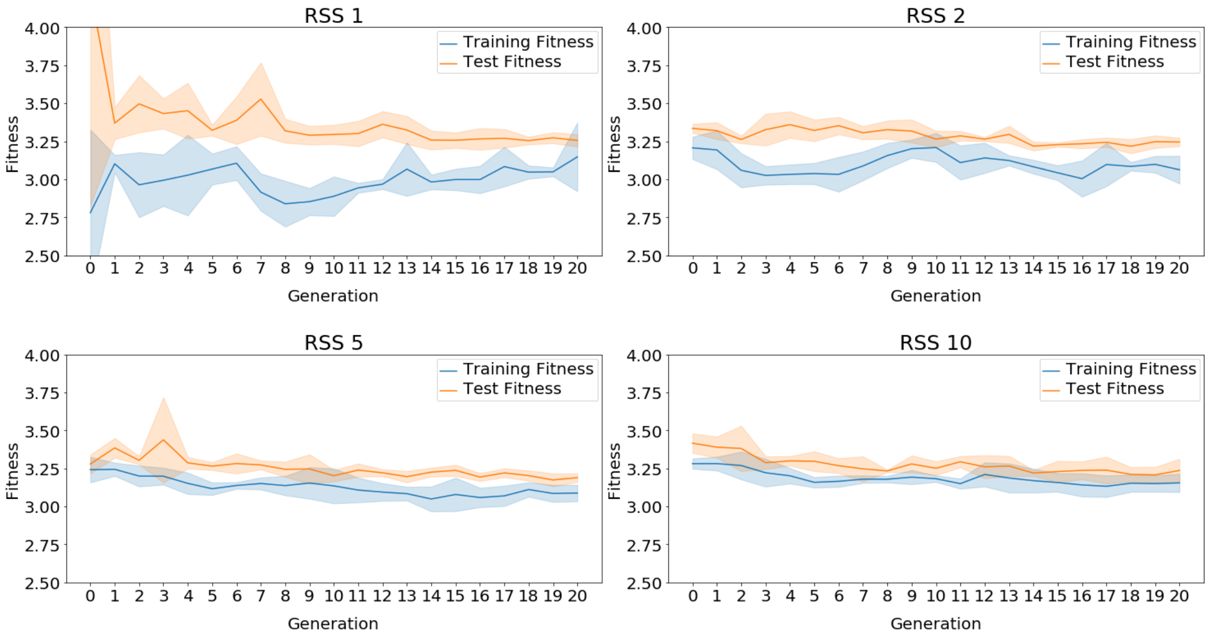

Figure 7.2: Evolution plots for different values of the RSS parameter. The y axis represents the fitness (average response time in mins) of the best individual (deployment configuration). Note: the curves show the mean and 95% confidence intervals around the mean (using the bootstrap method).

The evolution of population diversity displayed in Figure 7.3 indicates that all four values of the RSS parameter result in a gradual decline in population diversity throughout the learning. This indicates that the GA did not converge too quickly towards a local optima and instead continued to explore the wider parameter space.

Considering these preliminary results, the RSS value is set too $k = 1$ in the single-objective GA developed in this thesis. As such, at every generation, a single time period is randomly sampled from the training set (for a given demand scenario) on which all individuals in the population are evaluated. This time period is randomly re-sampled at each generation from the training set (2018).

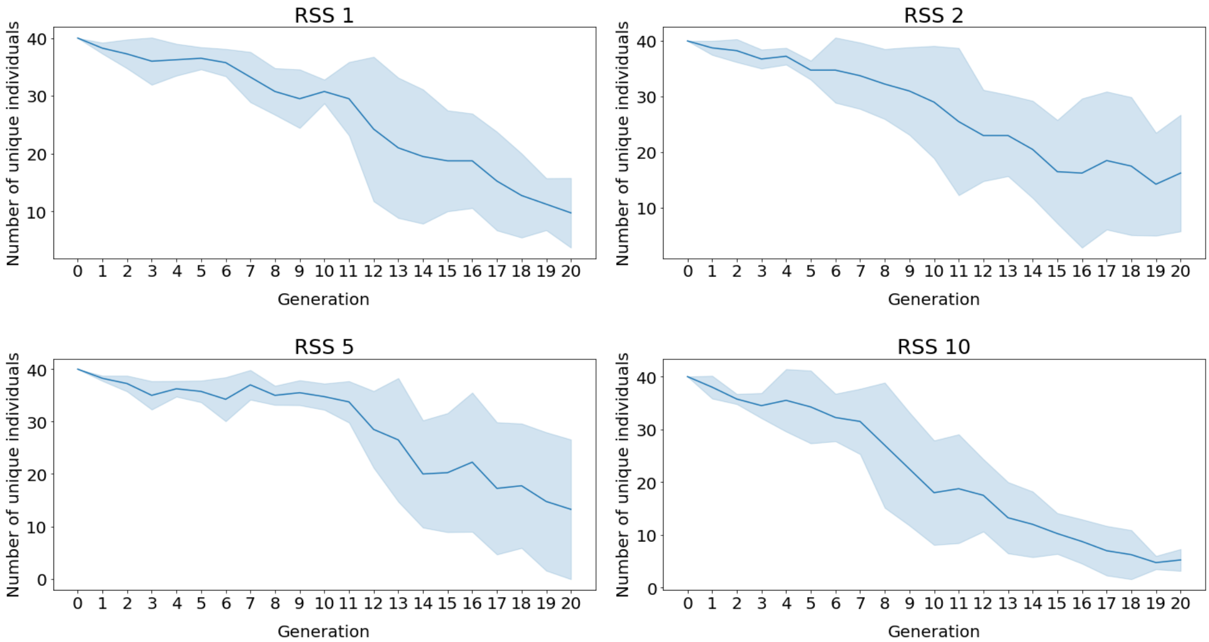

Figure 7.3: Population diversity throughout the learning for various values of the RSS parameter. Note: the curves show the mean and 95% confidence intervals around the mean (using the bootstrap method).

7.2.3 Results

Visualising the learning

Figure 7.4 provides an overview of the learning of the single-objective GA under both demand scenarios. The graphs on the left-hand side show the increase in the average number of agents featured in individuals as well as the simultaneous decrease of the median response time across individuals in the population. The median is used here instead of the mean because the penalty on fitness for 'unfeasible' individuals (response time = 1000 mins) makes the mean response time fluctuate in ways that do not represent the true average across individuals. The individuals (i.e. deployment configurations) that make up the population may differ from one generation to the next, as the GA learns and selects better individuals.

The graphs on the right-hand side show the decrease in population diversity throughout the learning. As previously mentioned, this is an expected behaviour which indicates the convergence of the model towards a small subset of highly-fit individuals (i.e. high quality deployment configurations). From the first generations onward, the GA learns that configurations with more agents produce faster response times. Through the selection and breeding of better individuals (mostly those with more agents), the GA eventually converges towards a smaller number of fitter individuals featuring 60 agents – the maximum number of agents as per the constraint.

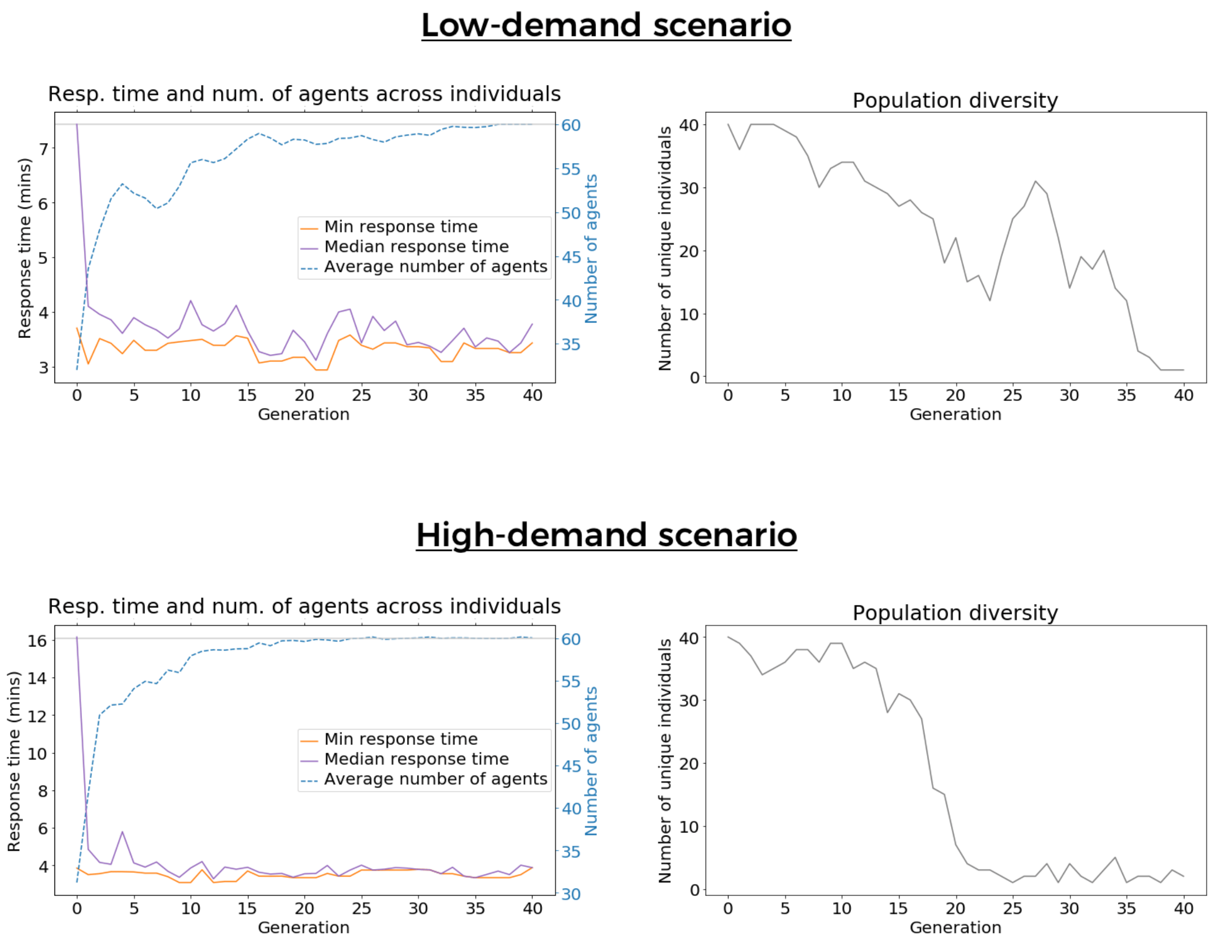

Figure 7.4: Visualisation of the learning of the single-objective GA on low (top) and high (bottom) demand scenarios throughout the generations. Left: evolution of median and min response time (fitness) as well as average number of agents across individuals. Right: evolution of population diversity throughout the generations.

Interestingly, although the GA eventually converges towards 60-agent solutions under both demand scenarios, it does so more quickly under the high-demand scenario (within about 22 generations) than under the low-demand one (within about 37 generations). This may be due to the higher pressure applied in the high-demand scenario. Indeed, as the system is more stretched under high-demand, small changes to configurations may lead to considerable increases in response time (as seen in the ABM experiments produced in Chapter 5).

An equilibrium is found when a handful of 60-agent individuals become present in multiple copies (clones) and thus take over the population. As these individuals breed, the crossovers become more likely to produce 'unfeasible' offspring which get assigned the aforementioned fitness penalty. Since these 'unfeasible' individuals have a low chance to reproduce, the rest of the search space is thus no longer explored and the learning slows down.

The most striking improvement in the learning for both scenarios happens in the first two generations. After these two generations, the individuals in the population appear to already produce satisfactory average response times (below 5 minutes). Although the GA has not yet converged and continues to fine-tune the individuals, the subsequent improvements to response time are unlikely to make a significant difference for police agencies. Technically speaking, this means that randomly sampling an individual (i.e. deployment configuration) from the population at generation 2 – or better still, choosing the individual with the minimum response time – should already provide the police with a deployment configuration that yields fast response times (below 5 minutes).

In order to confirm that the GA continues to explore the parameter space throughout the learning, a Hamming distance is calculated between each pair of individuals in the population at each generation. The Hamming distance, which represents the number of positions in which two individuals differ, is a commonly used measure of similarity between individuals of a GA population and is particularly useful when the individuals are represented as binary vectors, as is the case for the PDOP. Figure 7.5 shows the evolution of the mean Hamming distance between individuals in the population throughout the learning. The figure seems to confirm that the GA continues to explore the search space until the maximum number of agents is reached (40 generations in low-demand scenario and 20 generations in high-demand scenario) after which the population quickly becomes saturated with identical individuals that are no longer 'improvable'.

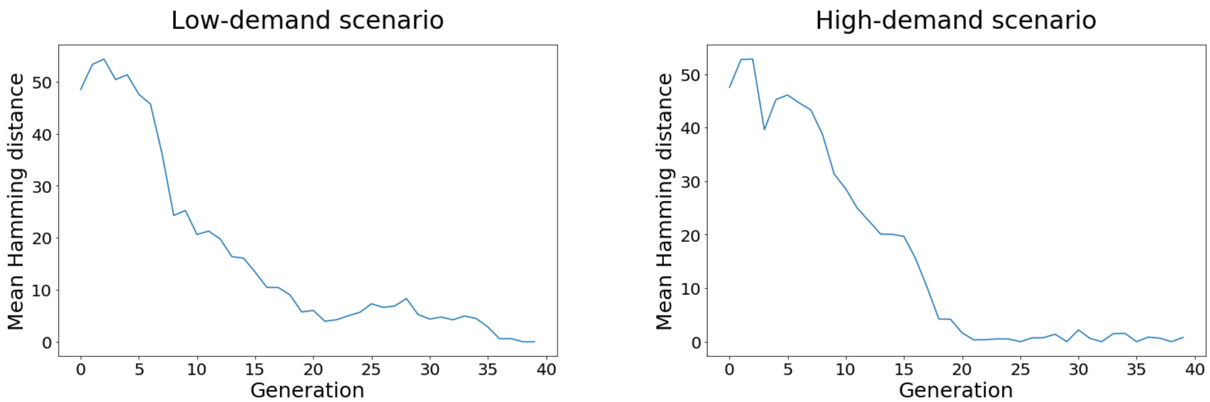

Figure 7.5: Evolution of the mean Hamming distance between individuals in the population throughout the generations.

Visualising the GA-identified solutions

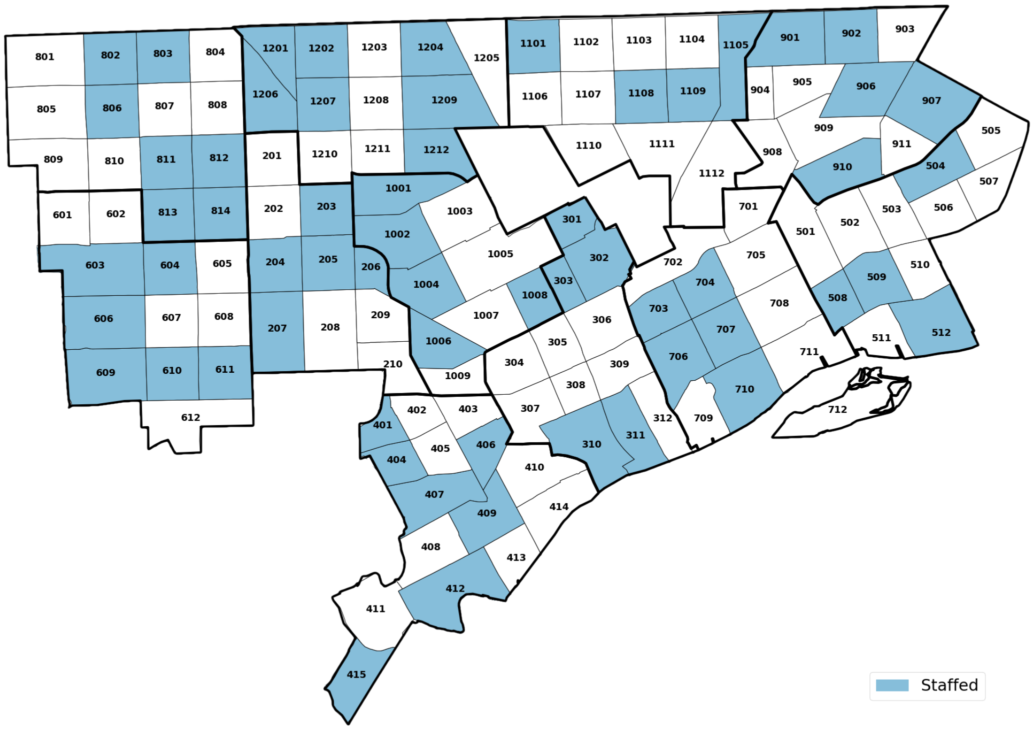

Figure 7.6: Best deployment configuration identified by the single-objective GA trained on low-demand scenarios. The configuration features 60 agents.

