ABSTRACT

Title of Document: INTEGRATED MANGEMENT OF EMEREGENCY VEHICLE FLEET

Saini Yang, Doctor of Philosophy, 2006

Directed By: Professor Ali Haghani
Department of Civil and Environmental Engineering

The growing public concerns for safety and the advances in traffic management systems, that have made the availability of real-time traffic information a reality, have created an opportunity to build integrated decision support systems that can improve the coordination and sharing of information between agencies that are responsible for public safety and security and transportation agencies to provide more efficient Emergency Response Service.

In an Emergency Response System, reduction of the duration of response time can yield substantial benefits. The response time plays a crucial role in minimizing the adverse impacts: fatalities and loss of property can be greatly reduced by reducing the response time for emergencies. In this dissertation, we have developed an integrated model that can assist emergency response fleet dispatchers in managing the fleet. This model can help reduce the response time and improve service level by specifically accounting for the following:

- Vehicle Deployment: given real-time information about the status of the emergency response fleet, traffic information and the status of emergency calls, select proper fleet assignment schemes that satisfy various operation requirements.
- Vehicle Routing: given real-time traffic information, provide real-time route guidance for drivers of dispatched vehicles. This goal is achieved by applying various shortest path algorithms into the solution procedure.
- Planning and Evaluation: given the status of the fleet and the frequency of emergency calls in various areas of a region, the model can help evaluate the performance of the current system and help plan for potential sites for the relocation of vehicles and allocate an appropriate fleet of vehicles to these sites.

The vehicle deployment problem is formulated as an integer optimization problem. Since this problem has been shown to be NP-hard and because of the nature of emergency response, we developed heuristics which can provide quality solutions with short computational times. Several test algorithms are proposed to solve the emergency response vehicle deployment problem. Different methods for obtaining lower bounds for the value of objective function are analyzed in this dissertation. To evaluate the performance of the system under various scenarios, a simulation model is developed. The simulation system is calibrated based on real-world data. The results of simulation and analysis show the proposed system can effectively improve the emergency response service level. Application of this model in facility allocation illustrates its usage in other relevant operational scenarios.

INTEGRATED MANAGEMENT OF EMERGENCY VEHICLE FLEET

By

Saini Yang

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2006

Advisory Committee:
Professor Ali Haghani, Chair
Professor Martin Dresner
Professor Steven Gabriel
Professor Elise Miller-Hooks
Professor Paul Schonfeld

Dedication

This work is dedicated to my beloved husband,

my parents, and to the memory of my grandparents

Acknowledgements

I would like to express my sincere appreciation to my advisor, Dr. Ali Haghani, for his guidance, support and encouragement over the years. His understanding and patience and valuable advice have been the keys to the success of this work. Also a special gratitude is acknowledged to Dr. Paul Schonfeld for his helpful instructions and extensive suggestions through this study. Thanks are extended to the other committee members Dr. Kelly Clifton, Dr. Elise Miller-Hooks and Dr. Martin Dresner for their constructive comments in this research.

I am grateful to the CapWIN program, for funding this work, and to the Arlington County Fire Department, for providing real-world data on emergency response. I would like to thank my friends, M. Hamedi, P, Lin, and Y. Luo, for their support and help throughout all these years.

Finally, with deep sense of gratitude, I especially want to thank my husband, my parents, my brother, and my grandparents who first encouraged me to pursue the PhD degree. Without their tremendous love, I would not have been able to concentrate on my study and endure some difficult times through all these years.

Table of Contents

Dedication ii
Table of Contents iv
List of Tables vii
List of Figures viii
Chapter 1: Introduction 1
1.1 Emergency 1
1.1.1 Emergency Response Service 2
1.1.2 Emergency Response Time 3
1.2 Problem Statement 5
1.3 Research Scope and Tasks 9
1.4 Research Approach 11
1.5 Organization of the Dissertation 12
Chapter 2: Literature Review 13
2.1 Introduction 13
2.2 Modeling Approaches 14
2.2.1 Dispatching Problem 15
2.2.2 Emergency Vehicle Location Problem 19
2.2.3 Emergency Vehicle Relocation Problem 30
2.3 A Brief Review of Heuristics 33
2.3.1 Simulated Annealing 34
2.3.2 Genetic Algorithm 35
2.3.1 Tabu Search 38
2.4 Simulation 40
2.5 Related Issues 43
2.5.1 Shortest Path Algorithms 44
2.5.2 Short Term Travel Time Prediction 49
2.6 Trend of Research and Application 49
Chapter 3: Problem Statement and Mathematical Model 52
3.1 Problem Statement 52
3.2 Dynamic Characteristic 58
3.3 Components and Properties 61
3.3.1 Emergency Response Vehicle Fleet 61
3.3.2 Emergency Calls 63
3.3.3 Assumptions and Constraints 64
3.4 Mathematical Model 68
3.4.1 Notation 68
3.4.2 A Basic Dispatching Model 72
3.4.3 An Extension of the Dispatching Model 73
3.4.4 An Extension of the Relocation Problem 76
3.4.5 A Deployment Model 77
3.5 Formulation and Algorithmic Considerations 78
Chapter 4: Solution Approaches and Heuristics 80

4.1 Rolling Horizon Approach 80
4.1.1 Framework 80
4.1.2 Size of Variables and Computational Time 81
4.2 Initial solution 86
4.2.1 Initial Solution 1: CPLEX based algorithm 86
4.2.2 Reduction of Problem Size 88
4.2.3 Greedy Search for Vehicle Relocation 91
4.2.4 Initial Solution 2: Greedy Search Algorithm 95
4.3 Improvement Methods 97
4.3.1 SWAP move 97
4.3.2 The framework of Tabu Search 98
4.3.4 ADD Move 100
4.3.4 DROP Move 101
4.3.6 Search Strategy 104
4.3.7 Neighborhoods, Tabu Type and Tabu Tenure 105
4.4 Conclusions 106
Chapter 5: Lower Bounds 108
5.1 Analysis of Lower Bound Algorithms 108
5.2 Linear Relaxation Method 109
5.2 Decomposition 110
5.3 Lagrangian Relaxation Method 112
5.4 Upper Bounds 114
5.5 Solution Analysis 114
5.5.1 Analysis of Computational Times 115
5.5.2 Comparison with Lower Bounds 116
5.5.3 Analysis of Parameters in Heuristics 118
5.6 Conclusions 122
Chapter 6: Simulation Model 123
6.1 System Module 123
6.1.1 System Framework 123
6.1.2 Dispatching Strategies 125
6.1.3 Time Control Scheme 128
6.1.4 Status Update 130
6.2 Essential Modules 134
6.2.1 Vehicle Module 134
6.2.2 Emergency Module 134
6.2.3 Optimization Module 135
6.2.4 Travel Time Module 137
6.2.5 Statistics Module 140
6.3 Simulation Frameworks 141
6.4 Input Analysis 147
6.4.1 Selection of Probability Models 147
6.4.2 Random Number Generator 151
6.5 Output Analysis 151
6.5.1 Length of Warm-up Period 154
6.5.2 Independent Replications 156

6.5 Conclusions 159
Chapter 7: Case Study 160
7.1 Background 160
7.2 Current Operations 160
7.2.1 Street Network 160
7.2.2 Operational Data 162
7.3 Case Study 169
7.3.1 Comparison of Dispatching Strategies 169
7.3.2 Comparison of Shortest Path Algorithm 172
7.4 Sensitivity Analysis 174
7.4.1 The System Workload 174
7.4.2 Penalty Parameters 178
7.4.3 Benefit Threshold Sensitivity Analysis 180
7.4.4 Depot Locations and the Fleet Allocation 181
7.5 Conclusions 185
Chapter 8: Conclusions and Future Research 187
8.1 Summary and Conclusions 187
8.2 Future Research 191
8.1.1 Lower Bound Method 192
8.1.2 Heuristic Algorithms 193
8.1.3 Others 195
Appendix I 197
Appendix II 206
Bibliography 224

List of Tables

Table 1-1: Statistics of Fire in United States 2
Table 2-1: Comparison of the Three Modern Heuristics 40
Table 3-1: NFPA Guidelines, Response Times 63
Table 4-1: Experiments for CPLEX Computational Times.. 84
Table 4-2: Comparison of CPLEX Computational Times of *M3* and *M3*’................ 90
Table 5-1: Comparison of Optimal Solutions and Lower Bounds Obtained with Linear Relaxation... 112
Table 5-2: Comparison of Computational Time... 116
Table 5-3: Comparison of Solution Quality... 117
Table 5-4: Comparison of Solutions under Variable Stop Criteria........................... 120
Table 5-5: Comparison of Solutions under Variable Time Contours 121
Table 6-1: Summary of Simulation Timing and Relevant Events............................ 129
Table 6-2: Change of Status for Emergency Calls... 131
Table 6-3: Change of Status for Ambulance.. 132
Table 6-4: Change of Status for Fire Engine 133
Table 6-5: Change of Status for Police Car 133
Table 6-6: Reference of Solver Selection 137
Table 6-7: Results for 10 Independent Replications.. 154
Table 7-1: Histogram of Link Lengths 162
Table 7-2: Vehicle Dispatching by Emergency Type.. 164
Table 7-3: Summary of Number of Dispatched Vehicles... 165
Table 7-4: Distribution Estimation Results of Emergency Inter-arrival Time 166
Table 7-5: Distribution Estimation Results of Vehicle travel time........................... 167
Table 7-6: Comparison of Dispatching Strategies 171
Table 7-7: Comparison of Shortest Path Algorithms Based on Average Response Time 173
Table 7-8: Sensitivity Analysis of System Workload.. 177
Table 7-9: Sensitivity Analysis of Parameters under the Deployment Strategy....... 180

List of Figures

Figure 1-1: Components of Emergency Response System 3
Figure 1-2: Emergency Response Time 4
Figure 1-3: The Routine of Emergency Response 7
Figure 1-4: The Structure of an Emergency Response System. 8
Figure 3-1: Sample Response Area 54
Figure 3-2: A Simple Example of Dispatching and Routing Problem 55
Figure 3-3: A Simple Example of Coverage Problem 56
Figure 3-4: A Simple Example of Coverage Problem 57
Figure 4-1: A Sample Network for Analyzing Computational Time 83
Figure 4-2: An Example of Size Reduction of Candidate Relocation Sites 89
Figure 4-3: An Example of the Utility of a Move 92
Figure 4-4: Flow Chart for Initialization Solution I 94
Figure 4-5: Flow Chart for Initialization Method II 96
Figure 4-6: Flow Chart for SWAP Improvement Method 98
Figure 4-7: Generic Framework of Tabu Search Heuristics 99
Figure 4-8: Flow Chart for ADD Move 101
Figure 4-9: Flow chart for DROP move 103
Figure 6-1: Framework of the Simulation Model 125
Figure 6-2: Link Travel Time Prediction 139
Figure 6-3: Simulation Flow Chart of FCFS 143
Figure 6-4: Simulation Flow Chart of Nearest Origin 144
Figure 6-5: Simulation Flow Chart of Flexible Assignment 145
Figure 6-6: Simulation Flow Chart of Deployment Only 146
Figure 6-7: Test Example for Output Analysis. 153
Figure 6-8: An Example of Simulation Throughput. 156
Figure 6-9: Averaged Process for Simulation Throughput 156
Figure 6-10: An Example of Batch Mean. 158
Figure 7-1: A Real Street Network 161
Figure 7-2: Comparison of Real Service Time and Fitted Model: Inter-arrival Time 166
Figure 7-3: Comparison of Real Service Time and Fitted Model: Travel Time 167
Figure 7-4: Comparison of Real Service Time and Fitted Model: Service Time 169
Figure 7-5: Response Time in 24 Hours 173
Figure 7-6: Comparison of Average Response Time 176
Figure 7-7: Comparison of Longest Response Time 176
Figure 7-8: Comparison of Under-performed Percentages under 10 Parameter Scenarios 179
Figure 7-9: Impacts of Diversion Benefit Threshold on Performance Measures 181
Figure 7-10: Impacts of Fleet Allocation on Average Response Time 182
Figure 7-11: Impacts of Number of Depots on Average Response Time 183
Figure 7-12: Impact of Fleet Size on Average Response Time (3 Depots) 184
Figure 7-13: Impact of Fleet Allocation on Average Response Time (3 Depots) ... 184
Figure II-1: An Example of Chromosome 208
Figure II-2: An Example of Crossover Operation 211

Figure II-3: An Example of Mutation Operation 213
Figure II-4: The Sample Street Network 215
Figure II-5: Depot Locations without Cost Constraints.. 216
Figure II-6: A Large Street Network 217
Figure II-7: Locations of Depot with Cost Constraint and Time Coefficient = 10... 218
Figure II-8: Locations of Depot with Cost Constraint and Time Coefficient = 8..... 219
Figure II-9: Locations of Depot with Cost Constraint and Time Coefficient = 7..... 220
Figure II-10: Locations of Depot with Cost Constraint and Time Coefficient = 5 .. 221
Figure II-11: Locations of Depot with Cost Constraint and Time Coefficient = 3 .. 222
Figure II-12: Comparison of ART of Heuristic Solutions and Real Configurations 223

Chapter 1: Introduction

1.1 Emergency

Emergencies happen in various forms, including earthquakes, hurricanes, terrorist attacks, traffic accidents, arsons, personal sicknesses, etc. In 2001, in the US, fires alone caused an estimated direct property loss of $10.6 billion. More than 1.7 million fires were reported in one year and an estimated 45,500 intentionally set structural fires resulted in 3,745 civilian deaths and 20,300 civilian injuries. 84 percent of fire deaths occurred in residences. The fire death rate in 2001, including the results of September 11 event, is 22.1 civilian deaths per million. The U.S. has one of the highest fire death rates in the industrialized world.