After 40 generations of training, the unique 'feasible' individuals present in the final population are evaluated one last time on the test set – composed of 100 time periods from the year 2019. This is done for both demand scenarios. The overall-fittest individual with the lowest response time is chosen and here forth called the GA-identified solution.

Coincidentally, in this instance, the final population for each scenario contained one unique 'feasible' individual present in multiple copies, alongside 'unfeasible' ones. After the final evaluation on the test set, the GA-identified configuration under the low-demand scenario yielded an average response time of 3.5 minutes while that under the high-demand scenario produced an average response time of 3.8 minutes. The GA-identified deployment configurations are displayed in Figures 7.6 (low-demand scenario) and Figure 7.7 (high-demand scenario).

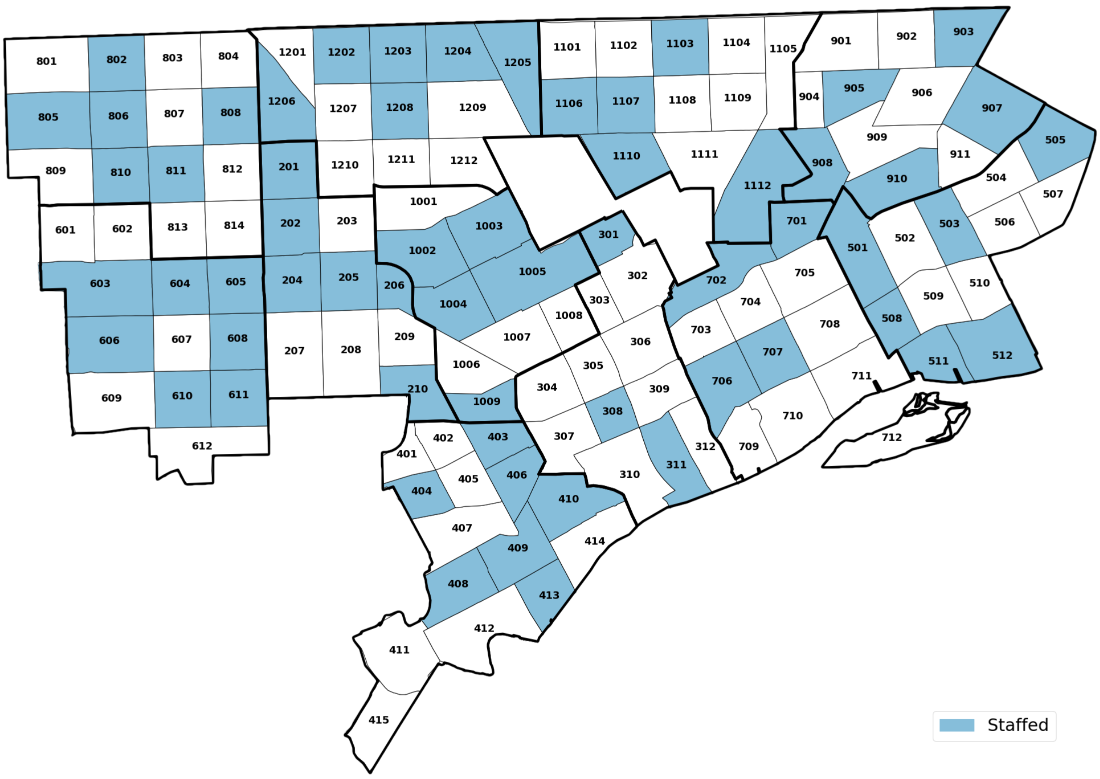

Figure 7.7: Best configuration identified by the single-objective GA trained on high-demand scenarios. The configuration features 60 agents.

Comparing results with a targeted deployment based on historical CFS

Having used the single-objective GA to find the best configuration under each demand scenario – i.e. that which yields the lowest average response time, it is now pertinent to compare these GA-identified configurations with their equivalent targeted deployment configurations that were introduced in Chapter 5. As a reminder, these consisted in a targeted deployment based on the number of historical CFS incidents that took place in each patrol beat during the 100 time periods that make up each scenario's training set (2018). In essence, a comparison is here made between a GA-derived solution and one that might be designed by police analysts based on historical demand. The GA-identified configurations for both scenarios feature 60 agents. As such, it is the corresponding 60-agent targeted deployment configurations – first introduced in Chapter 5, that are used for direct comparison.

The questions that are explored here concern (1) whether there are similarities in the staffing of both the GA-identified and the targeted configurations and (2) whether they yield similar average response time. In other words, the idea is to assess whether the GA is able to prescribe a

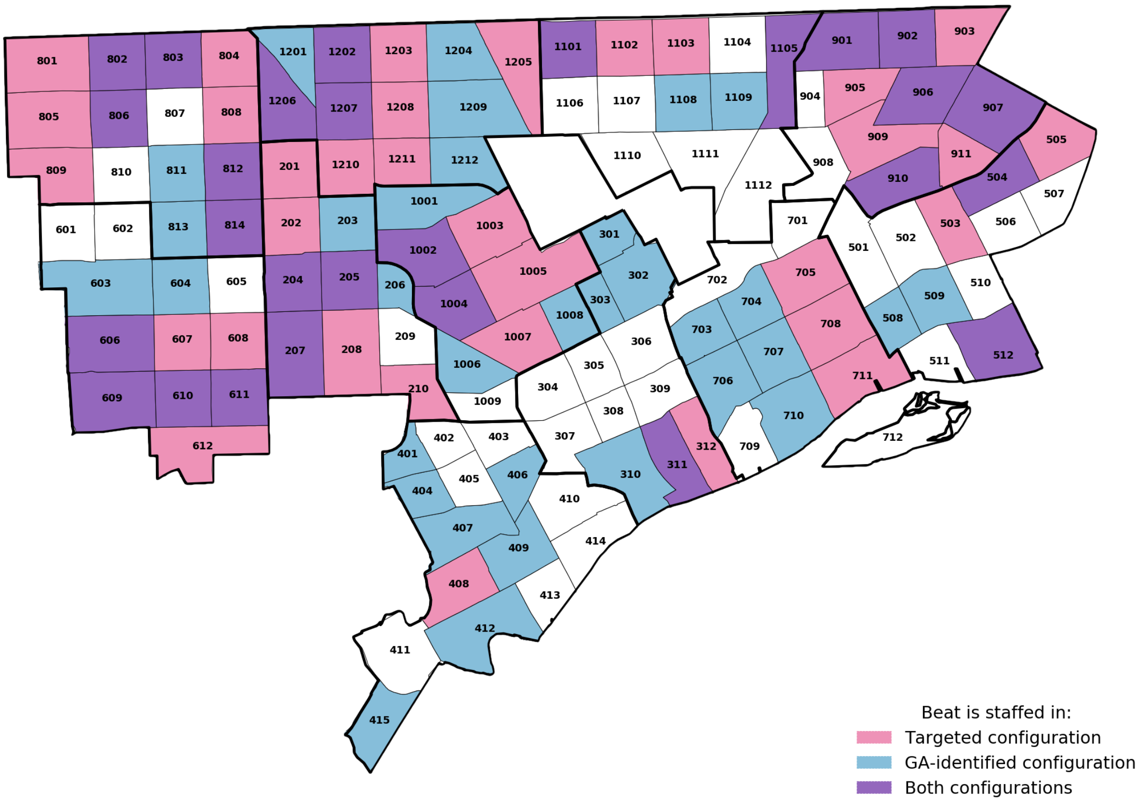

Figure 7.8: Comparison of the targeted and GA-identified deployment configurations for a low-demand scenario

configuration that yields a better response time than the simple targeted configuration designed in Chapter 5.

The two deployment configurations are directly comparable. Throughout the learning, the GA evaluates the response time yielded by configurations based on the CFS events that arise in the selected time periods from the training set (2018). This is the same set used to count historical CFS incidents in the algorithm used to design the targeted deployment configuration (see Chapter 5). As such, the GA-identified and the targeted deployment configurations are thus, in some ways, both targeted towards specific patrol beats based on historical CFS demand from 2018.

The staffed beats in each configuration as well as the overlap between them are shown in Figure 7.8 (low-demand scenario) and in Figure 7.9 (high-demand scenario). For both scenarios, the main difference between the configurations is that the GA-identified deployment appears to be more homogeneous than the targeted one. A heterogeneous level of staffing may lead to an unbalanced coverage with some entire precincts being over-staffed while others are under-staffed

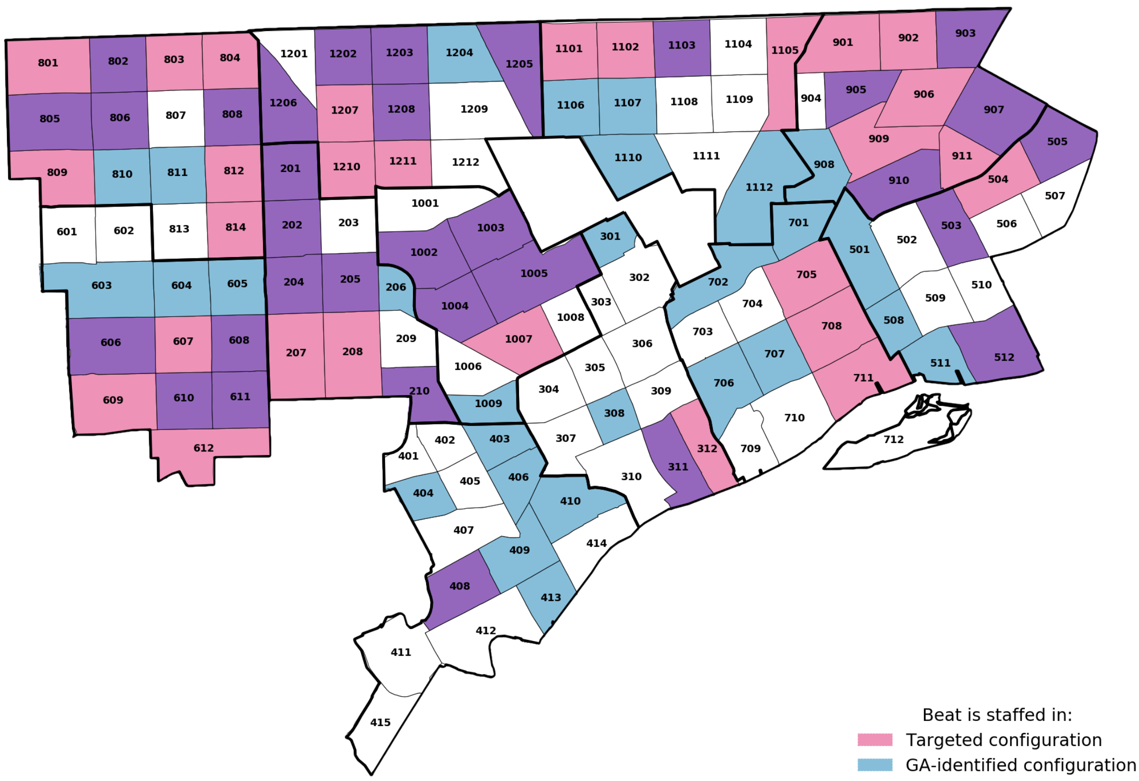

Figure 7.9: Comparison of the targeted and GA-identified deployment configurations for a high-demand scenario

(e.g. precinct 4 under the targeted deployment configuration).

In order to compare the performance of the targeted and GA-identified configurations, both configurations were evaluated on the same 100 time periods from each demand scenario's test set (from the year 2019). The resulting distributions of average response times under both low-demand and high-demand scenarios are displayed in Figure 7.10.

The figure suggests that, while both configurations appear to produce similar average response times, the values yielded by the GA-identified deployment tend to be more consistent than those produced by the targeted one. This seems to be true under both demand scenarios. For instance, the targeted configuration – featuring an unbalanced level of staffing – produced an average response time as high as 9 minutes on some low-demand time periods and as high as 15 minutes on some high-demand time periods. In comparison, the average response times yielded by the GA-identified configuration fluctuated less. The highest average response time value observed was around 4 minutes for some low-demand time periods and 5 minutes for some high-demand ones. One potential explanation is that the GA is better at generalising its solutions to any time period, as it was trained on a single randomly sampled time period at each generation, as explained earlier in this section.

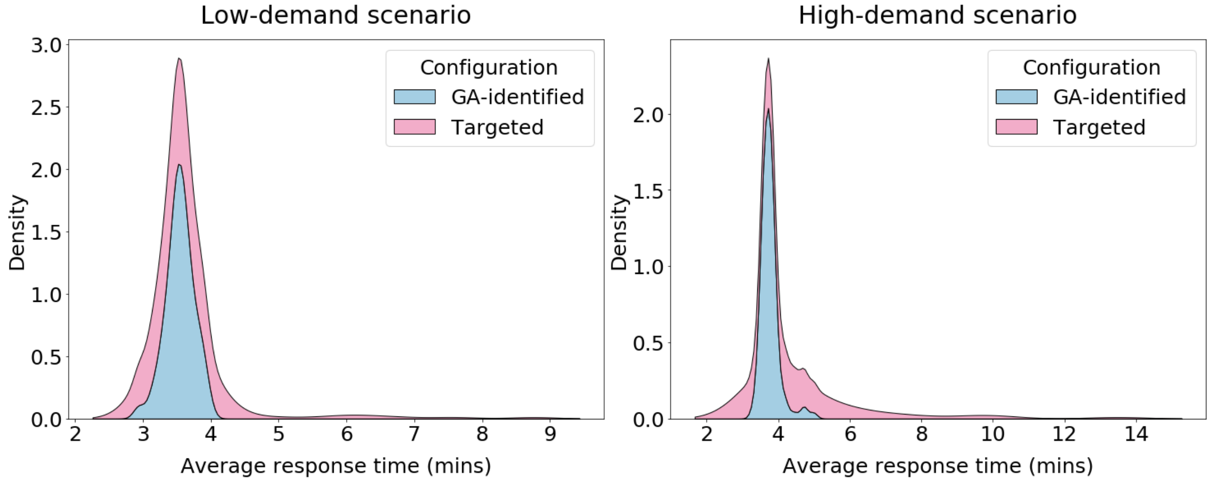

Figure 7.10: Distribution of the average response time for the targeted and GA-identified deployment configurations

To verify whether the average response times yielded by both configurations were statistically significantly different, a Kruskall-Wallis test followed by an effect size calculation were performed in a similar fashion to Chapter 5. The results for both demand scenarios are displayed in Table 7.2. The results suggest that, for a 60-agent configuration under a high-demand scenario, the GA-identified deployment brings a significant improvement in response time over the simple targeted one, with a moderate effect size. Specifically, responses were on average 22.08% faster with the GA-identified deployment under the high-demand scenario. However, the difference in average response time between GA-identified and targeted configurations was negligible under the low-demand scenario.

Table 7.2: Mean difference in response time between GA-identified and targeted deployment under low-demand and high-demand scenarios. Both configurations featured 60 agents.

Scenario	**Mean response time (mins)**			**% change**	**P-value**	**Effect size**
	Targeted	**GA-identified**	**Diff.**			
Low-demand	3.74	3.54	−0.20	−5.46 %	0.85	Negligible
High-demand	4.82	3.76	**−1.07**	**−22.08 %**	**<0.01**	Moderate

Note: the values in bold are those showing a sizeable and statistically significant difference in mean response time with a moderate to strong effect size.

Similarly, a Kruskall-Wallis test and effect size calculation were performed in order to verify whether there is a statistically significant difference between the percentage of 'failed' responses yielded by both types of deployment. The results, displayed in Table 7.3, suggest that, for a 60-agent configuration under either demand scenario, the GA-identified deployment brings a significant reduction in percentage of 'failed' responses over the simple targeted one. The effect size was moderate under the low-demand scenario and strong under the high-demand one. Specifically, the GA-identified deployment lead to an average reduction of 0.46% in 'failed' responses under the low-demand scenario and 1.89% under the high-demand one.

Table 7.3: Mean difference in percentage of 'failed' responses between GA-identified and targeted deployment under low-demand and high-demand scenarios. Both configurations featured 60 agents.

Scenario	**Percentage of 'failed' responses**			**P-value**	**Effect size**
	Targeted	**GA-identified**	**Diff.**		
Low-demand	0.00	0.46	$-$**0.46**	<**0.01**	Moderate
High-demand	0.16	2.04	$-$**1.89**	<**0.01**	Strong

Note: the values in bold are those showing a sizeable and statistically significant difference in percentage of 'failed' responses with a moderate to strong effect size.

Further work may explore whether the GA remains beneficial for configurations featuring a lower number of agents. This could be achieved by lowering the range for the constraint on number of agents (e.g. 40 agents instead of 60 in Detroit).

Summary

In this section, a GA was trained for each demand scenario to find solutions to the single-objective version of the PDOP applied to the city of Detroit. Results suggested that the GA identified a generalised configuration of 60 agents which yields more consistent response times than an equivalent targeted deployment configuration merely based on count of historical CFS incidents.

However, despite occasional longer responses, in the majority of cases, the targeted configuration yielded similar response times to those produced by the GA-identified one – i.e. the means were similar. As a police agency, DPD may find the results produced by the targeted deployment configuration sufficient. The pursuit of further reducing response times could arguably yield diminishing returns given the computational cost of training a GA.

As seen in this section, the single-objective version of the GA has a tendency to become 'greedy' and converge towards configurations which feature the maximum number of agents possible (within the constraint). This is of course not realistic as real-world resources represent a significant cost to police agencies. The number of deployed agents is arguably itself a metric to minimise. Additionally, as mentioned in Chapter 2, response time is not the only criteria to be considered in the PDOP. Other objectives include: (1) minimising the percentage of 'failed' responses (where the response time exceeded a threshold), (2) maximising crime deterrence through patrolling and (3) minimising the number of officers on duty (thus reducing cost).

When including these additional metrics, the PDOP becomes a multi-objective one. It is when faced with the complex nature of multi-objective problems that the true benefits of the GA come to light. The next section presents the results of a multi-objective GA and the benefits of presenting police agencies with a selection of optimal configurations from which to choose.

7.3 Multi-objective GA applied to Detroit

This chapter has thus far focused on a single-objective GA variant. Hopefully this variant has served as an effective introduction to applying this approach to the PDOP. However, as briefly discussed above, devising deployment strategies in the real world is a task which requires satisfying multiple constraints. The following section explores the use of a multi-objective GA to solve a version of the problem which features multiple conflicting objectives. The section begins with an introduction to multi-objective optimisation problems, including two important notions namely that of dominance and the Pareto front. Then, the design decisions specific to the multi-objective GA are listed. Finally, the results of the GA applied to the city of Detroit are presented, which prescribe a portfolio of configurations from which policy makers can choose.

7.3.1 Multi-objective optimisation problems (MOOPs)

When including several performance metrics – instead of only the average response time as was the case in the previous section, the PDOP becomes a multi-objective optimisation problem (MOOP). In this thesis, the PDOP is concerned with the four objectives summarised in Table 7.4.

Table 7.4: Objectives of the multi-objective version of the PDOP

Metric	**Objective**
Average response time (mins)	Minimise
Percentage of 'failed' responses	Minimise
Total deterrence score	Maximise
Number of deployed agents	Minimise

Conflicting objectives

In the ABM built in this thesis to simulate real-world patrol activities, agents can either be deterring crime through patrolling (proactive policing) or responding to CFS incidents (reactive policing). Since time is a finite resource, the more time an agent spends responding to incidents during their shift (driving or at the scene), the less they are able to patrol to deter crime. The objectives of minimising response time (reactive policing) and maximising deterrence score (proactive policing) are thus directly conflicting.

Additionally, it was shown in the ABM experiments conducted in Chapter 5 that deploying more agents yields better performance metrics: (1) faster incident response times (which was confirmed by the results of the single-objective GA), (2) a smaller percentage of 'failed' responses and (3) more crime deterrence through patrolling. As such, minimising the number of deployed agents is also in conflict with optimising these three performance metrics.

While one configuration can best satisfy one objective, it might not be optimal for another. As such, a single configuration does not exist that optimises all conflicting criteria (Eiben and Smith, 2015). Instead, it is typically desirable in MOOPs to present a diverse set of possible solutions representing a range of different trade-offs between objectives. Given that the priorities of police agencies may differ from one agency to the next or vary throughout the year based, for instance, on the currently available resources, it makes sense to allow practitioners to choose their preferred configuration based on their needs.

Dominance and Pareto front

In single-objective optimisation problems, the superiority of a solution over other solutions is easily determined by comparing their objective function values. In multi-objective optimisation problems, on the other hand, the goodness of a solution is determined by the concept of dominance (Eiben and Smith, 2015).

According to the dominance test (Eiben and Smith, 2015), solution x_1 dominates solution x_2 if (1) solution x_1 is no worse than solution x_2 in all objectives and (2) solution x_1 is strictly better than x_2 in at least one objective. For conflicting objectives, there exists no single solution that dominates all others. However, there may be multiple solutions that are called non-dominated, i.e. that are not dominated by any other (Eiben and Smith, 2015). All non-dominated solutions possess the attribute that their quality cannot be increased with respect to any of the objective

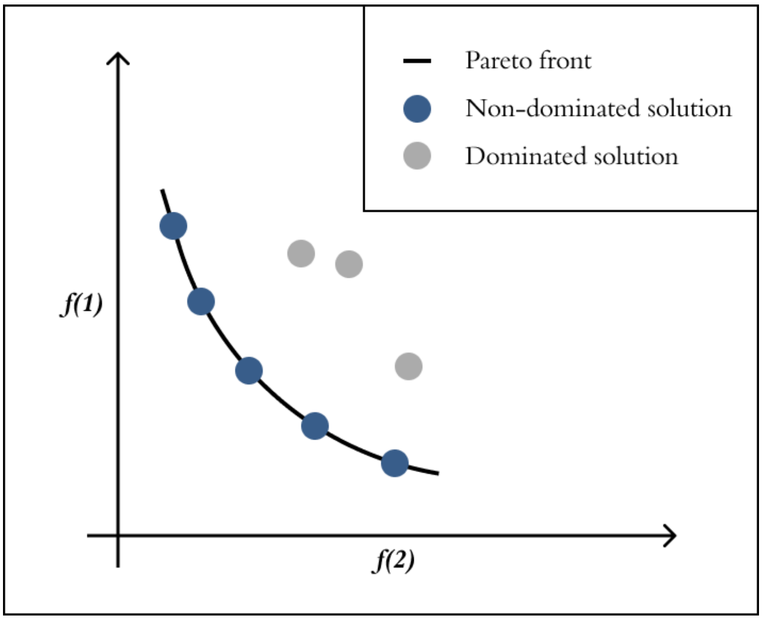

Figure 7.11: Illustrative Pareto dominance for the minimisation of two objective functions

functions without detrimentally affecting one of the others (Eiben and Smith, 2015).

The 'Pareto-optimal' front (or Pareto front), as illustrated in Figure 7.11 is the boundary defined by the set of all non-dominated solutions – or Pareto-optimal solutions (Eiben and Smith, 2015). Unlike in single-objective GAs where the outcome is a few highly fit diverse solutions, often present in multiple copies in the population, multi-objective GAs seek to distribute the population evenly along the Pareto front (Eiben and Smith, 2015). In other words, the goal of multi-objective GAs is to converge along the Pareto-optimal front whilst maintaining as diverse a distribution as possible.

An optimisation problem which requires to identify multiple local optima instead of a single global one is referred to as a multimodal problem (see Chapter 2). As previously mentioned, GAs are particularly well suited to solving multimodal problems, because they are able to promote and preserve diversity within the population while searching the parameter space.

One of the first genetic algorithms proposed for multi-objective optimisation was Srinivas and Deb's Non-dominated Sorting Genetic Algorithm (NSGA) (Srinivas and Deb, 1994). In order to converge towards the Pareto-optimal front, the GA needs to balance breeding and selecting non-dominated solutions, whilst maintaining diversity in the population. To that end, Deb and colleagues proposed the revised NSGA-II (Deb et al., 2000), which improves upon the initial idea of non-dominated fronts by incorporating the concept of 'crowding' for diversity maintenance and 'elite-preservation' for faster convergence (Eiben and Smith, 2015). These two concepts are now detailed.

The crowding distance metric is defined for each point as the average side length of the cuboid defined by its nearest neighbours in the same front (Deb et al., 2000). The larger this value, the fewer solutions reside in the vicinity of the point. When selecting individuals, it is the dominance rank that is typically first considered, then, in the case of co-dominance, the crowding distance is used to select individuals that are 'distant' from each other (Deb et al., 2000). Practically speaking, this process aims to guide the GA towards solutions that are uniformly spread-out along the Pareto-optimal front, ultimately providing a more diverse portfolio of solutions.

Elite-preservation is a commonly used approach to speed up the convergence of GAs (Eiben and Smith, 2015). In essence, at each generation, the best individuals – called elites – are automatically inserted into the next population without undergoing any change (crossover or mutation). In a multi-objective framework, any non-dominated solution can be considered an elite. Elite-preservation can very rapidly increase the performance of GAs, because it prevents losing the current fittest members of the population (Eiben and Smith, 2015). To achieve elite-preservation, the NSGA-II is combined with a $(\mu + \lambda)$ survivor selection strategy (see details below). In this type of survivor selection approach, the μ individuals from the current generation's population are combined with the λ offspring resulting from the mating process. The new population is then obtained by accepting individuals from progressively inferior fronts until it is full. This approach allows for elite non-dominated individuals to remain unchanged in the population from one generation to the next.

Summary: MOOPs

In multi-objective optimisation, multiple objective functions need to be optimised simultaneously. Usually, this means that there is no single optimal solution that satisfies all objective functions, but multiple 'Pareto optimal' solutions. The NSGA-II is amongst the most commonly used multi-objective GA algorithms for its ability to preserve diversity in the population throughout the search while converging towards the Pareto front of non-dominated solutions. It is this algorithm that is chosen for the multi-objective GA developed in this thesis. The next section introduces the design decisions and parameter values used specifically for the multi-objective GA.

7.3.2 Parameters of the multi-objective GA

Much like for the single-objective GA variant, individuals in the initial population of the multi-objective GA are randomly generated with a number of agents n. In the case of Detroit, this number of agents n is randomly sampled between 1 and 60 for each individual, where 60 is the maximum number of patrol vehicles to be deployed across the force.

There are, however, some differences in the design of the multi-objective GA, to account for the multiple conflicting objectives. These differences are now described.

Fitness function

The fitness function is composed of 4 objective functions, one for each of the four chosen metrics: (1) minimising the average response time (mins), (2) minimising the percentage of 'failed' responses, (3) maximising the total deterrence score and (4) minimising the total number of deployed agents.

All four metrics are given equal weights in the fitness function. When two solutions are co-dominant, they are ranked by the selection algorithm based on a secondary metric called a weighted crowding distance. This metric considers the specified weights to calculate the density of other solutions surrounding each solution in the frontier. For example, the algorithm considers an increase of 1 unit in response time to be as valuable as an increase of 1 unit in deterrence score. By adjusting the relative importance of different objectives, weights are used in the fitness function to compute the fitness of each individual, allowing the GA to combine the performance on different objectives into a single value. Ultimately, the weights in the fitness function balance the trade-offs between different objectives being optimised and guide the search towards a set of Pareto optimal solutions.

For each individual, the four metrics are calculated across all k time periods considered for evaluation at each generation (where k is the RSS value, i.e. the number of time periods on which each deployment is evaluated). If an low-quality individual is evaluated on a single time period featuring a low number of CFS incidents, it may yield a low response time and low percentage of failed responses, thus being considered a non-dominated solution. To prevent too many poor solutions from being counted as non-dominated by chance while maintaining a low computational cost, the RSS value is raised to $k = 2$ (instead of $k = 1$ in the single-objective GA).

Since the deterrence score was not relevant to the single-objective GA, it was not necessary to provide the model with a dataset of historical crimes. However, since crime deterrence is amongst the objectives of the multi-objective GA, it is essential to provide the model with such a dataset, so that agents may be given specific patrol routes that deter the most crime within their beat, and to calculate the total deterrence score achieved by the agents at the end of each ABM simulation. For details about the historical crime dataset used for Detroit – which contains all the relevant time periods for the demand scenario for the year 2017 – see Chapter 4.

Much like for the single-objective GA, a penalty is applied to individuals featuring a number of agents outside of the predefined range ($[1, 60]$ for Detroit). When penalised, individuals have (1) their average response time set to 1000 mins, (2) their percentage of 'failed' responses set to 100% and (3) their deterrence score set to 0. This is so as to set a very poor overall fitness across performance metrics for these 'unfeasible' individuals, and ultimately save computational time by not evaluating those deployment configurations that cannot be implementable in the real world due to supply constraints.

Parent and survivor selection

As mentioned earlier in this section, the chosen parent selection uses a modified tournament operator that first considers the dominance rank between two individuals, or crowding distance if the two individuals do not inter-dominate. Unlike the tournament technique used in the single-objective GA, this tournament algorithm does not use replacement. Instead, each individual can only be selected once at most. Through this tournament method, a subset of the whole population – called the breeding pool – is selected. Here the size of the breeding pool was arbitrarily set to 12 individuals out of the 40 that make up the population, so as to provide some selective pressure without risking to lose too many good solutions in the process.

The use of elitism is important to ensure that good solutions are not lost in the learning, due to the stochastic nature of the parent selection, crossovers and mutations. To that end, a $(\mu + \lambda)$ survivor selection approach is used with NSGA-II as the selection algorithm (see details provided earlier in this section).

Hall of fame archive

Keeping a 'hall of fame' is an additional approach to prevent good solutions from disappearing due to the stochastic nature of the selection process. An archive was here kept of the non-dominated solutions encountered throughout the learning. As the learning progresses, new non-dominated solutions are discovered and added to the archive. At each generation, the individuals in the archive are ranked based on domination and those individuals which are dominated are removed from the archive.

Summary: parameters of the multi-objective GA

All in all, while the multi-objective GA proposed in this thesis was designed with the same basis as the single-objective one, a few additional decisions were made to account for the presence of multiple metrics to optimise. These decisions mainly revolve around the design of the fitness function and the choice of parent and survivor selection algorithms. Table 7.5 provides a summary of the parameters used for the multi-objective GA applied to the PDOP in Detroit.

Table 7.5: Parameters of the multi-objective GA for a given demand scenario in Detroit

Parameter	**Value**
Runs	1
Population size μ	40
Generations	60
RSS	2 time periods
Parent selection	Tournament based on dominance between two individuals, or on crowding distance if the two individuals do not inter-dominate. Selects 12 parents without replacement.
Crossover operator	Two-point crossover between two parents, $p_c = 0.9$
Mutation operator	Bit flip, $p_m = 0.1$, $mutpb = 0.025$
Offspring size λ	40
Survivor selection	NSGA-II algorithm
Replacement	$(\mu+\lambda)$ selection: selecting μ individuals for the next generation
Penalty handling	For individuals outside of the predefined range ($[1, 60]$ for Detroit): response time set to 1000 mins, percentage of 'failed responses set to 100% and deterrence score set to 0.

7.3.3 Results

Visualising the learning

Unlike for a single-objective GA, where the success of the learning can be visualised by the steady decline in population diversity towards a single final solution, the performance of the learning in a multi-objective GA is less straightforward to assess, as a set of diverse solutions is expected to be maintained in the population throughout the learning. Because the true Pareto front for the PDOP is unknown, it is not possible to evaluate whether the set of non-dominated solutions identified by the GA at the end of the learning is indeed optimal.

One way to assess the quality of the learning is to visualise the number of non-dominated solutions that are identified by the GA throughout the learning. Figure 7.12 shows the steady increase in the number of non-dominated individuals present in the archive throughout the learning. In total, throughout the learning, the GA evaluated a maximum of 40 individuals x 60 generations = 2,400 configurations. As new individuals were evaluated, the non-dominated ones entered the archive. The observed steady increase in archived non-dominated solutions suggests that the GA continued to learn throughout the entire training phase. This increase appears to slow down in both scenarios when approaching 60 generations.