As shown in Table 1-1, according to the fire statistics reported on the official website of the US Fire Administration, the number of fires has decreased slowly in the past 10 years; however, the value of the assets lost have increased constantly.

The surge in public concerns for safety and the advances in traffic management systems, improvements in communication systems, and the availability of real-time traffic information have created an opportunity for designing integrated decision support systems that improve the coordination and sharing of information between agencies that are responsible for public safety and security and transportation agencies to provide more efficient emergency response service.

Table 1-1: Statistics of Fire in United States

Year	# of Fires	# of Deaths	# of Injuries	Direct Dollar Loss In Millions
1995	1,965,500	4,585	25,775	$9,182
1996	1,975,000	4,990	25,550	$9,406
1997	1,795,000	4,050	23,750	$8,525
1998	1,755,000	4,035	23,100	$8,629
1999	1,823,000	3,570	21,875	$10,024
2000	1,708,000	4,045	22350	$11,207
2001[1]	1,734,500	3,745	20,300	$10,583
2001[2]	1,734,500	2,451	800	$33,440
2002	1,687,500	3,380	18,425	$10,337
2003	1,584,500	3,925	18,125	$12,307

[1] Excludes the events of September 11, 2001.

[2] These estimates reflect the number of deaths, injuries and dollar loss directly related to the events of September 11, 2001.

1.1.1 Emergency Response Service

Emergency response is the implementation of processes that are in place as a result of planning and preparedness actions, and application of resources that must be utilized to mitigate consequences and recover from an emergency. As shown in Figure 1-1, this procedure involves many essential components, as well as real-time information from transportation agencies, agencies that are responsible for public security and hospitals. The coordination between these components directly influences the effectiveness of activities involved in emergency response. For example, the use of real-time traffic information in guiding emergency vehicles to less congested routes may reduce the travel time to emergency locations and reduce the loss of property and save lives.

This dissertation focused on developing an emergency response vehicle fleet management system that can help integrate the information from different agencies and coordinate the activities of fire and rescue personnel, paramedics, and police so that the overall efficiency of the emergency response service is improved.

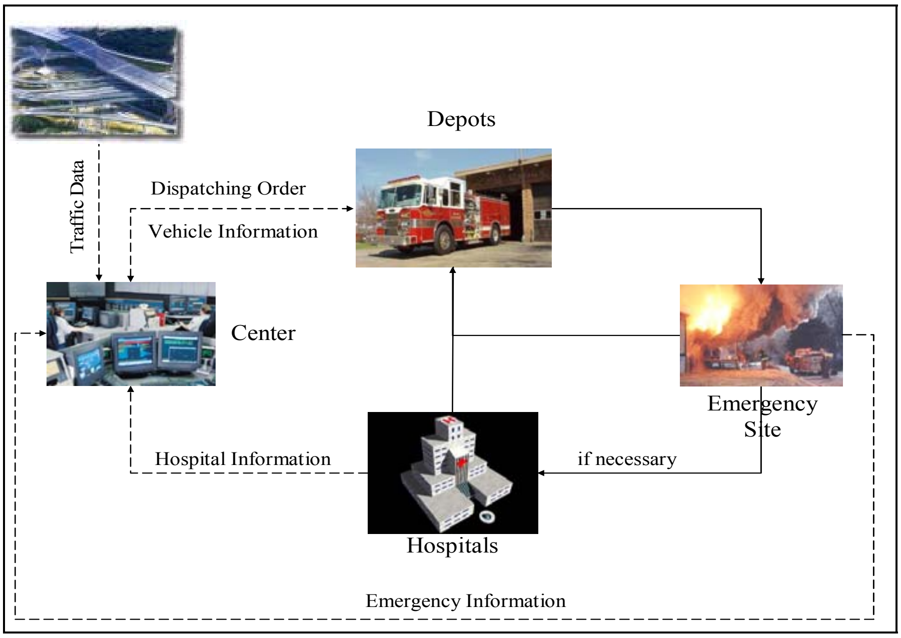


Figure 1-1: Components of Emergency Response System

1.1.2 Emergency Response Time

As the Fire Protection Handbook (2003) states: To provide effective service, crews must respond in a minimum amount of time after the incident has been reported and with sufficient resources to initiate fire, rescue, or emergency medical services. The duration of an emergency can be divided into 4 phases: detection time, preparation time, travel time and treatment time (see Figure 1-2). Response time is defined as the duration from the time an emergency call arrives at the station to the time an emergency vehicle arrives at the scene, which is the sum of preparation time and travel time.

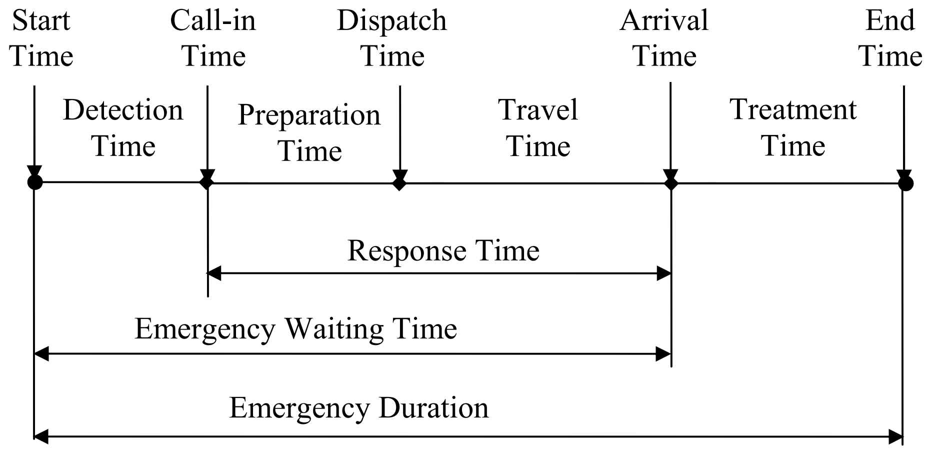


Figure 1-2: Emergency Response Time

Utilizing precise information and reducing the response time can effectively minimize the negative impacts. For example, sudden cardiac arrest (SCA) is the most common critical emergency needing prompt medical intervention. It strikes more than 350,000 people a year in the U. S. About 90% of those treated within two minutes survive; while only about 10% survive if treated after 6 minutes (R. R. Bowman, 1997). As a result of construction innovation, fast response time in fire emergencies is much more important. In the mid-1950s, the average flashover point (the temperature point at which the heat in an area or region is high enough to ignite all flammable material simultaneously) in burning structures was reached in about 12 to 15 minutes. By the mid-1990s, however, the flashover point was reached in about 3-7 minutes. The reason for this decrease in time is the increased fire loading through the use of plastics and other polycarbon materials in furniture and fittings, and increased insulation that has the effect of keeping the heat in (similar to an oven) and increases the likelihood of backdraft. These changes in construction materials and furnishings have also caused temperatures at the ceiling level to rise from 750 degrees centigrade in the mid-1950s to 1,100 degrees centigrade in the mid-1990s (Commission on Fire Accreditation International (2000)). These materials have also increased the volumes of toxic gases such as hydrogen cyanide, hydrogen chloride, and carbon monoxide produced in structure fires. As a result of the hotter fires that reach the flashover point faster, fire and rescue personnel are faced with a reduced time frame to undertake rescue efforts and to suppress fires.

Preparation time and travel time are noticeably affected by the availability of vehicles, traffic situation, and allocation of fleet to facilities. Therefore, the proposed emergency response vehicle fleet management system should be able to optimize the operations in order to improve service performance and reduce fatalities and loss of property.

1.2 Problem Statement

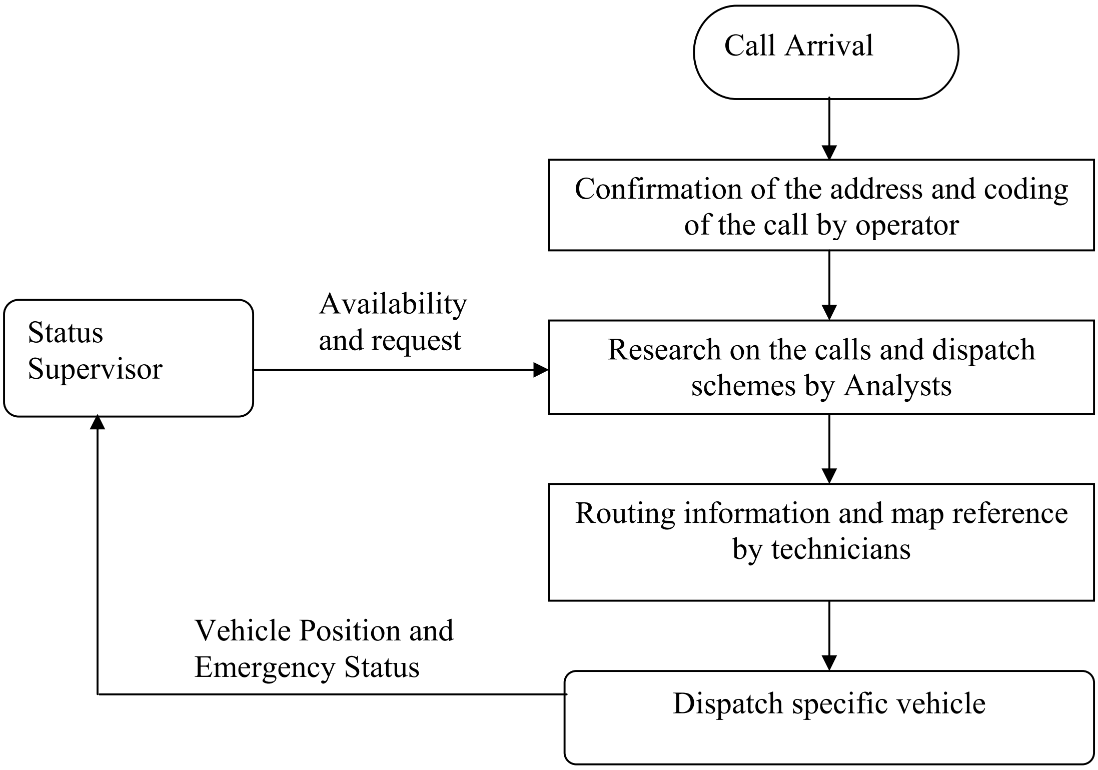
The emergency response vehicle fleet management system deals with the following problems:

1. Deployment Problem: when an emergency call arrives, dispatch the proper vehicles from one or more stations. When needed, relocate the available

emergency vehicles to potential relocation site to provide enough coverage for the entire region.

2. Routing Problem: provide route guidance for the dispatched vehicles.
3. Location Problem: select the best temporary/permanent location or locations for the vehicles in a set of potential locations.
4. Allocation Problem: select good vehicle/facility allocation scheme.
5. Evaluation: provide a sound comparison base to evaluate system performance under possible policy changes and to examine what-if scenarios.

The main focus of an emergency management center is to coordinate the actions to mitigate, prepare for, respond and recover from the effects of an emergency. The emergency response procedure includes various activities, such as alert and warning, damage assessment, emergency operation, evacuation, and fire and rescue. The routine in a real-world response operation is summarized in Figure 1-3. Based on this routine, we develop a system which is composed of four internal modules: travel time predictor, shortest path calculator, dispatching optimizer and simulator, and three external modules: evaluator and other optimizers. Figure1-4 shows the structure of the system.