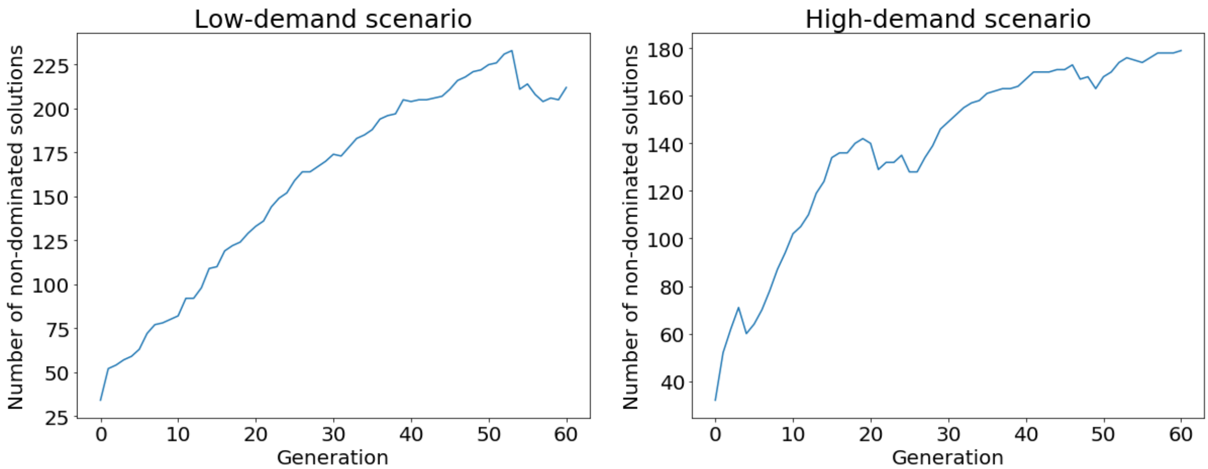

Figure 7.12: Steady increase in number of non-dominated solutions identified by the multi-objective GA throughout the learning

Visualising the non-dominated solutions

At the end of the training phase, a population of unique individuals is created for each demand scenario by combining the all-time-best individuals in the archive alongside the individuals in the last population. Then, much like for the single-objective GA, this population is evaluated one last time using the 100 time periods of the test set of the corresponding demand scenario. This is essential to provide a fair evaluation of these individuals that may have been added to the archive because they obtained a good fitness on the particular time period(s) on which they were evaluated during the learning process. This final evaluation on the test set allows for the identification of the true non-dominated individuals that form the Pareto front. The final evaluation returned 124 unique non-dominated solutions for the low-demand scenario and 91 for the high-demand one.

Because the multi-objective GA developed in this thesis considers four metrics, the Pareto front – which joins of all the non-dominated solutions identified by the GA – is a four-dimensional entity. In order to visualise the Pareto front in two dimensions, the non-dominated solutions are displayed according to pairs of metrics in Figure 7.13 (low-demand scenario) and Figure 7.14 (high-demand scenario).

Low-demand scenario

The performance of the 124 non-dominated solutions identified by the multi-objective GA for the low-demand scenario is displayed in Figure 7.13, showing one pair of metric at a time.

On the whole, results show that deployment configurations featuring more agents tend to lead to lower response times as well as produce more crime deterrence. These results are in agreement with those described in the experiments of Chapter 5 and those derived from the single-objective GA in Section 7.2.

The relationship between number of agents and crime deterrence score appears to follow a clear positive linear relationship. This finding is consistent with the experimental outcomes presented in Chapter 5. As previously stated, the current version of the model establishes the deterrence score as a direct function of the time that agents spend patrolling while idle. Therefore, the observed increase in crime deterrence score associated with deploying additional agents can be interpreted as a direct outcome of the increase in patrol time.

There is, however, no clear relationship between the number of agents and the percentage of 'failed' responses. There is nonetheless a clear positive relationship between the average response time and the percentage of 'failed' responses: a higher average response time correlates with a higher percentage of 'failed' responses, which is to be expected.

Looking at specific solutions along the Pareto front, it becomes clear that all non-dominated solutions offer performance trade-offs between metrics. For instance, while the cost of implementing solution 1 is minimal (only 1 agent deployed), it unsurprisingly yields the worst average response time (around 34 minutes) and the worst deterrence as a direct consequence of the low number of agents deployed (i.e. the single agent has little time to patrol). Additionally, 4% of

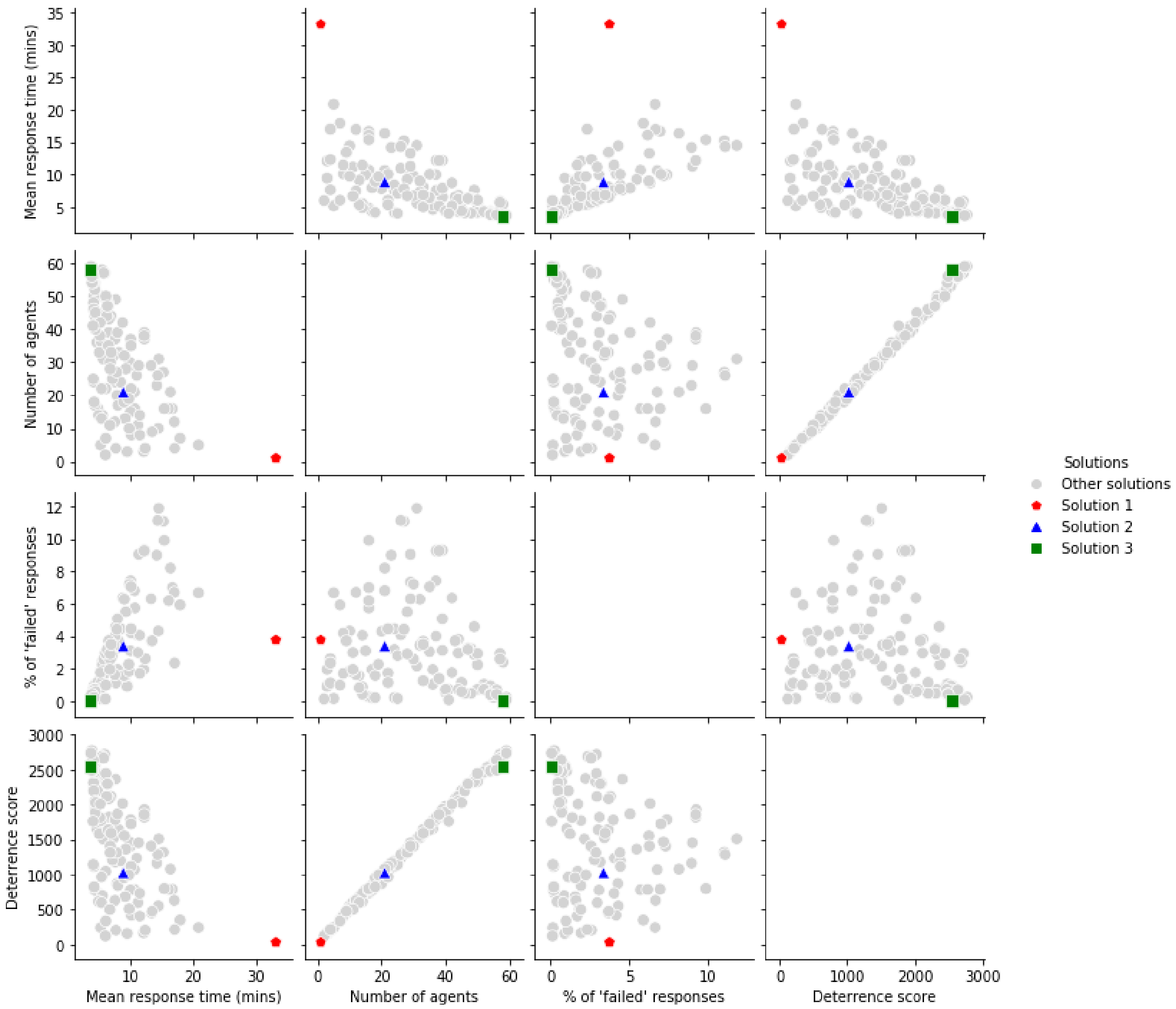

Figure 7.13: Performance of the 124 non-dominated solutions identified by the multi-objective GA under a low-demand scenario, displayed according to pairs of metrics. In particular, the performance of 3 selected solutions is highlighted across metrics.

responses end up 'failed' with that configuration. Solution 3, on the other hand, produces a high deterrence (i.e. much idle time for the agents), the lowest average response time (around 3 minutes), the lowest percentage of 'failed' responses but involves the deployment of 59 patrols, which may be very costly. Solution 2 offers a midpoint with 20 agents, an average response time of around 10 minutes, around 3.5% of responses 'failed' and an intermediate workload for the agents (which converts into time patrolling).

High-demand scenario

The performance of the 91 non-dominated solutions identified by the multi-objective GA for the high-demand scenario is displayed in Figure 7.14, showing one pair of metric at a time. The observed patterns are similar to those produced under the low-demand scenario. However, the negative relationship between number of agents and percentage of 'failed' responses appears to be clearer under the high-demand scenario. The relationship suggests that configurations featuring more agents tend to more consistently produce low percentages of 'failed' responses.

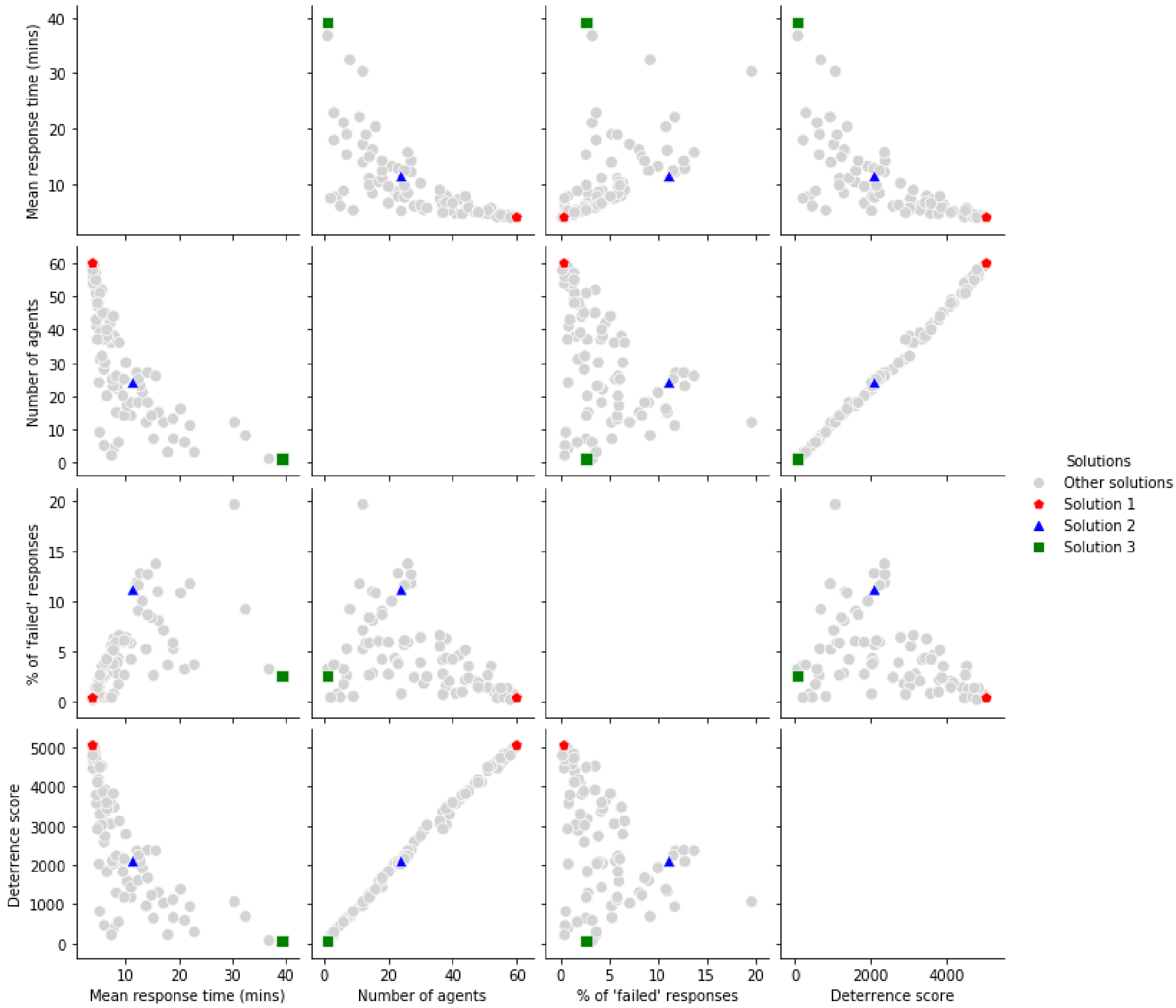

Figure 7.14: Performance of the 91 non-dominated solutions identified by the multi-objective GA under a high-demand scenario, displayed according to pairs of metrics. In particular, the performance of 3 selected solutions is highlighted across metrics.

The interpretation of the selected individual solutions is similar to that of the low-demand scenario. While the cost of implementing solution 3 is minimal (only 1 agent deployed), it unsurprisingly yields the worst average response time (40 minutes) and the worst deterrence as a direct consequence of the low number of agents deployed (i.e. the agent has little time to patrol). Additionally, 4% of responses end up 'failed' with that configuration. On the other end of the spectrum is solution 1, which produces the most deterrence (i.e. most time idle for the agents), the lowest average response time (around 3 minutes), a very low percentage of 'failed' responses but involves the deployment of 60 patrols, which may be very costly. Solution

2 offers a midpoint with 24 agents, an average response time of just above 10 minutes, around 11% of responses 'failed' and an intermediate workload for the agents (which converts into time patrolling).

Overall, for both scenarios, the non-dominated solutions appear to be evenly spread along the Pareto front, which highlights the success of the learning. In addition, the observed performance patterns for configurations featuring a varying number of agents are aligned with those described in Chapter 5.

A portfolio of solutions

As previously mentioned, the GA identified 124 solutions for the low-demand scenario and 91 for the high-demand scenario. This may represent too many solutions for policy makers to choose from. As such, a portfolio of 20 diverse configurations is here extracted from the Pareto front for each scenario. The solutions are chosen based on crowding distance, thus offering a diverse range of performance (as summarised in Table 7.6 for low demand and Table 7.7 for high demand) depending on the priorities of the police agency.

Low-demand scenario

For the low-demand scenario (see Table 7.6), consider the example of the proposed 60-agent configuration. With such a configuration, for which the spatial placement is displayed in Figure 7.15, the simulation results suggest that DPD might expect an average response time of 3.77 mins, 0.24% of 'failed' responses and a deterrence score as high as 2775.14. This average response time seems coherent with previous results from the single-objective GA for a 60-agent configuration. On the other end of the spectrum is the suggested 5-agent configuration (see corresponding spatial placement in Figure 7.15), which, according to the GA, yields an average response time of 20.85 mins, 6.67% of 'failed' responses and a deterrence score as low as 239.04.

Importantly, the spatial placement of patrols may differ considerably between each of the prescribed solutions, even when these feature a similar number of agents. Consider for instance the 47-agent and 48-agent configurations. As shown in Figure 7.15, their spatial placement do not overlap completely. As a result, even though the number of deployed patrols is higher in the 48-agent configuration than in the 47-agent one, the average response time is higher for the 48-agent one (6.28 mins compared with 4.08 mins). This is because the 48-agent configuration did not make it to the non-dominated front because of its average response time but instead

Table 7.6: Performance metric values for 20 configurations in the portfolio of chosen solutions for the low-demand scenario

	Num. of agents	**Avg. resp. time (mins)**	**% ‘failed’ responses**	**Deterrence score**
A	5	20.85	6.67	239.04
	9	13.34	3.71	446.56
B	10	14.44	4.32	549.01
	13	9.13	1.57	682.38
	16	10.68	5.73	780.35
	20	7.80	4.29	868.66
	22	10.62	6.80	1069.50
	26	7.06	2.80	1209.14
	28	6.72	2.91	1312.47
C	30	11.27	9.04	1453.00
	33	8.09	3.68	1641.35
	35	10.04	7.04	1722.63
	37	4.58	1.04	1654.06
	39	5.10	1.39	1812.90
D	47	4.08	0.48	2194.83
E	48	6.28	3.17	2296.49
	50	7.60	4.59	2362.22
	52	4.26	0.72	2469.26
	56	4.05	0.67	2588.39
F	60	3.77	0.24	2775.14

was chosen by the GA for its dominance with regards to the deterrence score. Indeed, the latter is better in the 48-agent solution (2296.49) than in the 47-agent one (2194.83).

These differences in spatial placement between configurations of a portfolio make it difficult to interpret trends resulting from the multi-objective GA. Instead, the portfolios provided in this thesis aim at informing policy makers on the potential performance they may expect from various deployment configurations. These configurations were identified by the GA for their ‘dominant’ character over others. Depending on a police agency’s number of available resources on a particular shift (e.g. 30 patrols), or on their current performance priority (e.g. minimising the percentage of ‘failed’ responses), they may wish to implement a particular configuration from the portfolio. In the case of DPD, for instance, the 13-agent configuration could, according to the model, yield an average response time of 9.13 mins, 1.57% ‘failed’ responses, and a deterrence score of 732.48. This configuration thus seems advantageous when response time and percentage of ‘failed’ responses are the main priorities, but resources are in short supply.