Call Arrival
Confirmation of the address and coding of the call by operator
Availability and request
Status Supervisor
Research on the calls and dispatch schemes by Analysts
Routing information and map reference by technicians
Vehicle Position and Emergency Status
Dispatch specific vehicle

Figure 1-3: The Routine of Emergency Response

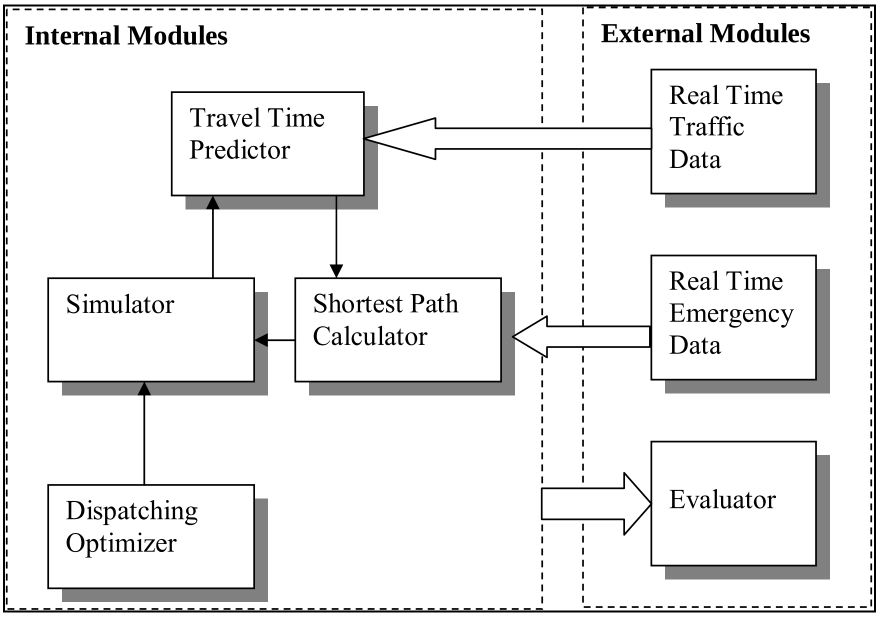


Figure 1-4: The Structure of an Emergency Response System.

The internal modules are the mathematical models that optimize system operations and assist the dispatch personnel in their daily emergency response operations. The external modules can be used by decision makers for planning purposes and system evaluation. Each module in this system can be a specific topic in the research of emergency response.

The benefits of the developed integrated emergency vehicle fleet management system include:

1. Reduce emergency response time: The loss of life and assets mainly depend on the planning phase and response operation. Proper planning of location and fleet can help speed up the emergency response. Fast and accurate response to an emergency can save precious time and improve the efficiency of the system.

2. Reduce operational cost: With better planning and operation guidance, the same crew or fleet will be able to handle the responsibility more efficiently and with higher performance levels.
3. Improve information utilization: The efficiency of the response system is heavily based on the information. Real-time traffic volume on streets has great influence on travel speed. The fleet surveillance system can track the status and location of vehicles and help decide the vehicle assignment plan for new emergency calls. A GIS maps can precisely locate the emergency site, and a GIS database can help establish the magnitude of life, property and effort involved, determining the risk zones based on land use data, building and activity in tune with the National Building Code guidelines.
4. Evaluate the efficiency and the effectiveness of services: An integrated fleet management can help record all of the necessary information needed for evaluation, so as to improve the performance in the future. Also the system will be able to record the causes and effects of the emergency so that more effective mitigation can be applied.

1.3 Research Scope and Tasks

In general, the scope of this study is to build up an Emergency Response Vehicle Fleet Management System, which can provide real-time fleet deployment and a routing function, and help the planning for emergency facility location and allocation. The research tasks include:

1. Identification: identify related issues and possible approaches to improve the efficiency of the system, and select proper methodology for the research.
2. Development:
    - Development of the deployment model: this model is used to dispatch vehicles to the emergency sites or other locations and provide routes for vehicles. This model accounts for real-time information, operational requirements, and other real-world constraints. It specifically addresses the diversion, route-change and area-coverage concerns.
    - Development of algorithms to solve the deployment model: in addition to the concern for solution quality, the computational time is very important in this research. The algorithms developed are able to provide quality solutions in a reasonable amount of time.
3. Simulation: Since it is impractical to test new models/strategies in real-world operations directly, simulation can provide a better pre-application testing platform for research. The simulation mimics the real-world emergency vehicle response. It generates emergencies and real traffic information, as well as keeps track of the location and status of each vehicle in the fleet and each emergency in the system. An optimization module is called iteratively to dispatch vehicles to appropriate destination. Various performance measures of the system are recorded and exported.
4. Applications: the proposed model is tested in a large-scale case study and is exercised in applications of planning and evaluation.

1.4 Research Approach

The research approach in this study is summarized below:

1. To solve the emergency vehicle deployment problem, a rolling horizon approach is selected. A mathematical optimization model is proposed and several efficient algorithms are developed and tested. These algorithms could replace a full enumeration search or fixed dispatching strategies, and also have some important applications in other transportation networks.
2. In order to evaluate the performance of the proposed system and different algorithms, a stochastic, microscopic Monte-Carlo simulation model is developed in this study. The developed simulation model is to:
    - Mimic the real operation requirements and emergency patterns.
    - Capture performance measures, such as the average response time and the worst response time routinely.
    - Provide a potential for further development of other special functions in the emergency response vehicle management system, such as demand variations, fleet variation and control alternatives.

   To test the developed system, we calibrated the simulation system with real operational data and compared the statistical results of real operations and simulation.
3. The developed mathematical model, algorithms and simulation model are applied in other relevant emergency response operations optimizations.

1.5 Organization of the Dissertation

The organization of this dissertation is as follows:

Chapter 1 introduces the background and the motivation for this research. The problem statement and research approach were presented in this chapter as well. In Chapter 2, we discuss the existing research relevant to emergency vehicle fleet management. Chapter 3 presents the analysis of the problem and the mathematical formulations. Chapter 4 discusses the heuristic methods proposed to initialize and improve the solution. Chapter 5 discusses various lower bound methods. It is shown why Lagrangian Relaxation cannot produce good lower bound and our lower bound method is based on a decomposition technique. The analyses of the solution quality are presented in this chapter as well. Chapter 6 discusses the simulation model which utilizes the mathematical model and heuristics and the experiments conducted. Chapter 7 presents the results of a comprehensive case study based on real-world data. The results of sensitivity analysis also are discussed in this chapter. Finally, Chapter 8 presents conclusions and directions for future research.

Chapter 2: Literature Review

In this chapter, we give a complete review of the emergency vehicle deployment problem and relevant topics. Section 2.1 provides a brief overview of the main issues related to emergency fleet management. Section 2.2 focuses on the existing research approaches on these topics. We review in detail General Assignment Problem (GAP) and Emergency Vehicle Location and Relocation Problem. The standard formulations and solution approaches are discussed and summarized. Section 2.3 summarizes the current meta-heuristics and Sections 2.4 and 2.5 summarize relevant research issues which includes simulation, travel time prediction, shortest path algorithms and GIS – aided Emergency Response Procedure. The research trends and conclusion are presented in the last section.

2.1 Introduction

The Emergency Response Vehicle Fleet Management System proposed in this research is aimed at improving the efficiency of emergency response by providing real-time emergency vehicle deployment. The system can also be used to optimize the temporary/permanent location of the emergency response facilities and the allocation of fleet to those facilities.

The core of the system is an optimization model that can dispatch and route vehicles on-line, while meeting various operational requirements, such as the response time limits and vehicle type requirements. Such a model requires a variety of information including the geographic information of the street network and building locations, the fleet configuration, operational policies and the fleet management strategies for emergency vehicles. The information mentioned above can be deemed as static information. Other real-time information includes location of vehicles, traffic volumes, real-time travel times, accident information and work zone information. Since no existing model has been developed before, we examine the relevant existing literature mainly focused on the assignment problem and location problem. Since the emergency response system should provide route information to dispatched vehicles, literature related to shortest path algorithms and travel time prediction methods are reviewed as well. To test and evaluate the performance of the proposed models and operation disciplines, a virtual environment is necessary to mimic the mechanism of real operations. Simulation modeling for similar systems becomes another important issue in this research. In Section 2.2, the existing modeling approaches will be introduced. Related topics of routing and dispatching will be discussed in Section 2.3. Section 2.4 summarizes the existing literature and research trends.

2.2 Modeling Approaches

Most of the literature in emergency response is focused on Emergency Medical Service (EMS) systems, and deals with the study of location, fleet size, and operational performance. These have been important subjects for operations researchers and management scientists. Similar research also includes many other public services such as emergency repair and traffic incident management.

2.2.1 Dispatching Problem

The key problem in an emergency response system is the Vehicle Dispatching Problem. When emergency calls arrive at the emergency response system, the most important responsibility of the dispatcher is to decide the number and types of required vehicles, and to dispatch these vehicles to emergency scenes. When real-time traffic information is available, it is desirable to provide route guidance as well to avoid congested areas in the transportation network. Intuitively, it is preferred to send the nearest available vehicles to the emergencies. Since the number of available vehicles is limited, when the number of emergencies that need attention grows, the system becomes heavily loaded and the response to some less severe emergency calls may have to be delayed to deal with the more severe ones. In this process, some emergency response vehicles that were dispatched earlier to respond to less severe emergencies may also be re-assigned to the new more severe ones and re-routed.

Limited literature exists that relates to this specific problem. Haghani et al. (2002) proposed a mathematical model that deals with the time-dependent EMS dispatching and re-routing. In their model, the vehicle dispatching problem is formulated as an integer model with an objective function that minimizes the total travel time in the system. A time-dependent shortest path algorithm is used in the calculation of travel times from vehicles to emergency locations. In their model, only one type of vehicle is considered and a simplifying assumption that each emergency call needs one and only one vehicle is made.

The relevant literature on the Generalized Assignment Problem (GAP) is quite extensive. The Generalized Assignment Problem deals with the question of how to assign *n* tasks to *m* machines in the best possible way. It consists of two components: the assignment as underlying combinatorial structure ("possible") and an objective function modeling the "best way".

The GAP is a well-known, NP-complete combinatorial optimization problem (Fisher, 1985). A typical GAP is the Knapsack Problem. Given *n* items and *m* knapsacks, with p_{ij} as the cost associated with assigning item j to knapsack i, w_{ij} as the weight of assigning item j to knapsack i, and c_i the capacity of knapsack i, assign each item j to exactly one knapsack i, not exceeding knapsack capacities.

$$\text{Min} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} \tag{2-1}$$

$$\text{Subject to } \sum_{j=1}^{n} w_{ij}x_{ij} \le c_i, i = 1,...,m, \tag{2-2}$$

$$\sum_{i=1}^{m} x_{ij} = 1, \text{j} = 1,...\text{n}, \tag{2-3}$$

$$x_{ij} \in \{0,1\}, i = 1,...,m, j \in N. \tag{2-4}$$

Early work on the generalized assignment problem concentrated on exact solutions to the problem using enumerative schemes with bounding methods (Martello et al., 1981, Ross, et al., 1975, Fisher, 1986). However, these types of methods usually are computational expensive. Since GAP is NP-complete, it is unlikely to find any efficient method for finding an exact solution. Later, more heuristics were developed. Catrysse, et al. (1992) surveyed the heuristics for the GAP. Quite a few of them are based on the linear relaxation of the General Assignment Problem (Brown, et al., 1985, Nulty, et al., 1988, Trick, 1992, Lorena, et al., 1996, Narciso, et al. 1999), and genetic algorithm heuristics (Lorena, 2002, Chu, 1997).

In our research, when assigning vehicles to emergency calls, we need to consider the number of vehicles needed and the vehicle types and destinations at the same time. At the same time vehicles need to be dispatched to the emergency sites with many other operational constraints. For example, vehicles should arrive at the emergency locations within a certain time limit; diversion of the destinations of vehicles should generate certain benefits to the system; and the area coverage requirements should be maintained. To some extent, the problem can be summarized as an expansion of the Axial Three Dimensional Assignment Problem (ATDAP). Multi-dimensional Assignment Problems are natural expansions of the linear assignment problem. They have been considered for the first time by Pierskalla (1967). The Axial Three Dimensional Assignment Problem is the most prominent representative of this class. The formulation of a standard ATDAP is as follows:

$$\text{Min} \quad \sum_i \sum_j \sum_k P_{ijk} X_{ijk} \tag{2-5}$$

$$\text{Subject to} \quad \sum_j \sum_k x_{ijk} = 1 \quad \forall\, i \tag{2-6}$$

$$\sum_i \sum_k x_{ijk} = 1 \quad \forall\, j \tag{2-7}$$

$$\sum_i \sum_j x_{ijk} = 1 \quad \forall\, k \tag{2-8}$$

$$x_{ijk} \in \{0,1\} \quad \forall\, i, j, k \tag{2-9}$$

Where the 0-1 decision variable x_{ijk} = 1 if, and only if, job *j* is assigned to worker *i* on machine *k*. p_{ijk} is the cost of assigning job *j* to worker *i* on machine *k*. Each constraint implies that each element of a set is assigned to exactly one element of each of the other two sets.