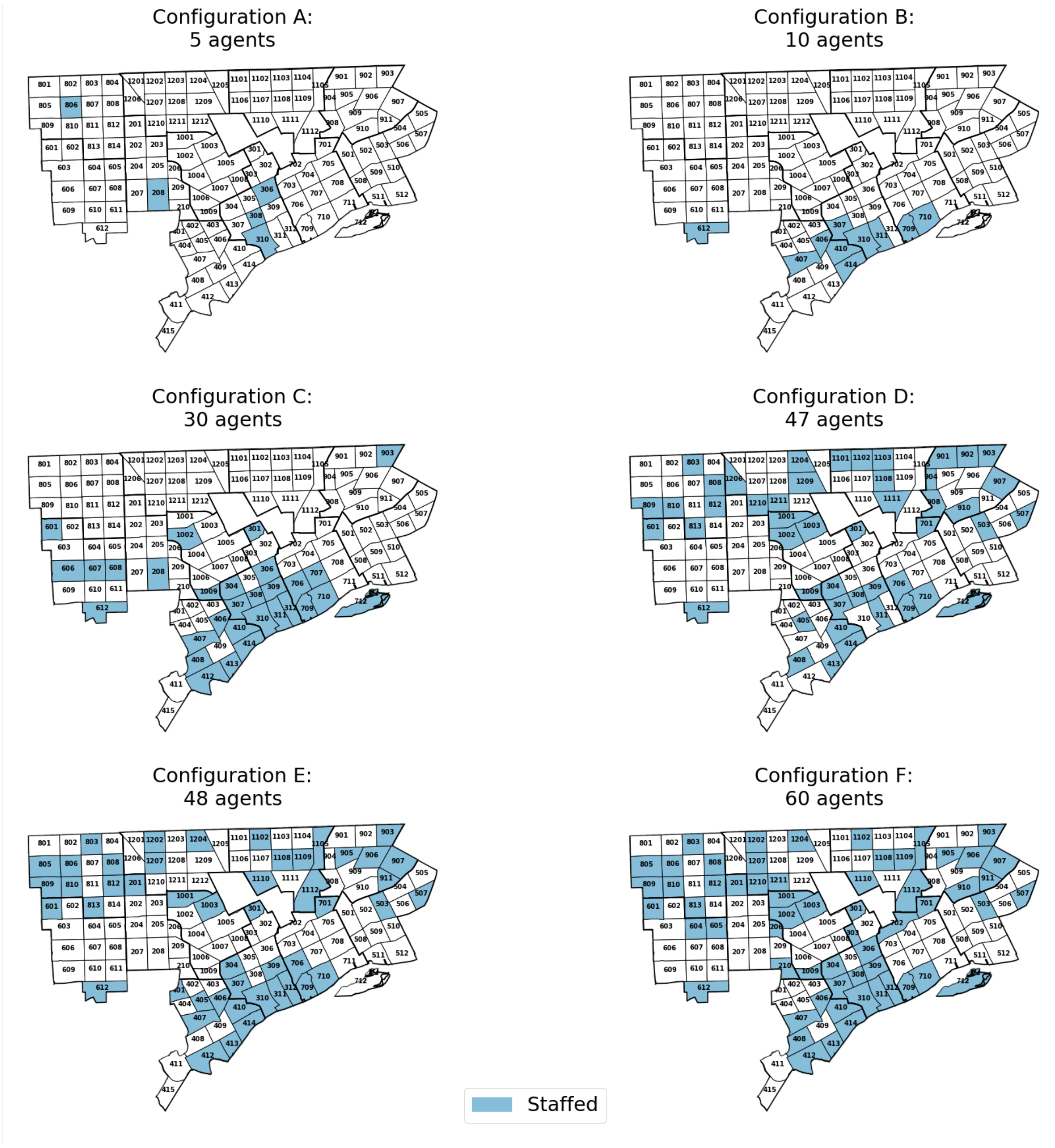

Figure 7.15: Example of the spatial placements for 6 configurations in the portfolio of 20 configurations prescribed by the GA for a low-demand scenario

High-demand scenario

Table 7.7 summarises the values of the three performance metrics for the 20 configurations prescribed under a high-demand scenario. Additionally, Figure 7.16 shows the spatial placements of some example configurations in this high-demand portfolio. According to the results, the prescribed 60-agent configuration may yield an average response time of 3.83 mins, a percentage of 'failed' responses of 0.36% and a deterrence score as high as 5040.38. This corresponds to almost twice the deterrence score that was yielded by the 60-agent configuration prescribed under the low-demand scenario. This result aligns with those obtained through the ABM experiments of Chapter 5 which revealed that agents are able to deter twice as much crime in a high-demand scenario because the density of historical crime on each segment is higher on these time periods. However, it is important to remind the reader that, unlike in the experiments of Chapter 5, the spatial placements of the 60-agent configurations prescribed for each scenario do not necessarily overlap and as such, their performance cannot be directly compared (see Figure 7.15 for low demand and Figure 7.16 for high demand).

Table 7.7: Performance metric values for 20 configurations in the portfolio of chosen solutions for the high-demand scenario

	Num. of agents	**Avg. resp. time (mins)**	**% 'failed' responses**	**Deterrence score**
	1	39.22	2.60	60.80
A	5	5.95	0.45	458.24
	8	32.42	9.20	684.06
	9	5.15	0.52	815.05
	13	18.84	5.84	1112.47
B	15	16.03	10.96	1324.49
	18	10.31	6.00	1585.13
	22	13.11	10.01	1924.14
	26	8.26	5.92	2231.78
	27	15.64	13.73	2361.79
C	31	6.13	3.88	2730.38
	33	5.58	2.50	3015.37
	38	6.24	4.16	3356.18
	39	7.80	6.25	3469.99
D	46	5.68	2.39	3870.10
	47	4.83	2.06	4080.92
	49	4.63	1.39	4114.12
E	51	4.50	1.43	4399.18
	54	3.92	0.36	4464.30
F	60	3.83	0.36	5040.38

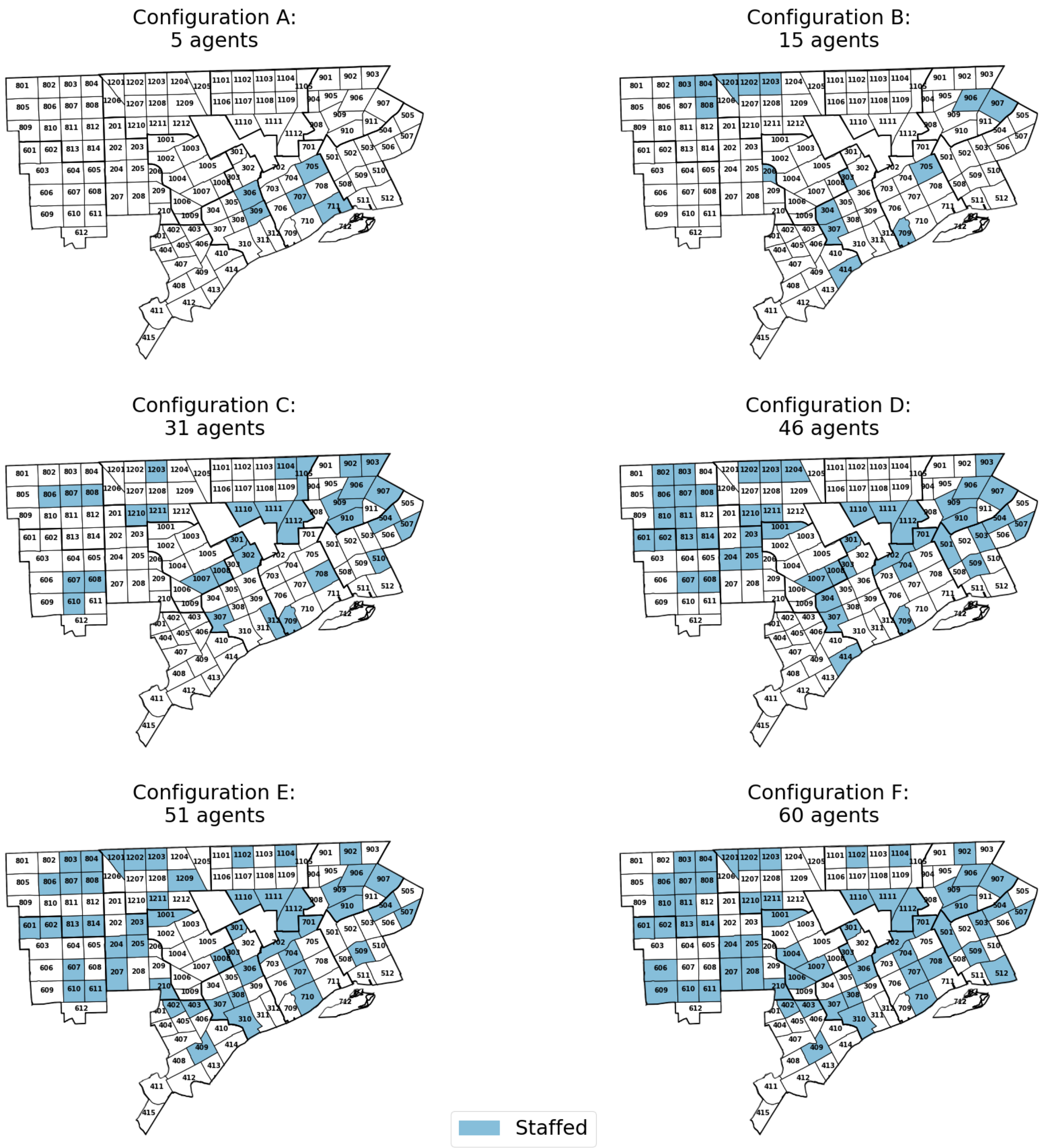

Figure 7.16: Example of the spatial placements for 6 configurations in the portfolio of 20 configurations prescribed by the GA for a high-demand scenario

7.4 Summary: applying the GAs to finding solutions to the PDOP in Detroit

This chapter provided the results of a single-objective and a multi-objective GA applied to the PDOP in the city of Detroit. The sole objective in the single-objective GA was to minimise the average response time. Results showed that, if left to learn for long enough, the GA converges towards an optimal solution (deployment configuration) featuring the maximum number of agents possible (60 in the case of Detroit). The deployment configuration prescribed by this single-objective GA was statistically significantly more effective in terms of response time than a deployment strategy devised using only historical CFS.

In the proposed multi-objective GA, additional metrics are included into the fitness function such as the percentage of 'failed' responses, the total deterrence score achieved by patrolling agents, and the number of deployed agents, which comes with a financial cost for real-world police agencies. Results from this multi-objective GA showed a steady learning process towards non-dominated solutions on the Pareto front. For a given demand scenario, the GA was able to identify a portfolio of 20 diverse solutions from which policy makers may choose depending on their specific priorities (e.g. a set number of patrols available to be deployed, minimising response time, maximising deterrence etc.).

The next chapter summarises the results of this thesis and highlights their implications for the modelling and optimising of police patrol deployment. The chapter will also discuss some limitations of the methodology and suggest potential avenues of investigations to take this research further.

Chapter 8

Discussion

This chapter concludes the thesis. It begins with Section 8.1 which summarises the main research findings and highlights the extent to which the project objectives – as outlined in Chapter 1, have been met. Several potential limitations of the research are discussed in Section 8.2, and a series of prospective avenues for further research are set out in Section 8.3. Finally, Section 8.4 provides concluding remarks.

8.1 Summary of research findings and their implications

A key challenge for police agencies concerns the strategic deployment of patrol units in order to provide an effective service at a minimal cost. As such, the overarching aim of this research was to develop a decision-support tool for supporting the design of efficient patrol deployment configurations that effectively deter crime while also providing timely response to incoming CFS.

Defining and formulating the Police Deployment Optimisation Problem

Patrol deployment is a complex issue due to the multiple interdependent responsibilities related to reactive and proactive policing. Past research on the topic of police deployment either included over-simplistic models of the patrol activities (namely equation-based models or ABMs with abstract environments), or considered only individual aspects of the police deployment problem (see Chapter 2 for a review of the literature). For instance, studies concerned with effectiveness focused on designing patrol routes to optimise crime deterrence, or on positioning a preset number of patrols to provide optimal CFS coverage. Other studies focusing solely on efficiency have sought to identify the minimum number of patrols required to achieve a desired service quality. This thesis addresses these gaps in past literature through proposing a new formulation of the Police Deployment Optimisation Problem (PDOP) which seeks to identify optimal deployment configuration(s) while also incorporating efficiency, reactive effectiveness and proactive effectiveness. Specifically, it considers the four following interconnected objectives:

- minimising the number of cars deployed (efficiency)
- minimising the average incident response time (reactive effectiveness)
- minimising the percentage of 'failed' responses (reactive effectiveness)
- maximising the deterrence score while patrolling (proactive effectiveness)

Through taking a more holistic and all-encompassing approach, this thesis aims to create a model that better represents the types of real-world deployment challenges faced by police agencies.

Designing and validating a high-fidelity model (agent-based model) of police patrol activities in which the performance of various deployment strategies can be accurately evaluated.

To explore solutions to the PDOP, this thesis implemented a novel simulation-based optimisation approach comprised of (1) a realistic ABM which acts as an evaluation tool for deployment configurations, and (2) a GA which performs an efficient search for solutions.

The ABM takes a deployment configuration as input and simulates the individual-level activities of motorised police patrols throughout their shift, including their movement along the road network as they patrol and respond to incidents. At the end of the simulation, the model returns a series of aggregated metrics (e.g. average response time, deterrence score etc.) to help assess the performance of the simulated deployment configuration. Elements of the ABM were informed through discussions with UK police and calibrated and validated using publicly available data from Detroit Police Department (Michigan), as an exemplar force.

Typically, evidence-based policing relies on the implementation of randomised field trials. As discussed in Chapter 2, while the importance of empirical experimentation cannot be overstated, field experiments are rarely implemented by police agencies for a number of reasons. First, some types of complex policing problems – such as the PDOP – present logistical challenges as they feature too many candidate solutions which cannot feasibly all be implemented in the field. Another shortcoming of field experiments is their inability to 'rewind time' to provide a controlled comparison of multiple strategies under the exact same conditions. For instance, one cannot compare the outcome of two different deployment strategies on the same evening shift.

In contrast, the ABM built in this thesis allows for the comparison of the simulated outcome of many deployment strategies by controlling for all other factors. It is the hope of this author that the evidence generated in these experiments could be considered useful for evidence-based policing, as suggested by other studies (Groff and Birks, 2008; Groff and Mazerolle, 2008). In this way, the ABM acts as a computational laboratory in which to cheaply and rapidly evaluate the performance of many alternative deployment configurations and explore their consequences free from logistical and ethical constraints.

Although other models of police deployment have been devised in the past, they have typically relied on simplifications that restricted their ability to accurately represent real-world police systems. The ABM used in this thesis however was built with high fidelity in mind, achieving a more accurate representation of real-world policing through two key considerations. First, it models the behaviour of police patrols at the individual patrol vehicle level – rather than providing a general force-wide approximation. Second, the ABM considers the real road network as part of a police force environment, rather than using an abstract grid-like representation. With these two considerations, the ABM is able to produce more accurate estimations of the performance of the system under a given patrol deployment configuration.

Applying the model to explore the outcome of various deployment designs for the case study of Detroit Police Department.

To test and illustrate the usefulness of the developed ABM, the model was used to conduct a series of deployment experiments for the case study of Detroit Police Department. These experiments explored the impact of various aspects of police deployment decisions (i.e. number of patrols, random deployment versus one targeted towards historical CFS) on system performance.

Results from these experiments showed that, under both a low and a high-demand scenario, increasing the number of deployed patrols significantly improves the performance of the system, both in terms of proactive and reactive effectiveness. Interestingly, the relationship between number of deployed patrols and average response time is not linear but instead shows signs of diminishing returns after deploying about 40 patrols for the low-demand scenario and 50 patrols for the high-demand one. In contrast, the relationship between number of patrols and crime deterrence score appears to be completely linear, as deploying more patrols leads to more idle time for each individual patrol.

Additionally, the experiments revealed that, when resources are stretched (high-demand and low-supply), a targeted patrol deployment based on historical CFS could yield a better reactive effectiveness compared with a random one (up to 40% faster responses and up to 6% fewer 'failed' responses). In a low-demand-high-supply scenario, on the other hand, a targeted deployment does not bring significant improvements in reactive effectiveness, yet leads to more crime deterrence as agents spend less time travelling and more time patrolling.

These results demonstrate that the ABM built in this thesis is able to inform police agencies on what performance they can expect with any particular patrol deployment. As such, the ABM tool has clear potential to assist police agencies in designing more effective and efficient deployment strategies, which would help address their dual goals of reducing crime whilst providing increased value to taxpayer funding.

Designing efficient metaheuristic algorithms (genetic algorithms) from which to derive high-quality solutions to the PDOP in an reasonable time.

As discussed in Chapter 7, the PDOP is a NP-hard problem for which it is impossible to exhaustively evaluate every single candidate solution. In order to automate and speed up the search for optimal solutions to the PDOP, this thesis employed a simulation-based optimisation approach. In this approach, a GA is used to generate new candidate solutions and select those which yield the best performance. The ABM is utilised by the GA to evaluate the performance of each candidate solution created during the search.

Two GA variants were built, each exploring a different version of the PDOP. First, a single-objective GA was developed which solely sought to minimise the average response time. This was followed by the development of a multi-objective GA seeking to optimise multiple conflicting objectives related to efficiency (number of agents), reactive effectiveness (average response time and percentage of 'failed' responses) and proactive effectiveness (deterrence score).

The GAs developed in this thesis allow for an efficient search of the parameter space to be conducted. This makes it possible to identify a set of good solutions to the PDOP in reasonable time. The benefit of this cannot be overstated; finding the best deployment configuration for only an average-sized metropolitan police force such as that of Detroit (population 639k) without this GA would take one quattuordecillion, four hundred tredecillion years of simulation runtime. Put another way, that is one hundred and one decillion, four hundred and forty-nine nonillion, two hundred and seventy-five octillion, and three hundred and sixty-two septillion times longer than the current estimated age of the universe. In contrast, using good computational resources this thesis' GAs can find the best deployment configuration in under 48 hours.

Applying the resulting optimisation tool to the case study of Detroit Police Department.

The simulation-based optimisation tool built in this thesis was applied to the case study of Detroit Police Department (Michigan). Results suggested that, when seeking to deploy 60 patrol units across the force, the single-objective GA is able to prescribe a deployment configuration which produces significantly better response times overall compared with one that is designed solely based on historical CFS data. This is because the optimal configuration identified by the GA prescribes a more balanced coverage of the force – with patrol units present in every precinct – ultimately leading to consistently lower response times. In contrast, the basic algorithm which devises configurations based on historical CFS demand operates in a greedy fashion by prioritising the staffing of beats that are historically hottest. As a result, this algorithm prescribes configurations in which entire areas of the force may be unstaffed. Consequently, this can yield excessively long response times for incidents taking place in those under-staffed precincts – which could have catastrophic implications in a real-world scenario.

The multi-objective GA applied to the PDOP in Detroit converged towards a shortlist of Pareto-optimal configurations. These suggested configurations provide trade-offs between various objectives related to efficiency, reactive effectiveness and proactive effectiveness. This could be of particular value to police forces, empowering them with a suite of applicable deployment options from which a solution can be chosen to best address current supply levels and/or policing priorities.

8.2 Limitations and methodological critiques

The results of this study should be interpreted in the context of several potential limitations of the methodology. These limitations, relating to both the ABM and the GAs, are summarised in this section.

8.2.1 Limitations of the ABM

Design simplifications

Although the ABM developed in this thesis is amongst the most comprehensive and realistic built to date for modelling the behaviour of police patrols, some necessary simplifications were made throughout its design.

First, agents in the model are solely tasked with responding to emergency calls. However, real patrol units spend an important proportion of their time responding to non-emergency calls. Studies have shown that between 80% and 90% of police calls are related to non-criminal behaviour complaints such as welfare or nuisance (Boulton et al., 2017; College of Policing, 2015; Hill and Paynich, 2014; Johnson et al., 2009). In 2015 for instance, mental health incidents were estimated to account for 20% of police time in England and Wales (College of Policing, 2015). People with mental health issues often end up calling the police more than once. Repeat or frequent callers can generate a disproportionately high level of demand. For example, the Metropolitan Police Service (MPS) receives each year about 13,000 calls from mental-health-specific premises such as hospitals and mental health suites (HMICFRS, 2018), 4,000 of which result in officers being dispatched. This represents a call every 40 minutes and a dispatch every two hours to repeat callers from mental-health-specific premises. Overall, by focusing solely on emergency incidents, the current version of the model may not provide a complete picture of the workload that non-emergency calls place on patrol officers. This simplification of police response was necessary in keeping the model within manageable complexity. Future versions may further improve the realism of the simulate system by allowing agents to also respond to non-emergency calls when available.

Second, the agents in the ABM are tasked with patrolling and responding to incidents. Although these tasks are central to their day-to-day shift, officers may have many intersecting responsibilities as policing is a complex and culturally specific process. Apart from patrolling and responding to calls, officers are also expected to help with finding missing persons, fulfil administrative work (e.g., report writing), engage in informal face-to-face interactions with citizens (e.g., casual encounters, public relations contacts), bringing offenders back to the police station, amongst other tasks (Stinson et al., 2014). Therefore, these tasks should arguably be taken into account in order to derive an even more realistic model of police patrol activities.

Nevertheless, while these assumptions are likely to have influenced model outcomes, they do not reduce the viability of the model as a decision-support tool nor its ability to support the design of patrol deployments.

A further assumption made in designing the model relates to the deterrent effect that the act of patrolling has on crime. In the model, patrolling agents drive at the maximum allowed speed limit on each road segment. According to Koper (1995), intermittent patrol of micro-hot spots (street segments or blocks) of 10-16 minutes at least every two hours extends deterrence. In the real world, patrols drive at a much slower speed, or stop the vehicle to maximise their visibility and deterrence effect. In addition, the deterrence score calculation in this thesis does not take into account residual deterrence – i.e. the continuing deterrent effect that police presence has on disorderly and criminal behaviour after police depart from a location (Sherman, 1990). Finally, the calculation does not consider that frequent re-visits to the same streets may yield diminishing returns (Williams and Coupe, 2017). Ultimately, although the deterrence score calculation in the model could be improved, the impact of patrolling on crime deterrence is a complex and much studied topic that is beyond the scope of this research.

Validation

The police system is an open system influenced by external factors which, for the most part, are unknown to the researcher building the model. As such, it has been argued that numerical models of open systems are impossible to validate (Oreskes et al., 1994). A model that would exactly reproduce real-world empirical data would be “suspicious” (Polhill and Salt, 2017). As a result, it is widely recognised that ABMs, like most simulation models, cannot be fully validated, in the sense of knowing that they are a completely adequate representation of some real system (Crols and Malleson, 2019; Groff et al., 2019).

As discussed in Chapter 5, an incident-level validation is simply impossible given the complexity of the interactions between patrols themselves and their environment. As such, the ABM built in this thesis was instead validated through a population-level validation approach using data from the city of Detroit. This approach compared the overall distribution of real dispatch and travel times in Detroit to those generated by the model. Such a process allowed us to establish with confidence that the simulation was appropriate for its intended use, i.e. to evaluate the performance of a given deployment configuration. However, this validation was performed in the specific context of the city of Detroit: an urban environment featuring a dense grid-like road network. While this was not tested in the research presented here, it would be beneficial to also validate the model on a police force in a more rural setting.

Overall, as with all simulation research, one should interpret model results with the acknowledgement that they can only seek to evaluate system outcomes rather than measure them with exactitude, as would be possible in a real-world experiment.

Computational cost

One obvious limitation of the ABM technique is its computational cost. As the scale of the model increases (either the size of the environment or the number of agents), so do the required computational resources (Bonabeau, 2002). Given the significant number of deployment configurations that need to be evaluated in the context of the PDOP, the use of ABM as the model of choice may render the tool computationally expensive.

Nonetheless, ABMs are much easier to interpret by social scientists and policy-makers, making it an attractive option for this thesis. Furthermore, the ABM approach is currently the most appropriate available technique for modelling dynamic and complex systems such as the police system. The technique allows for much more realistic models than equation-based alternatives to be built. Finally, as previously mentioned, supercomputing clusters and parallelisation techniques were leveraged in this research to limit computing time.

8.2.2 Limitations of the GA

Tuning

The outcome of the GA learning is inherently dependent on any number of hyper-parameters (not to be confused with ABM parameters), such as the population size, mutation rate, crossover rate, elitism, etc. Poorly chosen hyper-parameters could lead to worse performance than random search, and should thus be avoided. While we attempted to choose reasonable values for these search parameters, it is likely that the efficiency of the algorithm could be improved by undergoing a thorough tuning process of these parameters.

Computational cost

GAs are computationally expensive to run, especially for large police forces containing a high number of patrol beats. In the computer environment used in this research, which uses a supercomputing cluster (HPC) and parallelisation techniques, the GA learning took about 6.5 hours per demand scenario and the subsequent final evaluation phase for that scenario took approximately 11 hours. The computational capacity constraints limited the extent of the exploration of the parameter space. As such it is acknowledged that allowing the GA to learn for longer would lead to further explorations of the parameter space, ultimately leading to the identification of better solutions.

Additionally, GAs are stochastic in nature, which means that multiple runs with the same hyper-parameters are likely to generate different outcomes. To circumvent this issue, it is common practice to combine the outcomes of multiple runs of the same GA. Unfortunately, as computational resources were limited for this thesis, it was not possible to conduct multiple runs of the GAs. However, it is acknowledged that doing so would lead to a more robust search for optimal solutions.

8.3 Recommendations for further research

Many of the limitations due to model assumptions mentioned in the previous section are merely shortcomings of the current version of the ABM. Townsley and Birks (2008) argued that simulation experiments should incorporate complexity in an incremental fashion. The models presented in this thesis represent a proof of concept upon which subsequent iterations may built and expand. Some improvement suggestions for future versions of the model are now proposed.

8.3.1 Explore dispatching consequences

Inter-sector dispatching

The version of the ABM developed in this thesis features absolute district integrity (also called intra-sector dispatching). In such a setup, patrol units are not dispatched across district boundaries. If no unit is available in the district, the incident enters a queue until a unit becomes free. However, many police departments use relative district integrity. For instance, in a 1971 study of the New York Police Department more than half of all dispatches were inter-sector dispatches (between districts). Usually, it is the nearest patrol unit that is dispatched to each CFS incident, and in doing so, they may cross the district boundaries. Further versions of the model could investigate the impact of inter-sector dispatching on system performance.

Multi dispatch

The current ABM models the dispatching of a single agent to each incident. However, it is common for emergency incidents to require the dispatch of multiple patrol units. As such, improvements to the current ABM version may aim to incorporate the ability to dispatch multiple agents and explore how this affects the performance of the system over the course of a shift.

8.3.2 Explore staffing level consequences

It is common for police agencies to assign one-officer cars to low-crime areas and two-officer cars elsewhere, or one-officer cars during day shifts and two-officer cars during night shifts. Since the model agents represent police vehicles, an additional attribute could be added to them that describes the level of staffing of that vehicle. This would allow to explore and better understand the impact of various staffing levels on system costs (efficiency).

8.3.3 Include other objectives in the PDOP

Further research into the PDOP may consider additional objectives. For instance, coverage equity – i.e. the balance between the level of service provided in all districts of the force (Goldberg, 2004) – is an equally important metric. It may not be acceptable to have districts that are poorly serviced while others receive an outstanding police coverage (see Marsh and Schilling, 1994, for a review of coverage equity issues in ambulance and fire services).

Another key objective to police agencies is workload equity – i.e. the balance between officer workload across districts. First, an imbalanced system decreases officer moral (Goldberg, 2004). Second, if the patrol car or officer is always busy responding to a call when another incident occurs, a car from a neighbouring district would have to respond. This leads to a domino effect in which cars pulled from their assigned districts would leave the district unattended, and therefore more vulnerable to criminal incidents

This being said, the GAs built in this thesis appear to naturally prescribe homogeneous deployment configurations that avoid the under staffing of entire districts and thus provide a balanced coverage.

8.3.4 Explore future demand

This study has focused on identifying better deployment configurations under a low-demand and high-demand scenarios. These scenarios were based on historical demand. Further research could investigate how robust these configurations would be under future demand. For instance, police forces face growing pressure to prevent and respond to large-scale emergencies such as terrorist attacks. Despite these being rare, the police are expected to anticipate the pressure that such demand would place on the system (HMICFRS, 2018). As a computational laboratory, the ABM built in this thesis lends itself well to exploring such questions.

8.4 Concluding remarks

Ultimately, this thesis has created a novel decision-support tool for exploring the problem of patrol deployment optimisation. The approach chosen here is that of a simulation-based optimisation combining an agent-based model with a genetic algorithm. The tool harnesses the power of ABM to produce a comprehensive model of patrol activities that can be used to evaluate the performance of the system under various deployment configurations. Meanwhile, the developed GA efficiently guides the search for optimal solutions in a very wide parameter space, and is able to identify a portfolio of deployment configurations each with their own strengths from which policy makers may choose.

Through its flexible framework designed to be applied to any police agency, the developed decision-support tool shows considerable potential in informing more cost-effective patrol deployments – ones that better use the resources at hand whilst helping to keep the public safe.

Bibliography

Adenso-Díaz, B., & Rodríguez, F. (1997). A simple search heuristic for the MCLP: Application to the location of ambulance bases in a rural region. *Omega*, *25*(2), 181–187.

Alp, O., Erkut, E., & Drezner, Z. (2003). An Efficient Genetic Algorithm for the p-Median Problem. In *Annals of operations research* (Vol. 122, *1-4*, pp. 21–42).

Aly, A. A. (1979). Police briefing stations: A location problem. *AIIE Transactions*, *11*(1), 12–22.

Axelrod, R. M. (1997). *The complexity of cooperation : agent-based models of competition and collaboration*. Princeton, N.J.: Princeton University Press.

Axelrod, R. M. (2006). Agent-based Modeling as a Bridge Between Disciplines. In L. Tesfatsion & K. Judd (Eds.), *Handbook of computational economics* (Vol. 2, pp. 1565–1584).

Axtell, R. L., & Epstein, J. M. (1994). Agent-Based Modeling: Understanding Our Creations. *The Bulletin of the Santa Fe Institute*.

Baesler, F., Gatica, J., & Correa, R. (2015). Simulation Optimisation for Operating Room Scheduling. *International Journal of Simulation Modelling*, *14*(2), 215–226.

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

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

Bichler, G., Schmerler, K., & Enriquez, J. (2013). Curbing nuisance motels: An evaluation of police as place regulators. *Policing: An International Journal of Police Strategies & Management*, *36*(2), 437–462.

Birks, D. J., & Davies, T. (2017). Street network structure and crime risk: an agent-based investigation of the encounter and enclosure hypotheses. *Criminology*, *55*(4), 900–937.

Birks, D. J., & Townsley, M. (2018). Agent-based decision-support systems for crime scientists. In R. Wortley, A. Sidebottom, N. Tilley, & G. Laycock (Eds.), *Routledge handbook of crime science* (Chap. 23, p. 16).

Birks, D. J., Townsley, M., & Stewart, A. (2012). Generative explanations of crime: using simulation to test criminological theory. *Criminology*, *50*(1), 221–254.

Block, R. L., & Block, C. R. (1995). Space, place, and crime: Hot spot areas and hot places of liquor-related crime. In J. Eck & D. Weisburd (Eds.), *Crime and place: Crime prevention studies*. Willow Tree Pr.

BMG Research. (2014). *Public Attitude Survey 2013-14*. BMG Research.

Bodily, S. E. (1978). Police Sector Design Incorporating Preferences of Interest Groups for Equality and Efficiency. *Management Science*, *24*(12), 1301–1313.

Boeing, G. (2017). OSMnx: A Python package to work with graph-theoretic OpenStreetMap street networks. *The Journal of Open Source Software*.

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. In *Proceedings of the national academy of sciences of the united states of america (pnas)* (Vol. 99, *SUPPL. 3*, pp. 7280–7287).

Bosse, T., & Gerritsen, C. (2009). Comparing Crime Prevention Strategies by Agent-Based Simulation. In *2009 ieee/wic/acm international joint conference on web intelligence and intelligent agent technology* (pp. 491–496).

Bosse, T., & Gerritsen, C. (2010). Social Simulation and Analysis of the Dynamics of Criminal Hot Spots. *Journal of Artificial Societies and Social Simulation*, *13*(2).

Boulton, L., Mcmanus, M., Metcalfe, L., Brian, D., & Dawson, I. (2017). Calls for police service: Understanding the demand profile and the UK police response. *Police Journal: Theory, Practice and Principles*, *90*(1), 70–85.