The ATDAP is an NP-hard problem (Karp, 1980). Pierskalla (1967, 1968) first proposed a heuristic for solving ATDAP, in which he used the classic branch and bound algorithm to solve the ATDAP to optimality. Balas and Saltzman (1989, 1991) introduced a branching strategy that exploits the structure of the problem and allows fixing several variables at each branching node. In the branch and bound procedure, strong lower bounds are essential. The most common approach used in ATDAP to get lower bounds is Lagrangian relaxation, by taking two blocks of the constraints of the ATDAP into the objective function. The relaxation is solved by a subgradient procedure. Burkard and Rudolf (1993, 1996) experimented with different branch and bound schemes for the ATDAP and reported satisfactory computational results.

Another approach is Hansen and Kaufman's (1973) primal-dual method, which is similar to the Hungarian method (Kuhn, 1955) for linear assignment problems.

In general, the ATDAP is a complicated problem and few heuristics have been developed for this problem. The existing solution methods are focused on finding the optimal solution of the problem while the computational time is not considered as a major restriction.

2.2.2 Emergency Vehicle Location Problem

A large portion of the existing literature is focused on the emergency facility siting problem and the most common approaches are to use mathematical programming and queuing methods. The allocation of emergency vehicles is an important part in this research. When taking the service coverage concern into consideration in the Emergency Vehicle Dispatch Problem, vehicle re-location is needed for better coverage of the service area and avoiding possibly extremely long travel times. The relevant literature review of this problem will be discussed in Section 2.2.3. Another possible application is for planning purpose in our proposed system, in which the emergency facilities need to be properly deployed to reduce the response time.

Hakimi was the earliest researcher who considered the location problems. First, Hakimi (1964) worked on determining the center of a network where the center is the point of the network from which the distance to the furthest points (worst case) is minimized. Later, Hakimi (1965) generalized the concept of a center and using

Boolean functions, he used an enumeration method to seek the minimum number of centers that covered all demand points within a specified maximum distance.

The process of formulating new models for new objectives, identifying new constraints, which take the models closer to reality, has attracted operations researchers over the years. Revelle (1989, 1997), Schilling et al (1993) and Marinov and Revelle (1995) provided a comprehensive review and perspective on these models.

The location/allocation models can be grouped into three categories:

- Basic deterministic covering models,
- Deterministic models which consider the value of additional service units, which can cover a node within pre-defined response time threshold, and
- Probabilistic models which allow randomness in service unit availability.

The basic deterministic covering models seek to position the least number of facilities needed to cover all points of demand within *S* distance or time units. Mathematically the problem is as follows:

$$\textit{Minimize} \quad z = \sum_{j \in J} x_j \tag{2-10}$$

$$\text{Subject to} \sum_{j \in N_i} x_j \geq 1 \quad \forall\, i \in I \tag{2-11}$$

$$x_j = 0,1 \quad \forall\, j \tag{2-12}$$

where:

J is the set of eligible emergency vehicle sites (indexed by j);

I is the set of demand nodes (indexed by i);

$x_j = 1$ if a facility is placed at node j and $x_j = 0$, otherwise; and

$N_i\{j \mid t_{ij} \leq S\}$ is the set of facility sites eligible to serve demand point i, where t_{ij} is the shortest time from the potential vehicle location j to demand node i; S is the time or distance standard for coverage. If a call for service is answered by available servers stationed inside this neighborhood, it will be answered within the time or distance standard.

Toregas et al. (1971) viewed the location of emergency facilities as a set of covering problems with uniform cost in the objective function. The sets are composed of the potential facility points within a specified time or distance of each demand point and linear programming is applied to solve the covering problem. A single-cut constraint is added as necessary to resolve fractional solutions. The model assumes an identical cost for all possible facility locations and a fixed upper limit of response time or distance. An equivalent problem is to minimize the total number of service facilities required to meet the response time or distance standards for each of the users. Rao (1974) provided two counter-examples. The first shows that a single cut may not always be sufficient and the second that the cut may not always result in an integer optimal solution. Toregas (1974) explained how the difficulty mentioned in Rao's paper could be overcome by an "in-and-out" algorithm.

Another derivative model is the *p*-center problem. The *p*-center problem seeks the locations of *p* facilities/stations which minimize the maximum distance separating any demand point and its nearest facility. A solution is obtained by solving a sequence of location set covering problems in which the maximum distance is successively reduced by the smallest unit of measurement of distance. The total number of facilities required may be *p* in number in each of a set of successive solutions as distance is decreased. When a value of *S* is reached at which the number of facilities increases to *p+1*, the preceding value of distance is the smallest maximum distance for which p facilities are feasible. Any smaller maximum distance requires a greater number of facilities. This method of solution was described by Minieka (1970) and Christofides and Viola (1971).

Church and ReVelle (1974) and White and Case (1974) framed a new problem that did not require the coverage of all nodes. This problem is called the Maximal Covering Location Problem or Partial Covering Problem. The standard formulation of the Maximal Covering Location Problem is as follows:

$$\textit{Maximize} \quad \mathrm{z} = \sum_{i \in I} w_i y_i \qquad (2\text{-}13)$$

$$\text{Subject to} \quad \sum_{j \in j} a_{ij} x_j \geq y_i \qquad \forall\, \mathrm{i} \in \mathrm{I} \qquad (2\text{-}14)$$

$$\sum_{j \in j} x_j = N \qquad \forall\, \mathrm{j} \in \mathrm{J} \qquad (2\text{-}15)$$

$$x_j = 0,1 \qquad \forall\, \mathrm{j} \in \mathrm{J} \qquad (2\text{-}16)$$

$$y_i = 0,1 \qquad \forall\, \mathrm{i} \in \mathrm{I} \qquad (2\text{-}17)$$

Where:

J is the set of eligible emergency vehicle sites (indexed by j);

I is the set of demand nodes (indexed by i);

$x_j = 1$ if a facility is placed at node j and $x_j = 0$, otherwise;

$y_j = 1$ if the node is covered by any facility and $y_i = 0$, otherwise;

$a_{ij} = 1$ if the facility at site j eligible to serve demand point i, $a_{ij} = 0$, otherwise; and

w_i is the weight of demand nodes (indexed by i).

The maximal covering idea was generalized for fire protection by Schilling et al. (1979). Their model accounts for coverage by two different types of service and both servers and the facilities were to be sited simultaneously. Batta and Mannur (1990) proposed a covering-location model for emergency situations that require multiple response units.

Some researchers recognized the possibility that in congested systems the first server might not be available at any call-in time and the importance of providing additional servers beyond the first within the coverage region of a demand area.
Two types of optimization models have been developed which address congestion: deterministic, or redundant coverage optimization models, and probabilistic optimization models.

Berlin et al. (1974) and Daskin and Stern (1983) structured models which utilized a number of facilities greater than or equal to the minimum number required by the location set covering problem. They focused on the total of redundant or additional coverage in the system beyond the first cover required by the location set covering problem. The mathematical formulation for the deterministic coverage optimization problem is:

$$\text{Max} \qquad Z = \sum_{i \in I} a_i r_i \tag{2-18}$$

$$\text{Subject to} \qquad r_i = \sum_{j \in N_i} x_j - 1 \quad \forall \quad i \in I \tag{2-19}$$

$$\sum_{j \in J} x_j = p \tag{2-20}$$

$$x_j \in \{0,1\} \qquad \forall \quad j \in J, i \in I, \tag{2-21}$$

Where:

r_i is an integer variable that measures the number of additional covering servers beyond one for demand node i;

p is the total number of stations in the neighborhood; and

a_i is a weight characterizing the importance of covering demand node i with redundant servers. All other notations are defined earlier.

In these models however, the weight a_i is constant, so that the locations with low population demands will receive the same coverage as those locations with very high population demands. Benedict (1983) and Eaton et al. (1986) restructured the objective of Berlin's and of Daskin and Stern's models by introducing the magnitude of demand. The new objective maximizes the product of population and the number of servers which provide additional cover to the population. However this modification makes additional coverers more likely for high demand nodes, but redundant coverers can pile up for some demand areas and not for others. Some demand nodes may be left with only a first coverer and no backup. Hogan and ReVelle (1986) modified the location set covering problem to maximize the population that has at least one redundant coverer. Later, they extended the model to the situation in which first coverage is not required, but is an objective as in the maximal covering location problem. In addition, second coverage is also a goal. The two objectives are to maximize the population achieving first coverage, and to maximize the population achieving second/higher coverage. The formulation for the extended model is:

$$\text{Max} \qquad Z_1 = \sum_{i \in I} a_i y_i \,, \; Z_2 = \sum_{i \in I} a_i r_i \tag{2-22}$$

$$\text{Subject to} \qquad r_i + y_i \le \sum_{j \in N_i} x_j \quad \forall \quad i \in I \tag{2-23}$$

$$r_i \le y_i \qquad \forall \quad i \in I \tag{2-24}$$

$$\sum_{j \in J} x_j = p \tag{2-25}$$

$$x_j, r_i, y_i \in \{0,1\} \;\; \forall \quad j \in J, i \in I, \tag{2-26}$$

Where:

$y_i = 1$ if demand node i is covered by the standard response (3 engines, 2 trucks), 0 otherwise;

$r_i = 1$ if node i is covered by more than one server, 0 otherwise. All other notations are as before.

In order to ensure that workloads are distributed evenly among the facilities and that quality of service is maintained, capacity constraints must be incorporated when designing the emergency service systems. Pirkul and Schilling (1988) formulated the siting models of emergency service system where facility workload is controlled and coverage for some or all demand points is provided. The objective is to minimize the sum of fixed and variable costs. An effective solution procedure was developed using a Lagrangian relaxation of the original model formulation. Haghani (1996) presented two formulations and two solution procedures for capacitated maximum covering location problem. One of the formulations is an extension of a maximum covering model with capacity constraints, and the second model accommodated the demands which are not covered by assigning them to located facilities which have excess capacity. A greedy adding heuristic and a Lagrangian Relaxation heuristic were proposed to solve these two problems with good quality solutions and computational times.

Although the deterministic additional coverage models are robust contributions to emergency service siting, it was a natural step for researchers to investigate probabilistic siting models later. In probabilistic siting models the formulation accounts for the randomness in the availability of the server. It is possible to develop models in which randomness exists in server availability, in time of travel, and in time of service.

The first probabilistic emergency model was constructed by Chapman and White (1974). They proposed a probabilistic location set covering model in which servers were not always available, and they named it maximum expected covering location problem. A simplified version of their model by Daskin (1983) used a single across-the-board estimate of the probability of the server being busy probability q. The advantage of this simplification is that the objective function can be easily formulated. The experiments show the results are very sensitive to q. A minor change in q may result in doubling the number of facilities. As a consequence, it becomes important to pursue different estimates of the region-specific busy fractions (ReVelle and Hogan, 1989).

$$\text{Max} \quad Z = \sum_{i \in I} a_i y_{i b_i} \tag{2-27}$$

$$\text{Subject to} \quad \sum_{k=1}^{b_i} y_{ik} \leq \sum_{j \in N_i} x_j \qquad \forall \quad i \in I \tag{2-28}$$

$$y_{ik} \leq \sum_{j \in N_i} x_j - 1 \qquad \forall \quad i \in I, k = 2,3,...b_i \tag{2-29}$$

$$\sum_{j \in J} x_j = p \tag{2-30}$$

$$x_j, y_{ik} \in \{0,1\} \qquad \forall \quad j \in J, i \in I, \tag{2-31}$$

where:

b_i is the smallest integer which satisfies $1-(\frac{\rho_i}{b_i})^{b_i} \geq \alpha$, ρ_i is the utilization ratio, so that the probability that at least one server is available within the standard response time to cover each demand node is forced to be no less than some reliability factor α ; and $y_{ik} = 1$ if k servers are potential coverers of demand node i , 0 otherwise. All other notations are the same as before.

All these models made the simplifying assumption that the probabilities of two vehicles being busy within the same region are independent. If this assumption is relaxed, the binomial distribution cannot be used for the two vehicles and queueing method needs to be applied to analyze the probability of each vehicle being busy.

Besides the linear models discussed above, Barker et al. (1989) developed an integer, nonlinear mathematical programming model to allocate emergency medical service ambulances to sectors within a county. The main constraint was to meet a mandated response-time criterion.

Another group of modeling methodology relies on queueing theory. A queueing system with exponential arrival time, exponential service time and c servers (M/M/c) is assumed in which servers are randomly selected without replacement until the first available server is found. Larson (1974) used a hypercube queuing model as a tool for facility location and redistricting in urban emergency services. The hypercube model, which is a nonlinear model, describes a spatially distributed queueing system with distinguishable servers. Each server may have two states: busy or free, and the state of the system of servers is given by a vector whose components are the individual states of each server. Thus, the system has 2^p states, where p is the total number of servers. The generalization to p servers leads to a state space which is formed by the vertices of a p-dimensional cube. With the strict assumptions that the call rates of each demand node are Poisson distributed, service times are exponentially distributed, and each call is served by one and only one server, closed-form expressions can be drawn from the steady state probabilities of each state. A computationally efficient algorithm for studying analytic behavior of a multi-server queuing system with distinguishable servers was developed by Larson (1975). Later, Chelst and Jarvis (1979) extended Larson's hypercube queueing model to enable it to calculate the probability distributions of travel time. Obviously, the assumption of only one server for each call cannot capture the characteristics of real-world operations. Chelst and Barlach (1981) described one exact and one approximate emergency service system model that can capture the simultaneous response of two identical units dispatched to a single call based on Larson's work. However, these models were still based on an M/M/c/0 system with Poisson arrival process, exponential service times, c servers and no extra waiting space, where blocked calls are lost, which may not be the case in real-world operations.

Another assumption in the available hypercube queuing model is that the service time is independent of the locations of calls and dispatched units. This assumption decreased the accuracy of the estimates for system's performance. Halpern (1977) performed a study in a simple two-server, two-customer system which showed a more accurate approximation for travel time is essential.

Berman and Larson (1985, 1987) studied the districting and location problem in the presence of queuing. Their studies involved single median, 2-median and p-median problems based on the M/G/C system, for which no closed-form expression exists for expected waiting time in the queue. An assumption of their queuing analysis was that a service unit on a congested network always returned to its home station. That is not true in the real world and not consistent with the objectives of this research.

The analytical techniques have some obvious shortcomings. First, it is difficult to build an appropriate sophisticated, analytical model for real-world systems. Furthermore, it may not be possible to solve the model using known analytical techniques.

2.2.3 Emergency Vehicle Relocation Problem

Instead of seeking for a solution to a static or probabilistic model of Emergency Facility Location Problem, a new approach is to dynamically relocate vehicles in real-time as vehicles are dispatched to calls. This is referred to as Emergency Vehicle Relocation Problem. An early dynamic model was proposed by Kolesar and Walker (1974) for the relocation of fire companies. Each relocation amounts to solving a static model subject to side constraints on vehicle moves. For example, one should avoid relocating too many vehicles at once or moving the same vehicle too often over a short period. Berman et al. (1982) considered optimal location-relocation decision for mobile servers. The randomness comes from the travel time and the objective is to minimize long-term expected average cost. A heuristic was developed for the problem on a general network with a single median. Berman et al. (1984) extended the study to a multi-facility problem. They developed a heuristic for this m-median problem and discussed simple bounds on the optimal objective function value. Carson et al.(1990) presented a case study of the relocation of a single ambulance. On the Amherst campus of SUNY Buffalo, the ambulance relocates with the moves of population throughout the day (from classroom buildings to dining halls to dormitories, etc.). Given the difficulties inherent in identifying probability distributions and estimating relocation costs in practice, a simplification was made to divide one day to four unequal time periods and a 1-median problem was solved in each period. Another dynamic, stochastic facility location problem was studied by Jornsten et al.(1994). The objective was to choose where and when to locate facilities over time in order to minimize the expected time-discounted cost with random production and distribution costs. Their algorithm used scenario aggregation and an augmented Lagrangian approach. Though this study is not on emergency response vehicles, the nature of the problem is same.

More recently Gendreau et al. (1999) developed a dynamic ambulance relocation model which can be applied in real-time through the use of tabu search algorithm and parallel-computing. Gendreau et al. (2003) proposed an a priori methodology for the dynamic relocation problem, in which several solutions are pre-computed in anticipation of future events and the appropriate solution. A similar model for physician cars is presented by Gendreau et al (2006). The assumptions made in the dispatch algorithm are similar to the work in Weintraub et al (1999) in that the closest unit is not always sent to a new call, but adapted for the pick-up and delivery nature of ambulance calls rather than the repair problem that is studied in Weintraub et al. (1999). The relocation algorithm is dynamic, that is, the problem is solved when there is a lack of ambulances somewhere in the area. Sathe et al. (2004) proposed a genetic algorithm in which they addressed the location-relocation decision for a fleet of response units in a transportation network, where travel conditions are uncertain. The problem was formulated with two objectives (maximize secondary coverage and minimize cost).One important issue in dynamic dispatching is the computational time. A new solution will be needed within a short time period when a call arrives or when the traffic information is updated. This can be time consuming or even infeasible when calls arrive with high frequency throughout the day. Marianov et al. (1995), Brotcorne et al.(2003) and Goldberg (2004) provided comprehensive surveys of the emergency vehicle relocation problem.

Police patrol services can be grouped into the emergency vehicle relocation problem as well. There are two major types of problems in literature. One is to determine police patrol areas; the other one is to determine the routes of police patrol service. The solution approaches used in the first type of problems are similar to the location and relocation problems discussed above, while the nature of the second type of problems is the arc routing problem. Due to the fact that police patrol service operational information is highly sensitive, the second type of problems is not considered in this dissertation.

2.3 A Brief Review of Heuristics

Reviewing the solution approaches for problems discussed above indicates that heuristics are developed when the problem size is large. In the past 20 years, meta-heuristics have become more and more popular in solving real-world large size optimization problems.

The term “meta-heuristics” was first proposed by Glover in 1986. Meta-heuristics contains all heuristics methods that show evidence of providing good quality solutions for the problem of interest within acceptable computational time. Generally, meta-heuristic approaches are capable of a wide variety of applications, rather than particularly tailored to specific problems. Meanwhile, meta-heuristics do not guarantee optimal global solutions.

Meta-heuristics can be classified into two groups: point-to-point methods and population-based methods. In the point-to-point methods, the search invokes only one solution at the end of each iteration from which the search will start in the next iteration. On the other hand, the population-based methods invoke a set of many solutions at the end of each iteration. Genetic algorithms are an example of population-based methods, and basic simulated annealing and tabu search are examples of point-to-point methods.

In this section, we highlight three meta-heuristics: simulated annealing, genetic algorithm and tabu search. We address the ideas and concepts behind these meta-heuristics rather than the coding details and step-by-step procedures, because the concepts and ideas are the essential elements that can be adapted to develop new algorithms.

2.3.1 Simulated Annealing

Simulated annealing is a popular technique from the early 1980's. It is motivated by the thermodynamic process of annealing in physics. The first algorithm of Simulated Annealing was proposed by Metropolis et al. (1953). They suggested this algorithm to simulate the equilibrium of a collection of atoms at a given temperature. This pioneering technique inspired Kirkpatrick, et al. (1983) to use it in optimization and call it Simulated Annealing (SA). The simulation was used to search the feasible solutions of an optimization problem, with the objective of converging to an optimal solution. Since then a number of studies that have used SA have emerged in the area of optimization and theoretical aspects as well as the applications of SA have been extensively studied.

In a typical SA algorithm, trial points are successively generated in a neighborhood of the current solution. Whether or not the current solution should be replaced by the trial point is determined based on a probability. Specifically, if a move from the current solution *x* to another trial point results in an inferior solution *x'*, and the difference between these two solutions is Δc, then the move to *x'* is accepted if

$exp\ (-\Delta c\ /T) < R$. where T is a control temperature parameter, and R is a uniform random number between (0, 1). Convergence to an optimal solution can theoretically be guaranteed after an infinite number of iterations controlled by a procedure called cooling schedule. The main control parameter in the cooling schedule is the temperature parameter T. The main role of T is to let the probability of accepting a new move be close to 1 in the earlier stages of the search and to let it be almost zero in the final stages of the search. The selection of the values of T depends on the nature of the problem, and a proper cooling schedule can speed up the finite-time implementation of SA to simulate the asymptotic convergence behavior of the SA.

Simulated Annealing is a point-to-point based searching method and is a simple procedure to apply. One of the most powerful features of SA is its ability of avoiding being trapped in local minima by accepting up-hill moves through a probabilistic procedure especially in the earlier stages of the search. On the other hand, the main drawbacks that have been noticed in SA are its suffering from slow convergence and its wandering around the optimal solution if high accuracy is needed. The efficiency and effectiveness of the algorithm depends on the selection of start solution neighborhood, starting temperature and cooling schedule. Further improvements can be made by detailed analysis of the problem characteristics or by combining SA with other optimization/analysis techniques.

2.3.2 Genetic Algorithm

A genetic algorithm (GA) is a population-based search methodology that tries to mimic the genetic evolution of a species. GA simulates the biological processes that allow the consecutive generations in a population to adapt to their environment. The adaptation process is mainly applied through genetic inheritance from parents to children and through survival of the fittest. Some pioneering works traced back to the1960s preceded Holland's (1975) main presentation of the GA. However, GAs had limited application until their multipurpose presentation of Goldberg (1989) in search, optimization, design and machine learning areas. Nowadays, GA is considered to be the most widely known meta-heuristic.

GA starts with an initial population whose elements are called chromosomes. The chromosomes consist of a series of variables which are called genes. A fitness function is used to evaluates and rank chromosomes in a population. This is a designed function that measures the goodness of a solution. To construct the complete structure of the GA procedure, there is a selection process and twp operators perform essential roles: *crossover* and *mutation* operators.

The selection process deals with selecting an intermediate population from the current population. The crossover and mutation operators are then applied to this population. In selection process, chromosomes with higher fitness function values have a greater chance to be chosen than those with lower fitness function values. Crossover operator aims to interchange the information and genes between chromosomes. Therefore, crossover operator combines two or more parents to reproduce new children, then, one of these children may hopefully collect all good features that exist in his parents. Crossover operator plays a major role in GA, so defining a proper crossover operator is very important in order to achieve a better performance of GA. Mutation Operator is applied to increase the variability of structure by altering one or more genes of a probabilistically chosen chromosome. Finally, another type of selection mechanism is applied to copy the survived members from the current generation to the next one. These GA operators have been extensively studied. Many effective settings of these operators have been proposed to fit a wide variety of problems in recent 20 years. All of these efforts have made genetic algorithms a mature approach to be used in a variety of fields.

Another important issue in GA is the form to express the chromosomes and genes, namely, coding. There are two major types of coding: binary coding and real coding. In binary coding, the chromosome is expressed as a binary string; while in real coding, the chromosome is express as the real value of the variables. Therefore, in binary coding, the search space needs to be mapped into a space of binary strings, and after reproducing an offspring, a decoder mapping is applied to bring them back to their original form in order to compute the fitness function values. Many researchers believe that the binary coding is ideal, while the real coding is more applicable and easy in programming.

One important advantage of Genetic Algorithms is that they are often fairly robust. With properly tuned crossover point, mutation rate, and an incremental replacement policy, the algorithms will often give good results. However, usually they cannot provide ideal solutions for Location Problem, especially when the computational time is limited.

2.3.1 Tabu Search

Tabu Search (TS) is a heuristic method first proposed by Glover (1986). TS has been proposed and developed for many combinatorial optimization problems, such as the Traveling Salesman Problem (TSP), Vehicle Routing Problem (VRP), GAP and Facility Location Problem. However, there are limited number of TS contributions in continuous optimization problems.

The main feature of TS is its use of an adaptive memory and responsive exploration. Simple TS combines a local search procedure with anti-cycling memory-based rules to prevent the search from getting trapped in local minima. Specifically, TS restricts returning to recently visited solutions by constructing a list of them called Tabu List (TL). In each iteration of a simple TS algorithm, trial solutions are generated in a neighborhood of the current solution. The trial solutions generation process is composed to avoid generating any trial solution that is already recently visited. The best trial solution found among the generated solutions will become the next solution. Therefore, TS can accept uphill movements to avoid getting trapped in local minima. TS can be terminated if the number of iterations without any improvement exceeds a predetermined maximum number. A simple TS structure described above is called short-term memory TS. Tabu tenure and aspiration criteria are used to update the memory-based TL. Tabu tenure is the number of iterations in which a tabu move is considered to remain tabu or forbidden, which means after one tabu tenure, the solution in tabu list can be removed from the listed. An improving solution can be accepted even if generated by a tabu move when the aspiration criteria are met.

The short-term memory is built to keep the recent information. In order to achieve better performance, long-term memory has been proposed in advanced TS that records attributes of special characters like elite and frequently visited solutions.

Intensification and diversification are often used to adapt the search process of TS. Intensification is to give priority to elite solutions in order to increase the possibility of obtaining better solutions in their vicinity, and diversification is to discourage attributes of frequently visited solutions in new move selection functions in order to diversify the search to other areas of solution space.

Similar to SA and GA, TS can provide good solutions even when we do not know much about the problem to be solved. The most obvious advantage of Tabu Search is its adaptive memory, while the other two methods are memoryless. These features have made TS a more and more widely accepted meta-heuristics for discretized combinatorial problems.

Glover (1997) summarizes the characteristics of these three heuristics as shown in Table 2-1.