Bowers, K. J., & Johnson, S. D. (2005). Domestic Burglary Repeats and Space-Time Clusters. *European Journal of Criminology*, *2*(1), 67–92.

Bradford, B., Jackson, J., & Stanko, E. A. (2009a). Contact and confidence: Revisiting the impact of public encounters with the police. *Policing and Society*, *19*(1), 20–46.

Bradford, B., Jackson, J., & Stanko, E. A. (2009b). Contact and confidence: Revisiting the impact of public encounters with the police. *Policing and Society*, *19*(1), 20–46.

Bradford, B., Stanko, E. A., & Jackson, J. (2009c). Using Research to Inform Policy: The Role of Public Attitude Surveys in Understanding Public Confidence and Police Contact. *Policing*, *3*(2), 139–148.

Braga, A. (2002). *Problem-oriented policing and crime prevention.* Monsey, NY: Criminal Justice Press.

Braga, A. A., & Weisburd, D. (2010). *Policing problem places: Crime hot spots and effective prevention.* New York, NY: Oxford University Press.

Braga, A. A. (2001). The Effects of Hot Spots Policing on Crime. *The ANNALS of the American Academy of Political and Social Science*, *578*(1), 104–125.

Braga, A. A. (2005). Hot spots policing and crime prevention: A systematic review of randomized controlled trials. *Journal of Experimental Criminology*, *1*(3), 317–342.

Braga, A. A., Weisburd, D. L., Waring, E. J., Mazerolle, L. G., Spelman, W., & Gajewski, F. (1999). Problem-oriented policing in violent crime places: a randomized controlled experiment*. *Criminology*, *37*(3), 541–580.

Brantingham, P. L., & Brantingham, P. J. (1982). Mobility, Notoriety, and Crime: A Study of Crime Patterns in Urban Nodal Points Environmental Criminology View project British Columbia Courts View project. *Journal of environmental systems*, *11*(1), 89–99.

Brantingham, P. L., & Brantingham, P. J. (1993). Environment, Routine, and Situation: Toward a Pattern Theory of Crime. In Ronald V. Clarke & Marcus Felson (Eds.), *Advances in criminological theory* (pp. 259–294).

Brantingham, P. J., & Brantingham, P. L. (1984). *Patterns in crime.* New York: Macmillan.

Brantingham, P. J., Dyreson, D. A., & Brantingham, P. L. (1976). Crime Seen Through a Cone of Resolution. *American Behavioral Scientist*, *20*(2), 261–273.

Brantingham, P. J., & Tita, G. (2008). Offender Mobility and Crime Pattern Formation from First Principles. In *Artificial crime analysis systems: Using computer simulations and geographic information systems* (pp. 193–208).

Bright, J. (1969). The Beat Patrol Experiment. *Home Office Police Research and Development Branch. Unpublished.*

Castiglione, F., Pappalardo, F., Bernaschi, M., & Motta, S. (2007). Optimization of HAART with genetic algorithms and agent-based models of HIV infection. *Bioinformatics*, *23*(24), 3350–3355.

Chaiken, J. M., & Dormont, P. (1978a). A Patrol Car Allocation Model: Background. *Management Science*, *24*(12), 1280–1290.

Chaiken, J. M., & Dormont, P. (1978b). A Patrol Car Allocation Model: Capabilities and Algorithms. *Management Science*, *24*(12), 1291–1300.

Chainey, S., Tompson, L., & Uhlig, S. (2008). The Utility of Hotspot Mapping for Predicting Spatial Patterns of Crime. *Security Journal*, *21*(1-2), 4–28.

Chawathe, S. S. (2007). Organizing hot-spot police patrol routes. In *Isi 2007: 2007 ieee intelligence and security informatics* (pp. 79–86).

Chen, H., Cheng, T., & Shawe-Taylor, J. (2018). A Balanced Route Design for Min-Max Multiple-Depot Rural Postman Problem (MMMDRPP): a police patrolling case. *International Journal of Geographical Information Science*, *32*(1), 169–190.

Chen, H., Cheng, T., & Wise, S. C. (2015). Designing daily patrol routes for policing based on ANT colony algorithm. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *2*, 103–109.

Chen, H., Cheng, T., & Wise, S. C. (2017). Developing an online cooperative police patrol routing strategy. *Computers, Environment and Urban Systems*, *62*, 19–29.

Chen, H., Cheng, T., & Ye, X. (2019). Designing efficient and balanced police patrol districts on an urban street network. *International Journal of Geographical Information Science*, 1–22.

Chiyoshi, F., & Galvão, R. D. (2000). A statistical analysis of simulated annealing applied to the p-median problem. *Annals of Operations Research*, *96*(1-4), 61–74.

Choi, S. P. M., Liu, J., & Chan, S.-P. (2001). A genetic agent-based negotiation system. *Computer Networks*, *37*(2), 195–204.

Chow, A. H. F., Cheung, C. Y., & Yoon, H. T. (2015). Optimization of Police Facility Locationing. *Transportation Research Record: Journal of the Transportation Research Board*, *2528*(1), 60–68.

Church, R. L., Corrigan, W., & Sorensen, P. (2001). Manpower deployment in emergency services. *Fire Technology*, *37*(3), 219–234.

Church, R. L., ReVelle, C., & Velle, C. R. (1974). The maximal covering location problem. In *Papers of the regional science association* (Vol. 32, *1*, pp. 101–118).

Clarke, R. V., & Harris, P. M. (1992). Auto Theft and Its Prevention. *Crime and Justice*, *16*, 1–54.

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences* (2nd Editio).

Cohn, E. G., & Rotton, J. (2000). Weather, seasonal trends and property crimes in Minneapolis, 1987-1988. A moderator-variable time-series analysis of routine activities. *Journal of Environmental Psychology*, *20*(3), 257–272.

College of Policing. (2015). *College of Policing analysis: Estimating demand on the police service.* College of Policing.

College of Policing. (2021). Introduction to vulnerability-related risk.

Cook, P. J. (1980). Research in Criminal Deterrence: Laying the Groundwork for the Second Decade. *https://doi.org/10.1086/449070*, *2*, 211–268.

Cordner, G. W. (1989). Police agency size and investigative effectiveness. *Journal of Criminal Justice*, *17*(3), 145–155.

Crols, T., & Malleson, N. (2019). Quantifying the ambient population using hourly population footfall data and an agent-based model of daily mobility. *GeoInformatica*, *23*(2), 201–220.

Curtin, K. M., Hayslett-McCall, K., & Qiu, F. (2010). Determining optimal police patrol areas with maximal covering and backup covering location models. *Networks and Spatial Economics*, *10*(1), 125–145.

D'Amico, S. J., Wang, S.-J., Batta, R., & Rump, C. M. (2002). A simulated annealing approach to police district design. *Computers & Operations Research*, *29*(6), 667–684.

Daskin, M. S. (1983). A Maximum Expected Covering Location Model: Formulation, Properties and Heuristic Solution. *Transportation Science*, *17*(1), 48–70.

Dau, P. M., Vandeviver, C., Dewinter, M., Witlox, F., & Vander Beken, T. (2021). Policing Directions: a Systematic Review on the Effectiveness of Police Presence. *European Journal on Criminal Policy and Research*, 1–35.

Davies, T. P., & Bishop, S. R. (2013). Modelling patterns of burglary on street networks. *Crime Science*, *2*(1), 1–14.

Davies, T., & Bowers, K. J. (2019). Patterns in the supply and demand of urban policing at the street segment level. *Policing and Society*.

Davies, T., & Johnson, S. D. (2015). Examining the Relationship Between Road Structure and Burglary Risk Via Quantitative Network Analysis. *Journal of Quantitative Criminology*, *31*(3), 481–507.

Davis, R. C. (2012). *Selected International Best Practices in Police Performance Measurement.* RAND Corporation. arXiv: arXiv:1011.1669v3

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms.* John Wiley & Sons.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *1917*, 849–858.

Eck, J. (1997). Preventing crime at places. In D. o. C. University of Maryland & C. Justice (Eds.), *Preventing crime: What works, what doesn't, what's promising* (pp. 17–62). Washington, DC: Office of Justice Programs, U.S. Department of Justice.

Eck, J. (2002). Preventing crime at places. In L. Sherman, D. Farrington, B. Welsh, & D. MacKenzie (Eds.), *Evidence-based crime prevention* (pp. 241–294). New York: Routledge.

Eck, J. E., Chainey, S., Cameron, J. G., Leitner, M., & Wilson, R. E. (2005). *Mapping Crime: Understanding Hot Spots.* U.S. Department of Justice Office of Justice Programs. Washington, DC.

Eck, J. E., & Liu, L. (2008). Contrasting simulated and empirical experiments in crime prevention. *Journal of Experimental Criminology*, *4*, 195–213.

Edleston, O. S. S. T., & Bartlett, L. (2012). A tabu search algorithm applied to the staffing roster problem of Leicestershire police force. *Journal of the Operational Research Society*, *63*(4), 489–496.

Eiben, A. E., & Smith, J. E. (2015). *Natural Computing Series: Introduction to Evolutionary Computing.*

Epstein, J. M. (1999). Agent-based computational models and generative social science. *Complexity*, *4*(5), 41–60.

Epstein, J. M., & Axtell, R. (1996). *Growing artificial societies: social science from the bottom up.* Brookings Institution Press.

Faccenda, J. F., & Tenga, R. F. (1992). A combined simulation/optimization approach to process plant design. In *Proceedings - winter simulation conference* (pp. 1256–1261).

Farrell, G., & Pease, K. (1994). Crime seasonality: Domestic Disputes and Residential Burglary in Merseyside 1988-90. *British Journal of Criminology*, *34*(4), 487–498.

Federal Bureau of Investigation - Uniform Crime Reporting (UCR) Program. (2019). Crime in the United States.

Felson, M., & Poulsen, E. (2003). Simple indicators of crime by time of day. *International Journal of Forecasting*, *19*(4), 595–601.

Fielding, N., & Innes, M. (2006). Reassurance policing, community policing and measuring police performance. *Policing and Society*, *16*(2), 127–145.

Fleming, J., & Grabosky, P. (2009). Managing the Demand for Police Services, or How to Control an Insatiable Appetite. *Policing*, *3*(3), 281–291.

Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., & Gagne, C. (2012). DEAP: Evolutionary Algorithms Made Easy. *Journal of Machine Learning Research*, *13*, 2171–2175.

Galán, J. M., Izquierdo, L. R., Izquierdo, S. S., Santos, J. I., del Olmo, R., López-Paredes, A., & Edmonds, B. (2009). Errors and Artefacts in Agent-Based Modelling. *Journal of Artificial Societies and Social Simulation*, *12*(1), 1.

Gendreau, M. (1997). Solving an ambulance location model by tabu search. *Location Science*, *5*(2), 75–88.

Geyik, F., & Dosdoğru, A. T. (2013). Process plan and part routing optimization in a dynamic flexible job shop scheduling environment: an optimization via simulation approach. *Neural Computing and Applications*, *23*(6), 1631–1641.

Gilbert, N., & Troitzsch, K. G. (2005). *Simulation for the Social Scientist second edition.* McGraw-Hill Education (UK).

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, *13*(5), 533–549.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning.* Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

Goldberg, D. E., & Holland, J. H. (1979). *Genetic Algorithms in Search, Optimization, and Machine Learning.*

Goldberg, J. M. (2004). Operations Research Models for the Deployment of Emergency Services Vehicles.

Gonçalves, I., & Silva, S. (2013). Balancing Learning and Overfitting in Genetic Programming with Interleaved Sampling of Training Data. In *Lecture notes in computer science* (Vol. 7831 LNCS, pp. 73–84).

Gonçalves, I., Silva, S., Melo, J. B., & Carreiras, J. M. (2012). Random sampling technique for overfitting control in genetic programming. In *15th european conference on genetic programming* (Vol. 7244 LNCS, pp. 218–229).

Green, L. (1984). A Multiple Dispatch Queueing Model of Police Patrol Operations. *Management Science*, *30*(6), 653–664.

Green, L., & Kolesar, P. (1984). The Feasibility of One-Officer Patrol in New York City. *Management Science*, *30*(8), 964–981.

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., ... DeAngelis, D. L. (2006). A standard protocol for describing individual-based and agent-based models. *Ecological Modelling*.

Grimm, V., Railsback, S. F., Vincenot, C. E., Berger, U., Gallagher, C., Deangelis, D. L., ... Ayllón, D. (2020). The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. *JASSS*.

Groff, E. R. (2007). 'Situating' Simulation to Model Human Spatio-Temporal Interactions: An Example Using Crime Events. *Transactions in GIS*, *11*(4), 507–530.

Groff, E. R. (2008). Characterizing the Spatio-Temporal Aspects of Routine Activities and the Geographic Distribution of Street Robbery. In *Artificial crime analysis systems: Using computer simulations and geographic information systems* (pp. 226–251).

Groff, E. R., & Birks, D. J. (2008). Simulating Crime Prevention Strategies: A Look at the Possibilities. *Policing*, *2*(2), 175–184.

Groff, E. R., Johnson, S. D., & Thornton, A. (2019). State of the Art in Agent-Based Modeling of Urban Crime: An Overview. *Journal of Quantitative Criminology*, *35*(1), 155–193.

Groff, E., & Mazerolle, L. (2008). Simulated experiments and their potential role in criminology and criminal justice. *Journal of Experimental Criminology*, *4*(3), 187–193.

Hansen, P., & Mladenović, N. (1997). Variable neighborhood search for the p-median. *Location Science*, *5*(4), 207–226.

Hawdon, J., & Ryan, J. (2003). Policing Tactics and Perceptions of Police Legitimacy. *Police Quarterly*, *6*, 469–491.

Hedstrom, P. (2005). *Dissecting the social: On the principles of analytical sociology.*

Helm, J. E., Lapp, M., & See, B. D. (2010). Characterizing an effective hospital admissions scheduling and control management system: A genetic algorithm approach. In *Proceedings of the 2010 winter simulation conference* (pp. 2387–2398).

Hill, B., & Paynich, R. (2014). *Fundamentals of crime mapping* (2nd ed.). Burlington, MA: Jones & Bartlett Learning.

Hipp, J. R., Bauer, D. J., Curran, P. J., & Bollen, K. A. (2004). Crimes of Opportunity or Crimes of Emotion? Testing Two Explanations of Seasonal Change in Crime. *Social Forces*, *82*(4), 1333–1372.

Hird, C., & Ruparel, C. (2007). *Seasonality in recorded crime: preliminary findings*. Home Office.

HMICFRS. (2018). *Policing and Mental Health: Picking Up the Pieces*. HMICFRS.

HMICFRS. (2019). *Public Perceptions of Policing in England and Wales 2018*. BMG Research.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: The University of Michigan Press.

Holland, J. H. (1984). Genetic Algorithms and Adaptation. In E. Selfridge, O.G. Rissland & M. Arbib (Eds.), *Adaptive control of ill-defined systems. nato conference series (ii systems science)* (pp. 317–333).

Holland, J. H. (1992). Genetic Algorithms. *Scientific American*, *267*(1), 66–73.

Holland, J. H. (2006). Studying complex adaptive systems. *Journal of Systems Science and Complexity*, *19*(1), 1–8.

Home Office. (2020). *Police workforce, England and Wales, 30 September 2019*. Home Office.

Home Office News Team. (2017). Fact sheet: Police funding for 2018/19 explained - Home Office in the media.

Hot spots of predatory crime: routine activities and the criminology of place. (1989). *Criminology*, *27*(1), 27–56.

Hunter, G. (2015). Fewest cops are patrolling Detroit streets since 1920s. Available from: https://bit.ly/38uRgg8.

Hunter, R., & Jeffrey, C. R. (1992). Preventing convenience store robbery through environmental design. In R. Clarke (Ed.), *Situational crime prevention: Successful case studies* (pp. 194–204). Albany, New York: Harrow and Heston.

Jennings, N. R. (2001). An agent-based approach for building complex software systems. *Communications of the ACM*, *44*(4), 35–41.

Johnson, S. D., & Bowers, K. J. (2014). Near Repeats and Crime Forecasting. *Encyclopedia of Criminology and Criminal Justice*, 3242–3254.

Johnson, S. D., Bowers, K., Birks, D., & Pease, K. (2009). Predictive Mapping of Crime by ProMap: Accuracy, Units of Analysis, and the Environmental Backcloth. In *Putting crime in its place* (pp. 171–198). New York, NY: Springer.

Karn, J. (2013). *Policing and crime reduction: the evidence and its implications for practice - The Police Foundation*. The Police Foundation. London.

Kelley, T. L. (1935). An Unbiased Correlation Ratio Measure. *Proceedings of the National Academy of Sciences of the United States of America*, *21*(9), 554–559.