Table 2-1: Comparison of the Three Modern Heuristics

Meta - heuristic	**Basic**	**Improved**
Simulated Annealing	M/S/1	M/N/P
Genetic Algorithm	M/S/P	M/N/P
Tabu search	A/N/1	A/N/P

Notation:

A: Adaptive Memory
N: Systematic Neighborhood Search
P: population Size P

M: Memoryless
S: Random Sampling
1: Population Size 1

Though these meta-heuristics can be used in most combinatorial problems, their ability in solving continuous problems is limited. Usually continuous problems need to be discretized in order to apply these meta-heuristics. Generally, a meta-heuristic can provide “good” solution without exploring the nature of the problem, but better understanding of the characteristics of the problem of interest can help in selecting proper method and improving the structure of these heuristics. The analysis of problem structure can also help in developing hybrids of different meta-heuristics or meta-heuristics with traditional operations search methods, which can be more efficient algorithms.

2.4 Simulation

Compared to mathematical modeling and queueing methods, simulation models enable us not only to find a good solution to some decision problems, but also to observe a system under different sets of assumptions. They also provide the possibility to test new operational strategies such as different ambulance locations or dispatching rules. In the past 30 years, simulation models have been developed and are commonly used to evaluate the performance of emergency response systems.

The model proposed by Savas (1969) is the earliest simulation model for the purpose of evaluating an EMS system. He built a simulation model to analyze the possible improvements in ambulance service that would result from proposed changes in the number of ambulances and location of New York City's ambulance system. The cost-effectiveness of several alternatives was examined. Carter et al. (1970) built simulation models for fire department operations, in which two urban emergency service units cooperate in responding to alarms or calls from the public in a specified region of a city. The model can specify which unit should respond to each call by defining a response area for each unit. The average response time to alarms and the workload of each unit are calculated as functions of the boundary that separates their response areas.

Fitsimmons (1973) introduced "CALL" (Computerized Ambulance Location Logic) program for the City of Los Angeles. This program deploys ambulances so as to minimize the mean response time. The final deployment substantially reduced the probability of excessive response times and smoothed the workload among the ambulance crews as well. These early simulation models are based on the First-In-First-Out system. No queue is considered in the simulation and these simplifications greatly reduced the realism of the models.

Ignall et al. (1978) used simulation to suggest approximate analytical models for use in police patrol and fire operations in New York City. The link between simulation and analytical models has been further analyzed and evaluated in a paper by Shantikumar and Sargent (1983), in which they suggested the use of a hybrid approach that embedded analytical models in a simulation procedure.

Lubicz et al. (1987) set up an event-driven EMS simulation model for rural areas in Poland. The main events in the model are the arrival of a new call, the end of service on the scene, and the arrival of patients in hospitals. This simulation model made a contribution in that it classified emergency calls into several priorities. Goldberg (1990, 1991) conducted a series of studies on EMS systems. In order to evaluate the emergency vehicle base locations for Tucson, AZ, Goldberg et al. (1990) together with the Tucson Fire Department developed a simulation model with detailed discussion of model development, data collection, model validation and experimentation. Their simulation model is a multi-server queuing system. The simulation model serves calls on a FIFO basis since the model does not consider priority scheduling of calls. The major difficulties in applying the model were developing the travel time model and validation. Goldberg (1991) later formulated an optimization model that extends the previous work by allowing for stochastic travel times, unequal vehicle utilizations, various call types, and service times that depend on call location. Goldberg and Szidarovszky (1991) presented two iterative methods for solving a model to evaluate probabilities of vehicles being busy for emergency medical service vehicles. The model considers location dependent service times and is an alternative to the mean service calibration method; a procedure used with the hypercube model, to accommodate travel times and location-dependent service.

Zografos (1993) developed a simulation model for evaluating the performance of emergency response fleet for an electric utility company, where response time is selected as the performance measure. Later, Zografos et al. (1995) developed another simulation model that focused on the operation of freeway emergency response units. In this research, average incident duration, dispatch and travel time and two dispatching policies (FIFO and Nearest Origin) are studied.

Simulation analysis is a necessary tool in the study of emergency vehicle fleet management. It is impossible to apply a new policy into real operation without validation in simulation and tests. In all these models the emergency calls were treated with the same priority, and no coordination of the different types of vehicles was considered. A simulation system which can represent real situations and accommodate various dispatching strategies will greatly benefit this study.

2.5 Related Issues

Because of the real-time feature of the proposed emergency response vehicle fleet management system, shortest path travel time plays an essential role as the base criterion for on-line vehicle dispatch and routing. Since the travel time on the links is time-dependent, to select the shortest travel route for each possible origin-destination pair, we need to deal with two issues: (1) what kind of shortest path algorithm should be used in the model, and (2) how to make short term travel time prediction in an urban/suburban street network.

2.5.1 Shortest Path Algorithms

The Shortest-path Problem is one of the most fundamental network optimization problems. It is commonly encountered in the study of transportation and communication networks. Numerous papers, reports and dissertations have been published on the subject.

Deo and Pang (1984) proposed a classification scheme to characterize algorithms for solving shortest path problems. The algorithms are classified according to:

- The problem type: e.g., usual path length and generalized path length, constrained and unconstrained paths, number of sources.
- The input type: e.g. local properties and global properties, directed or undirected network, probabilistic or deterministic link lengths, with or without negative links.
- The type of underlying technique employed to solve the problem: e.g., path finding or distance finding, different updating techniques, combinatorial or algebraic technique.

Gallo and Pallotino (1988) provided another survey of shortest path methods. The algorithms are derived from one single prototype method. The difference relies only on the particular data structure used in the implementation. Cherkassky et al. (1993) conducted an extensive computational study of shortest paths algorithms, which includes established methods, recently proposed algorithms and new algorithms. When travel time is relatively stable, the travel time by standard shortest path algorithms may provide a quality solution. Hall (1986) proved that standard shortest path algorithms (such as Dijkstra's algorithm) are not applicable to problems with fluctuating traffic speed. To take advantage of real-time traffic information, it is necessary to use a more sophisticated shortest path algorithm such as a dynamic or a stochastic shortest path algorithm.

Dynamic Shortest Path Algorithm

Dynamic shortest path problems can be categorized based on the following characteristics:

- Time horizon: based on how time is treated, the dynamic shortest path problem can be divided in two types: discrete and continuous. In discrete dynamic networks, time index is modeled as a set of integers. In continuous dynamic networks, time is treated as real-valued numbers.
- Objectives: in most cases, we are interested in the "fastest" path in a dynamic network and the travel time is the link cost. However, the cost of links can be more general, e.g., time-dependent marginal travel times encountered in system optimum dynamic traffic assignment models.
- First In First Out (FIFO): A FIFO network is one in which no one can depart later at the beginning of one or more links and arrives earlier at the end. When

the link travel times satisfy the FIFO condition, the static shortest path algorithms can be used without extra cost to solve the one-to-all fastest paths problems in dynamic networks. However, in transportation applications, FIFO may not be satisfied all the time.

- Waiting time: the discrete dynamic shortest path algorithm can be viewed as a static network obtained by using a time-space expansion representation. Obviously, when waiting time is allowed, the size of time-space expansion will be much larger than the case without waiting time. The waiting-is-allowed policy is conceptually a particular case of the waiting-is-not-allowed policy.

The earliest literature on dynamic shortest path algorithms appears to be by Cooke and Halsey (1966). Their algorithm is based on Bellman's principle of optimality. It discretizes the time horizon of interest into small intervals. Starting from the destination node, and calculates the path operating backwards. This problem can be seen as the first deterministic time-dependent shortest path algorithm, where the link-delay functions are deterministically dependent on arrival times at the tail node of the links. Ziliaskoulos et al. (1993) introduced an all-to-one time-dependent shortest path algorithm given a time horizon in a network with time-dependent link costs. The algorithm was coded, and tested on data from real streets and random networks. Chabini (1998) studied the algorithms for discrete dynamic shortest path problems, which included the all-to-one dynamic shortest path problems and the one-to-all fastest path problems.

Stochastic Shortest Path Algorithm

When real-time link traffic information is complete, precise and available all the time, the dynamic shortest path algorithm can provide precise travel time estimation for vehicle dispatching and routing. However, in reality, the traffic information for local streets is not available usually. Even for arterials the real-time traffic information will be updated over certain time intervals. An alternative for this situation is to use historical data to estimate the travel time on local links and use prediction techniques to generate the link travel times during the time interval.

When link travel times are probability functions of the environment variables, such as departure time, traffic condition and so on, the problem of determining the stochastic paths is a stochastic shortest path problem. In contrast to a deterministic model, which yields a single shortest path, the stochastic model yields a set of paths, each having a different probability to be the shortest path. The first paper on this topic is by Frank (1969), in which he derives a closed-form for solution for probability distribution function of the minimum path travel time in a stationary network, given discrete/continuous joint probability distribution function of the link travel times. When the link costs are either all non-negative or all non-positive, it is possible to transform this problem into a Markov chain with a single absorbing state and a binary cost for each transition. The distribution of the length of the shortest path is given by the distribution of the total cost incurred until the absorption. Kulkarni (1986) first introduced the method with the assumption that the link lengths are exponentially distributed. Corea and Kulkarni (1993) extended the link length to be independent, non-negative integer variables. Bertsekas and Tsitsiklis (1991) extended the research by removing the usual restriction that costs are either all non-negative or all non-positive.

A branch-and-bound method was proposed to find the least expected travel time path on this type of network (Hall, 1986). Sivanandan and Hobeika (1987) developed a heuristic method to find the stochastic shortest-path, which substantially reduces the computer storage and execution time. Psaraftis and Tsitsiklis (1993) considered the dynamic shortest paths in acyclic networks with Markovian link costs. They developed a dynamic programming procedure to solve the corresponding problem. The complexity of this method was shown to be O (n^2k+nk^3), where *n* is the number of network nodes and *k* is the number of Markov states at each node. Hadas and Ceder (1996) developed an algorithm to find a stochastic shortest path, based on the *k* shortest-path method and a simulation model. In this study, three distribution functions were considered: constant, exponential, and uniform. Miller-Hooks (1996, 1998) presented various algorithms to generate optimal or Pareto-optimal paths over a time period in a stochastic, time-varying network, and these algorithms were tested on some randomly generated networks. Barto et al. (1995, 1998) focused their work on developing off-line algorithms for solving Stochastic Shortest Path Problem by modifying the real-time dynamic programming algorithm. The solutions provided by these algorithms are more reliable than deterministic Shortest Path Algorithms, even without precise real-time traffic information. But it is not easy to determine the distribution function of the travel time on link in a real street network, and the computational time is a major disadvantage for most of the algorithms.

2.5.2 Short Term Travel Time Prediction

Since we are more interested in the travel time that the drivers *will* encounter, the precision of travel time prediction results determine the reliability of dispatching and routing schemes. Besides Historical Data-based Algorithm (Stephanedes, et al., 1981; Hoffman and Janko, 1988; and Kaysi, 1993), Time-Series Analysis Technique is the most discussed travel time prediction method (Eldor, 1977; Gafarian et al., 1977; Nicholson and Swann, 1974; Nahi, 1973; Chang and Gazis, 1975; and Cragg and Demetsky, 1995). Many Simulation Models (METANET, SIMRES (Simulation of the Regulation of a Reservoir), STM (Statistical Traffic Model) and DYNASMART) have been developed for travel time prediction. Unfortunately, they cannot support the online application in short term travel time prediction. Recently, prediction models based on Artificial Neural Networks are becoming widely used in short term prediction as well (e.g. Smith and Demetsky, 1995; Chang, 1999; and Huisken, 2003).

2.6 Trend of Research and Application

From the early 1970's, researchers noticed that better allocation of emergency facilities can help reduce the response times and improve the service levels. Chaiken and Larson (1972) provided a survey of methods for allocating urban emergency units, which discussed the four aspects of allocation policy: (1) determining the number of units to have on duty, (2) locating the units and facilities, (3) designing their response areas or patrol areas and (4) planning preventive-patrol patterns for police cars. In1978, Chaiken summarized the implementation process of six field-tested deployment models for emergency service agencies. Over half of those who acquired these models actually used them and nearly all users made operational changes based on the output. Such kind of research is valuable for the researchers in this area. Regretfully, in recent years, no similar work is available.

From the late 1980's, computers were widely used in emergency response systems. However, computers were used for recording emergency call information and for system evaluation purpose, and did not help with the vehicle dispatching and routing. At the same time, more complicated analytical models and simulation models were developed for research purposes due to the increasing processing ability of computers.

In the late 1990's, with the development of wireless communication technology, more on-line information became available. Quite a few research studies focused on the GIS software application in emergency response systems to provide shortest route and help with the risk assessment and mitigation. The research on vehicle dispatching problem focused more on meeting the detailed operation requirements such as balancing workload and real-time vehicle dispatching.

Recently, the interoperability and exchange of data across all public safety and transportation agencies is becoming common practice. This provides an extraordinary opportunity to improve coordination of activities of these agencies that play key roles in emergency response service. Cooperation and coordination among these agencies for improving emergency response service has never been explored in previous research. The current paradigm for interoperability and data and information exchange among agencies has created a tremendous opportunity for a major research contribution by developing a more integrated emergency response system.

Chapter 3: Problem Statement and Mathematical Model

Traditionally, Emergency Vehicle Deployment is decided by a group of experienced emergency analysts. Based on the code of each emergency call, the required vehicles in each type are decided and the vehicles are dispatched according to certain dispatching strategies. In the literature, very few practical mathematical formulations for emergency vehicle deployment are suggested. Many field operational constraints make the problem difficult to formulate. In this study, we formulate the Emergency Vehicle Deployment Problem as a General Assignment Problem with several extensions that are required for field operations.

In this chapter, different formulations to achieve the objectives in a real emergency vehicle fleet deployment are presented. In the first section, we analyze the service objectives, problem nature, important components in the problem and assumptions. In the following sections, several formulations for these service objectives and operational requirements are proposed.

3.1 Problem Statement

As the general operation procedure, when emergency calls arrive at the emergency response control center, the most important responsibility of the dispatcher is to decide the number and types of required vehicles, and to dispatch these vehicles to the emergency scene. When real-time traffic information is available, it is desirable to provide route guidance as well to avoid congested areas in the transportation network. Intuitively, it is preferred to send the nearest available vehicles to the emergencies. Since the number of available vehicles is limited, when the number of incidents that need attention grows, the system becomes heavily loaded and the response to some less severe emergency calls may have to be delayed to deal with the more severe ones. In this process, some emergency response vehicles that were dispatched earlier to respond to less severe incidents may also be re-assigned to the new more severe incidents and re-routed.

Figure 3-1 shows a simple example of response area. The response area has four zones. Each zone has an emergency station with one vehicle. The coverage area of each station is shaded. Here, we define the coverage area as the area that can be reached by the vehicle in a station within a certain travel time/distance. It is understandable that there might be part of the response area without proper coverage, and certain nodes, such as node *n*, may be far away from any station.

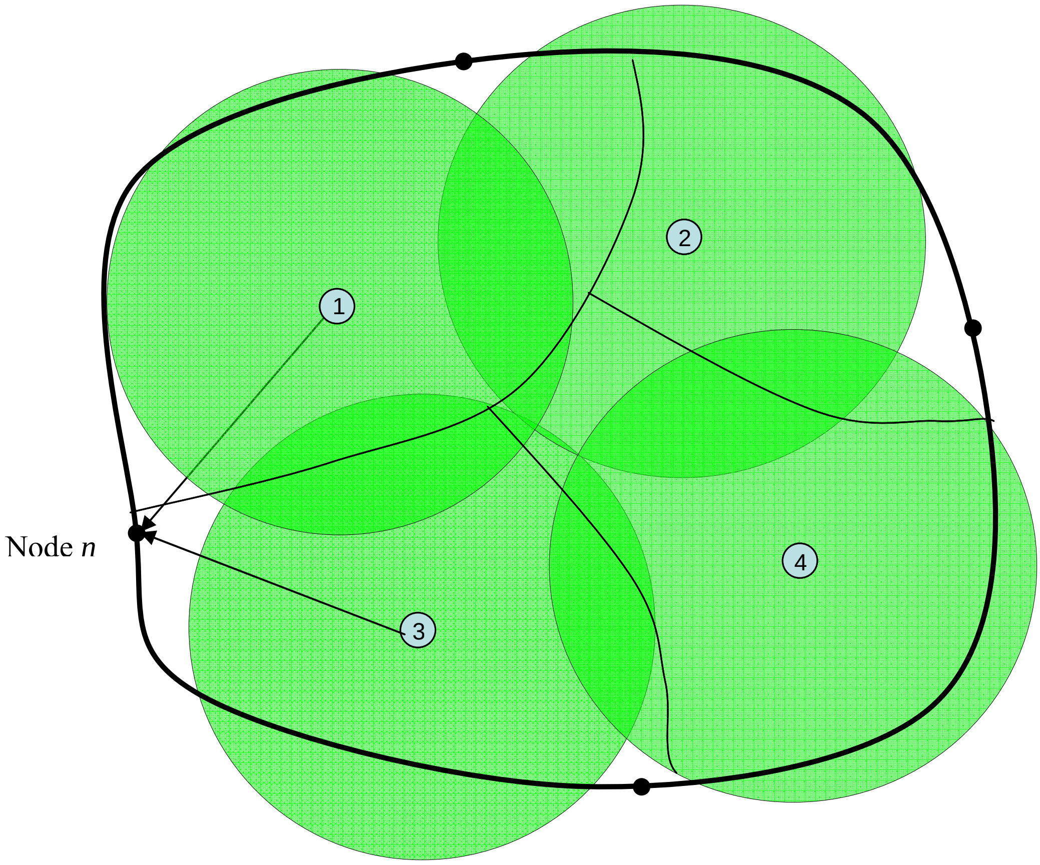


Figure 3-1: Sample Response Area

As shown in Figure 3-2, at time *t*, an emergency occurs at node *j*. The vehicle in station 2 is assigned to deal with it. At time *t*+1, another emergency at node *k* occurs. Since the vehicle in station 2 has already been assigned, the closest vehicle to the new emergency is the vehicle in station 1. If we re-assign and re-route the vehicle from station 2 to the new emergency and assign the vehicle from station 1 to the earlier one, we are able to avoid the long travel time from station 1 to node *k*. This makes the response time for both emergencies shorter. When real-time traffic information is available and congestion on pre-selected routes is detected, vehicles responding to both incidents can be re-routed to avoid the possible delays.

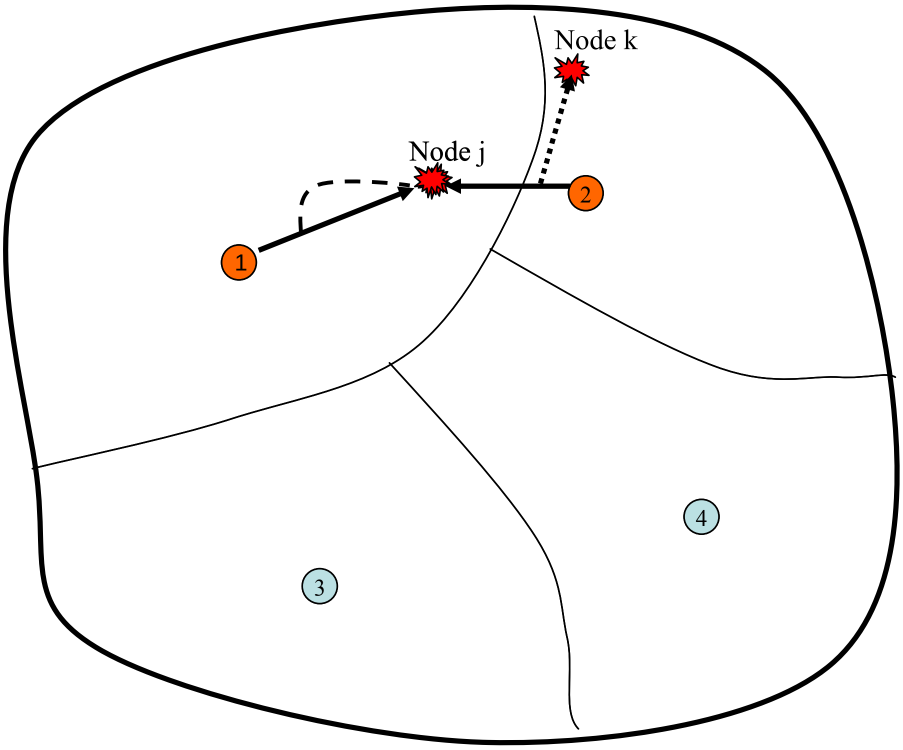


Figure 3-2: A Simple Example of Dispatching and Routing Problem

The example shows that the operations can be improved by utilizing on-line information. It will be very helpful if we can develop an online model that can handle the real-world operational requirements as well as optimizing the assignment scheme to reduce response times.

Another important issue that needs to be considered in emergency vehicle deployment is the service coverage. When emergency response vehicles arrive at emergency sites and are busy responding to incidents, gaps in the service area will be created which cannot be effectively covered. This means new emergency calls from within these areas may experience long delays in response. Figure 3-3 shows an example of this problem. In Figure 3-3 the two circles show the contours around the stations that can be reached within *i* minutes. If vehicles from stations 1 and 2 are assigned to emergency calls at nodes *j* and *k*, the blank areas in zones 1 and 2 are without coverage (within *i* minutes), and some points such as node *m* may experience very long travel times.

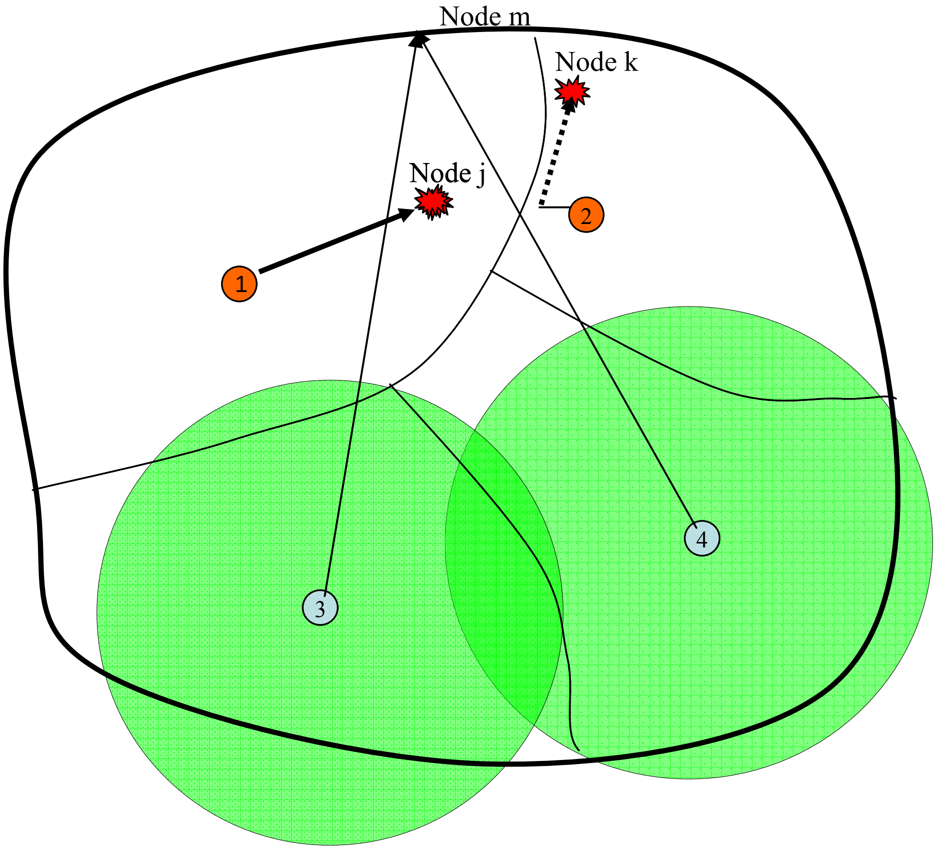


Figure 3-3: A Simple Example of Coverage Problem

To avoid this situation, we can relocate the vehicle at station 3 to station 1. As shown in Figure 3-4, both the size of the uncovered area and the longest travel time will decrease as a result. This shows that proper vehicle re-location and re-distribution may improve the service provided to general population. The two important aspects that need to be considered are the total area covered within the critical time (pre-set coverage time) and the longest travel time to an incident. These two aspects represent the emergency response system's average and worst-case performances.