Kelling, G. (1992). Measuring What Matters: A New Way of Thinking About Crime and Public Order. *City Journal*, 21–33.

Kelling, G. L., Pate, T., Dieckman, D., & Brown, C. (1974). *The Kansas City Preventive Patrol Experiment: A Summary Report*. Police Foundation. Washington DC: Police Foundation.

Kennedy, J., & Eberhart, R. (1994). Prognostic evaluation of abdominal echography in typhoid fever. *Giornale di Malattie Infettive e Parassitarie*, *46*(10), 1942–1948. arXiv: 9780201398298

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680.

Koper, C. S. (1995). Just enough police presence: Reducing crime and disorderly behavior by optimizing patrol time in crime hot spots. *Justice Quarterly*, *12*(4), 649–672.

Koper, C. S. (2014). Assessing the Practice of Hot Spots Policing: Survey Results From a National Convenience Sample of Local Police Agencies. *Journal of Contemporary Criminal Justice*.

Korytkowski, P., Wiśniewski, T., & Rymaszewski, S. (2013). An evolutionary simulation-based optimization approach for dispatching scheduling. *Simulation Modelling Practice and Theory*, *35*, 69–85.

Kwak, N. K., & Leavitt, M. B. (1984). Police Patrol Beat Design: Allocation Of Effort And Evaluation Of Expected Performance. *Decision Sciences*, *15*(3), 421–433.

Landau, S. F., & Fridman, D. (1993). The seasonality of violent crime: The case of robbery and homicide in israel. *Journal of Research in Crime and Delinquency*, *30*(2), 163–191.

Langdon, W. B. (2011). Minimising Testing in Genetic Programming. *RN*.

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

Larson, R. C. (1975). *Hypercube Queuing Model: Program Description*. New York City Rand Institute. New York: RAND Corporation.

Larson, R. C. (1978). Police Deployment—Introduction. *Management Science*, *24*(12), 1278–1279.

Laufs, J., Bowers, K. J., Birks, D. J., & Johnson, S. D. (2020). Understanding the concept of 'demand' in policing: a scoping review and resulting implications for demand management. *Policing and Society*, 1–24.

LeBeau, J. L. (2001). Mapping out hazardous space for police work. In A. Hirschfield & K. Bowers (Eds.), *Mapping and analysing crime data: Lessons from research and practice* (pp. 139–155). London: Taylor and Francis.

Leigh, J., Dunnett, S., & Jackson, L. (2019). Predictive police patrolling to target hotspots and cover response demand. *Ann Oper Res*, *283*, 395–410.

Li, X., & Yeh, A. G. (2005). Integration of genetic algorithms and GIS for optimal location search. *International Journal of Geographical Information Science*, *19*(5), 581–601.

Liu, Y., & Khoshgoftaar, T. (2004). Reducing overfitting in genetic programming models for software quality classification. *Proceedings of IEEE International Symposium on High Assurance Systems Engineering*, *8*, 56–65.

Macy, M. W., & Willer, R. (2002). From factors to actors: Computational Sociology and Agent-Based Modeling. *Annual Review of Sociology*, *28*, 143–166.

Malleson, N., Heppenstall, A. J., & See, L. (2010). Crime reduction through simulation: An agent-based model of burglary. *Computers, Environment and Urban Systems*, *34*(3), 236–250.

Mangla, C., Ahmad, M., & Uddin, M. (2021). Optimization of complex nonlinear systems using genetic algorithm. *International Journal of Information Technology*, *13*(5), 1913–1925.

Marchione, E., Johnson, S. D., & Wilson, A. (2014). Modelling Maritime Piracy: A Spatial Approach. *Journal of Artificial Societies and Social Simulation*, *17*(2).

Marsh, M. T., & Schilling, D. A. (1994). Equity measurement in facility location analysis: A review and framework and framework. *European journal of operational*, *74*(1), 1–17.

Mattila, V., & Virtanen, K. (2014). Maintenance scheduling of a fleet of fighter aircraft through multi-objective simulation-optimization. *Simulation*, *90*(9), 1023–1040.

McEwen, T., Ahn, J., Pendleton, J., Webster, B., & Williams, G. (2004). *Computer Aided Dispatch in Support of Community Policing, Final Report.*

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*(6), 1087–1092.

Metzger, K., & Booza, J. C. (2005). *Reality vs. Perceptions: An Analysis of Crime and Safety in Downtown Detroit.* (Archive) Michigan Metropolitan Information Center, Wayne State University Center for Urban Studies.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs* (3rd ed.).

Miller, H. F., & Knoppers, A. (1972). A Computer Simulation of Police Dispathcing and Patrol Functions. In *International symposium on criminal justice information and statistics systems proceedings* (pp. 167–176).

Mitchell, P. S. (1972). Optimal Selection of Police Patrol Beats. *Journal of Criminal Law, Criminology & Police Science*, *63*.

Mitchell, T. M. (1997). *Machine Learning.* McGraw-Hill Education.

Mladenović, N., Labbé, M., & Hansen, P. (2003). Solving the p-center problem with Tabu Search and Variable Neighborhood Search. *Networks*, *42*(1), 48–64.

Moore, M., & Braga, A. (2003). Measuring and improving police performance: The lessons of Compstat and its progeny. *Policing: An International Journal of Police Strategies & Management*, *26*, 439–453.

Mukhopadhyay, A., Zhang, C., Vorobeychik, Y., Tambe, M., Pence, K., & Speer, P. (2016). Optimal allocation of police patrol resources using a continuous-time crime model. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 9996 LNCS, pp. 139–158).

Murray, A. T., & Church, R. L. (1996). Applying simulated annealing to location-planning models. *Journal of Heuristics*, *2*(1), 31–53.

North, M. J., & Macal, C. M. (2007). *Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation.*

NPCC. (2017). *Better Understanding Demand - Policing the Future.* NPCC Performance Management Coordination Committee.

O'Neill, M., Vanneschi, L., Gustafson, S., & Banzhaf, W. (2010). Open issues in Genetic Programming. *Genetic Programming and Evolvable Machines*, *11*(3-4), 339–363.

Office for National Statistics. (2019). *Crime in England and Wales: year ending June 2019.* Office for National Statistics.

Oreskes, N., Shrader-Frechette, K., & Belitz, K. (1994). Verification, Validation, and Confirmation of Numerical Models in the Earth Sciences. *Science*, *263*(5147), 641–646.

Paul, R. J., & Chanev, T. S. (1998). Simulation optimisation using a genetic algorithm. *Simulation Practice and Theory*, *6*(6), 601–611.

Pease, K. (1991). The Kirkholt project: Preventing burglary on a British public housing estate. *Security Journal*, *2*(2).

Perry, J. D., & Simpson, M. E. (1987). Violent crimes in a city:Environmental Determinants. *Environment and Behavior*, *19*(1), 77–90.

Pham, D. T., & Karaboga, D. (2000). *Intelligent Optimisation Techniques.*

Polhill, G., & Salt, D. (2017). The Importance of ontological structure: Why validation by 'fit-to-data' is insufficient. *Understanding Complex Systems*, (9783319669472), 141–172.

Polvi, N., Looman, T., Humphries, C., & Pease, K. (1991). The time course of repeat burglary victimization. *The British Journal of Criminology*, *31*(4), 411–414.

Priest, T. B., & Carter, D. B. (1999). Evaluations of police performance in an african american sample. *Journal of Criminal Justice*, *27*(5), 457–465.

Pyle, G. F. (1976). Spatial and Temporal Aspects of Crime in Cleveland, Ohio. *American Behavioral Scientist*, *20*(2), 175–198.

Pyle, G. F., & Hanten, E. W. (1974). *The spatial dynamics of crime.* Chicago: Department Of Geography, University of Chicago.

Ratcliffe, J. H. (2002). Aoristic signatures and the spatio-temporal analysis of high volume crime patterns. *Journal of Quantitative Criminology*, *18*(1), 23–43.

Ratcliffe, J. H. (2004). The Hotspot Matrix: A Framework for the Spatio-Temporal Targeting of Crime Reduction. *Police Practice and Research*, *5*(1), 5–23.

Ratcliffe, J. H. (2015). Towards an Index for Harm-Focused Policing. *Policing*, *9*(2), 164–182.

Ratcliffe, J. H., Taniguchi, T., Groff, E. R., & Wood, J. D. (2011). The philadelphia foot patrol experiment: a randomized controlled trial of police patrol effectiveness in violent crime hotspots. *Criminology*, *49*(3), 795–831.

Rea, L. M., & Parker, R. A. (1992). *Designing and conducting survey research : a comprehensive guide.* San Francisco: Jossey-Bass Publishers.

Reeves, C., & Rowe, J. E. (2002). *Genetic Algorithms—Principles and Perspectives: A Guide to GA Theory.*

Reis, D., Melo, A., Coelho, A. L. V., & Furtado, V. (2006). Towards Optimal Police Patrol Routes with Genetic Algorithms. In *Intelligence and security informatics* (pp. 485–491).

Rengert, G. (1980). Spatial Aspects of Criminal Behavior. In D. E. Georges-Abeyie & K. D. Harries (Eds.), *Crime: A spatial perspective* (pp. 47–57).

ReVelle, C., & Swain, R. W. (1970). Central Facilities Location. *Geographical Analysis*, *2*(1), 30–42.

Rolland, E., Schilling, D. A., & Current, J. R. (1997). An efficient tabu search procedure for the p-Median Problem. *European Journal of Operational Research*, *96*(2), 329–342.

Rosser, G., Davies, T., Bowers, K. J., Johnson, S. D., & Cheng, T. (2017). Predictive Crime Mapping: Arbitrary Grids or Street Networks? *Journal of Quantitative Criminology*.

Sacks, S. R. (2000). Optimal Spatial Deployment of Police Patrol Cars. *Social Science Computer Review*, *18*(1), 40–55.

Sagovsky, A., & Johnson, S. D. (2007). When Does Repeat Burglary Victimisation Occur? *Australian & New Zealand Journal of Criminology*, *40*(1), 1–26.

Sasaki, S., Comber, A. J., Suzuki, H., & Brunsdon, C. (2010). Using genetic algorithms to optimise current and future health planning - the example of ambulance locations. *International Journal of Health Geographics*, *9*(1), 4.

Sherman, L. W. (1990). Police Crackdowns: Initial and Residual Deterrence. *Crime and Justice*, *12*, 1–48.

Sherman, L. W. (2013). The Rise of Evidence-Based Policing: Targeting, Testing, and Tracking. *Crime and Justice*, *42*(1), 377–451.

Sherman, L., & Weisburd, D. (1995). Does patrol prevent crime? The Minneapolis Hot Spots Experiment. In *Crime prevention in the urban community* (pp. 87–95).

Shi, Y., & Eberhart, R. (1998). Modified particle swarm optimizer. In *Proceedings of the ieee conference on evolutionary computation, icec* (pp. 69–73).

Silva, S., & Gonçalves, I. (2011). Experiments on Controlling Overfitting in Genetic Programming.

Skogan, W. G., & Frydl, K. (2004). *Fairness and effectiveness in policing : the evidence*. Washington DC: The National Academies Press.

Spelman, W., & Brown, D. (1984). *Calling the police: citizen reporting of serious crime*. U.S. Dept. of Justice, National Institute of Justice.

Srinivas, N., & Deb, K. (1994). Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, *2*(3), 221–248.

Steenbeek, W., & Weisburd, D. (2016). Where the Action is in Crime? An Examination of Variability of Crime Across Different Spatial Units in The Hague, 2001–2009. *Journal of Quantitative Criminology*, *32*(3), 449–469.

Stevens, J. M., Webster, T. C., & Stipak, B. (1980). Response Time: Role in Assessing Police Performance. *Public Productivity Review.*

Stinson, P. M., Brewer Jr, S. L., & Liederbach, J. (2014). *Lorain Police Department: A Study to Improve Patrol Deployment.*

Stonedahl, F., & Wilensky, U. (2011). Finding forms of flocking: Evolutionary search in ABM parameter-spaces. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 6532 LNAI, pp. 61–75).

Summers, L., & Johnson, S. D. (2017). Does the Configuration of the Street Network Influence Where Outdoor Serious Violence Takes Place? Using Space Syntax to Test Crime Pattern Theory. *Journal of Quantitative Criminology*, *33*(2), 397–420.

Sun, S. (2002). Measuring the relative efficiency of police precincts using data envelopment analysis. *Socio-Economic Planning Sciences*, *36*(1), 51–71.

Surkis, J., Gordon, G., & Hauser, N. (1970). Simulation in an urban system : the police emergency response system. *Socio-Economic Planning Sciences.*

Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation.* Wiley.

Taylor, P. E., & Huxley, S. J. (1989). A Break from Tradition for the San Francisco Police: Patrol Officer Scheduling Using an Optimization-Based Decision Support System. *Interfaces*, *19*(1), 4–24.

ten Broeke, G., van Voorn, G., & Ligtenberg, A. (2016). Which sensitivity analysis method should i use for my agent-based model? *JASSS*, *19*(1).

Tompkins, G., & Azadivar, P. (1995). Genetic algorithms in optimizing simulated systems. In *Winter simulation conference proceedings, 1995.* (pp. 757–762).

Tompson, L., & Bowers, K. J. (2013). A Stab in the Dark?: A Research Note on Temporal Patterns of Street Robbery. *Journal of Research in Crime and Delinquency*, *50*(4), 616–631.

Townsley, M., & Birks, D. J. (2008). Building better crime simulations: systematic replication and the introduction of incremental complexity. *Journal of Experimental Criminology*, *4*(3), 309–333.

Townsley, M., & Johnson, S. D. (2008). The Need for Systematic Replication and Tests of Validity in Simulation. In L. Liu & J. Eck (Eds.), *Artificial crime analysis systems* (pp. 1–18).

Vaughan, A., Wuschke, K. E., Hewitt, A. N., Hodgkinson, T., Andresen, M. A., Brantingham, P., & Verdun-Jones, S. (2018). Variations in Mental Health Act calls to police: an analysis of hourly and intra-week patterns. *Policing*, *41*(1), 58–69.

Wang, X., Liu, L., & Eck, J. E. (2008). Crime Simulation Using GIS and Artificial Intelligent Agents. In *Artificial crime analysis systems: Using computer simulations and geographic information systems* (pp. 209–225).

Weisburd, D. (2005). Hot Spots Policing Experiments and Criminal Justice Research: Lessons from the Field. *Annals of the American Academy of Political and Social Science*, *599*, 220–245.

Weisburd, D. (2015). The law of crime concentration and the criminology of place*. *Criminology*, *53*(2), 133–157.

Weisburd, D., Braga, A. A., Groff, E. R., & Wooditch, A. (2017). Can hot spots policing reduce crime in urban areas? An agent-based simulation. *Criminology*, *55*(1), 137–173.

Weisburd, D., & Eck, J. E. (2004). What Can Police Do to Reduce Crime, Disorder, and Fear? *Annals of the American Academy of Political and Social Science*.

Weisburd, D., & Green, L. (1994). Defining the street-level drug market. In D. MacKenzie & C. Uchida (Eds.), *Drugs and crime: Evaluating public policy initiatives* (pp. 61–76). Thousand Oaks, CA: Sage Publications.

Weisburd, D., Maher, L., & Sherman, L. W. (1992). Contrasting crime general and crime specific theory: The case of hot spots of crime. *Advances in Criminological Theory*, *4*.

Wilcox, P., & Eck, J. E. (2011). Criminology of the unpopular. *Criminology & Public Policy*, *10*(2), 473–482.

Williams, S., & Coupe, T. (2017). Frequency Vs. Length of Hot Spots Patrols: a Randomised Controlled Trial. *Cambridge Journal of Evidence-Based Policing*, *1*(1), 5–21.

Winsor, T. P. (2019). *State of Policing: The Annual Assessment of Policing in England and Wales 2018*. HMICFRS. London.

Wise, S. C., & Cheng, T. (2016). How Officers Create Guardianship: An Agent-based Model of Policing. *Transactions in GIS*, *20*(5), 790–806.

Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: theory and practice. *The Knowledge Engineering Review*, *10*(2), 115–152.

Zaki, A. S., Cheng, H. K., & Parker, B. R. (1997). A Simulation Model for the Analysis and Management of An Emergency Service System. *Socio-Economic Planning Sciences*.

Zhang, Y., & Brown, D. E. (2013). Police patrol districting method and simulation evaluation using agent-based model & GIS. *Security Informatics*, *2*(1), 7.

Zhao, J., Schneider, M., & Thurman, Q. (2002). The effect of police presence on public fear reduction and satisfaction: A review of the literature. *The Justice Professional*, *15*, 273–299.