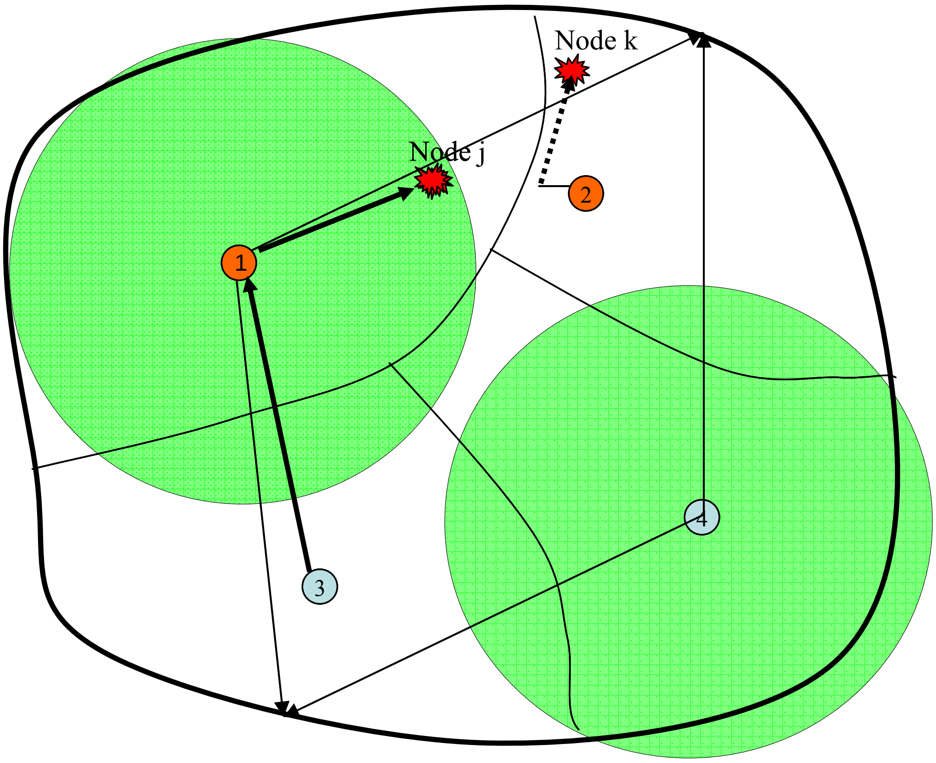


Figure 3-4: A Simple Example of Coverage Problem

In real-world operations, relocation of vehicles to provide better area coverage is considered occasionally. These decisions are made independent of the emergency response vehicle dispatching decisions.

Another issue is the crew scheduling. Since the crew scheme for each type of emergency vehicle are usually fixed as if we have proper crew on each vehicle, the diversion, rerouting or relocation should not affect the crew scheme. Therefore, we do not consider the crew scheduling in our study.

3.2 Dynamic Characteristic

As mentioned in the last section, the core of the emergency vehicle fleet management system is to provide efficient dispatching scheme which can better utilize the fleet. The dispatching scheme made for current incoming emergency call will influence the dispatching scheme of future calls.

In a simplified system with ambulances only that has a fixed First-Come-First-Service dispatching strategy, no diversion, re-routing, nor relocation, and in which a single vehicle is assigned to each request, the simplified dispatching problem can be formulated as follows:

When a call from location i comes in at time t, and j is the dispatched vehicle, then

$$w_i(t, a_j, l_j) = Max(0, a_j - t) + C_{ji} \qquad (3\text{-}1)$$

where w_i is the response time of call i, a_j is the available time of vehicle j, C_{ji} is the travel time from vehicle j 's location l_j to the location of call i.

$$j^* = Arg\min(w_i(t, a_{j^*}, l_{j*})) \tag{3-2}$$

In this case, j^* will be the closest available vehicle to emergency call *I*, which is the result of the Nearest Origin dispatching strategy. If the vehicle will send the patient of call i to hospital *h* after on-site treatment and stay there,

$$a_{j^*} = t + w_i(t, a_{j*}, l_{j*}) + C_{ji} + m_i + C_{hj} \tag{3-3}$$

where m_i is the on-site treatment time and C_{jh} is the travel time from emergency site to hospital. At the end of this procedure, the ambulance will stay at the hospital (*h*).

$$l_{j*} = h \tag{3-4}$$

If we only consider the current step, a greedy algorithm can be developed which provides a reasonably efficient dispatching scheme. However, when considering the future demands, the format of the problem becomes much more complicated.

Suppose the dispatcher knows the probabilistic request profile (i.e., the probability that the next call will occur at a particular node) and the average time between requests. The expected time of the next request is $t+\delta$, where δ is the average headway between requests. The probability that this request at location *i* will be serviced by vehicle *j* is $p_{ij}(t+\delta)$. If the dispatcher looks ahead to the next *N* assignments and seeks to minimize the total expected response time, the simplified problem can be formulated as a dynamic programming problem with a finite horizon. In the terminology of dynamic programming, let the dispatching policy be $\pi = \{j_0, \ldots,$

j_n, …, j_N}. In the *N*-stage problem, the expected cost of a policy π, given the initial location i_0, is

$$\phi_\pi(i_0) = w(t, a(j_0), l(j_{0)}, i_{0)} + \sum_{n=1}^{N} \varpi(t + n\delta, a(j_n), l(j_n)) \tag{3-5}$$

Where cost is measured in terms of expected response time, and

$$\varpi(t + ns, a(j_n), l(j_n)) = \sum_{n=1}^{N} \sum_{ij} (Max(0, a(j_n) - t - n\delta) + c(l(j_n), i) p_{ij}(t + n\delta) \tag{3-6}$$

If $j_n \in \pi$ then

$$a(j_{n+1}) = t + n\delta + w(t, a(j_{n),} l(j_n), i) + C_{ji} + m_i + C_{ih} \tag{3-7}$$

and

$$l(j_{n+1}) = h \tag{3-8}$$

otherwise

$$a(j_{n+1}) = a(j_n) \tag{3-9}$$

and

$$l(j_{n+1}) = l(j_n) \tag{3-10}$$

The optimal policy is

$$\phi_{\pi^*}(i) = Min(J_\pi(i)) \tag{3-11}$$

Note that the expected response time for the nth request, $\varpi(t + n\delta, a(j_n), l(j_n))$ depends on the preceding *n-1* assignments and corresponding requests. Consequently the calculation of the expected cost $\phi_\pi(i)$ is very time-consuming. If the fleet contains J ambulances then the total number of possible assignments is J^{N+1} and for each assignment after the first one, all possible requests need to be considered in the model. Consequently, the computational complexity for this problem is $O(J^{N+1}OD^{N})$, where J is the fleet size, and *OD* is the number of feasible origin-destination pairs (Bell et al. (2005)).

Due to Bellman's "curse of dimensionality", finding the optimal policy that minimizes the expected total response time over the rolling horizon is very computationally demanding. When real operational requirements and the flexible dispatching strategy (diversion, rerouting and relocation) are taken into consideration, the complexity of the problem will increase dramatically and it becomes impractical to obtain optimal solutions for large fleets in real-time. Therefore, we consider a rolling horizon approach as a practical solution method. As time unfolds, static problems which approximate the expected response time are solved repeatedly over the events found within the horizon.

3.3 Components and Properties

3.3.1 Emergency Response Vehicle Fleet

Emergency Response Vehicle Fleet includes three main types of vehicles, which are considered in this study:

- Fire engines: located in Fire Stations or outside on/off duty;

- Ambulances: are located in Fire Stations and Emergency Rescue Centers. They may also be located at hospitals. There are two service types:
    - Advanced Life Support (ALS): National Fire Protection Agency (NFPA) defines it as functional provision of advanced airway management including intubations, advanced cardiac monitoring, manual defibrillation, establishment and maintenance of intravenous access, and drug therapy (NFPA 1710, 2001 Edition).
    - Basic Life Support (BLS): NFPA definition defines it as functional provision of patient assessment, including basic airway management; oxygen therapy; stabilization of spinal, soft tissue, and shock injuries; stabilization of bleeding; and stabilization and intervention for sudden illness, poisoning and heat/cold injuries, childbirth, CPR, and automatic external defibrillator (AED) capability (NFPA 1710, 2001 Edition).
- Police cars: part of the police car patrol along the streets, partly reside in Police Stations.

The National Fire Protection Agency (NFPA) develops, publishes, and disseminates more than 300 codes which serve as guidelines in real operations. Table 2 below outlines the NFPA's standards for response times.

Table 3-1: NFPA Guidelines, Response Times

Fire Suppression Incident		Emergency Medical Incident	
First Arriving Engine Company Total Response Time	Full First Alarm Assignment Total Response Time	First Responder Unit Total Response Time	Advanced Life Support (ALS) Unit Total Response Time
5 minutes	9 minutes	5 minutes	9 minutes
90% Achievement Rate	90% Achievement Rate	90% Achievement Rate	90% Achievement Rate

According to financial year 2002 data from the International City Manager's Association (ICMA), of the cities surveyed with populations over 100,000, on average only 68% of emergency fire calls were responded to within five minutes. Response rates vary widely, with Berkeley, California responding to 100% of its emergency fire calls within five minutes, San Francisco, California responding to over 90% of fire emergency calls within five minutes, and other cities such as Orlando, Florida responding to only 55% of its fire emergency calls within five minutes.

3.3.2 Emergency Calls

Different emergency calls need different types of attention. In general, emergency calls can be divided into 5 categories: fire, crime report, traffic accident, medical and others. The fire and traffic accident calls usually involve all three types of vehicles; calls reporting crimes usually require police cars and ambulances, where emergencies of medical nature may require ambulances only. When ambulances are required, some may need to transfer the sick/injured people to hospital while the others may not.

In real applications, emergency analysis is done manually by a group of analysts. Emergency calls can be categorized into several classes according to their nature. Each class requires a different number of vehicles and vehicle types. The analysts will assign vehicles according to the emergency call codes. Since different types of emergencies have different priorities, the required vehicles should reach the emergency sites within suggested/required time limits. The higher the priority, the shorter the time limits.

3.3.3 Assumptions and Constraints

When assigning vehicles, we need to consider the required number of vehicles, vehicle types, destination and route selection at the same time, meanwhile, the vehicles need to be dispatched to the emergency site with many other operational concerns. For example, the vehicles should arrive at the emergency location within a certain time limit. If a vehicle is reassigned from a call to another one, this diversion should have certain level of positive impact on the overall system; and when the emergency vehicles are dispatched, the available vehicles should be relocated, if needed, to those areas where coverage is lacking due to current vehicle assignments.

Therefore, the mathematical model has to solve three problems simultaneously:

- Assign vehicles to emergencies;
- Determine the routes; and,
- Provide coverage for the whole area.

The model has to accomplish the following subject to operational constraints:

1. Minimize the total response time, which includes dispatching and travel times.
2. Ensure that sufficient number of vehicles can arrive at the emergency locations within waiting time limits.
3. Ensure that emergency calls can be serviced with the appropriate number and types of vehicles..
4. Ensure that vehicle diversions are made with a minimum level of positive impact on the overall system performance that outweighs the unnecessary burdens that such diversions can cause.
5. Ensure that the entire service region has good service coverage.

To clarify the problem, we abstract the real street network as a graph with *n* nodes and *m* directed links. The following assumptions are based on the real-world operations:

- Location of emergencies: it is assumed that the emergencies happen at nodes in the abstracted network only, and this assumption is reasonable when the street network is detailed enough.

- Type of emergencies: five types of emergencies are considered in this model. Each type of emergency has an upper bound for response time, hospital requirements, and the number and type of vehicles required.
- Type of vehicles: three types of vehicles are considered in this model. These are ambulances, fire engines and police cars.
- Coverage: each type of emergency vehicle has a certain coverage area, which is represented by the number of nodes that can be reached by that type of vehicles within certain time limit.
- *Availability* of vehicle: when a vehicle is dealing with an emergency call on-site or when the vehicle runs out of its supply (water, medicine or gas), it cannot be dispatched to any other tasks. For different types of vehicle, this availability varies:
    1. Ambulances: when an ambulance is available in station or after getting recharged in hospitals, they are *available* for an emergency call. For those on the way to an emergency site, when anther call requires the same type of ambulance, they are *available* for diversion if necessary.
    2. Fire engines: we assume the fire engines need to be recharged after each on-site treatment if it is not a false call. So fire engines are only *available* when idle in station or on-route to an emergency.
    3. Police cars: when a police car is in service, we assume only when the police car is servicing an emergency on-site it is *unavailable*. Otherwise, it is *available*.

- *Divertible* vehicles: among the available vehicle, those that are on the way to an emergency location, on the way back to depot, or are leaving to respond to an emergency, are characterized as having "*divertible*" status. This means that these vehicles can be reassigned to a new destination if the overall system benefits from the diversion.

The following summarizes the above.

1. The ambulance on the way to a hospital is *divertible*, but its destination must be another hospital.
2. The vehicle on the way to an emergency location is *divertible*.
3. The police car leaving an emergency site is *divertible*.
4. The ambulance on the way back to station without an on-site treatment (false call or leaving a hospital, recharged) is *divertible*.
5. To divert any vehicle, there must be certain benefit to the system.

There are three groups of constraints:

1. Operational Constraints: the waiting time of an emergency call should be within a pre-defined response time window. Vehicles will return to their original station after an assignment if no other assignment is made. The entire service region must have a minimum coverage level by the system.
2. Vehicle Constraints: Some of the emergency response vehicles must get reloaded or charged after providing service. These vehicles cannot provide service until they are replenished. One vehicle can only be assigned to one call at a time.

3. Hospital or other facility Constraints: The number of patients sent to one hospital within a time interval cannot exceed its capacity. The severity of the injuries, the distance to hospitals, and the hospital vacancies together determine the destination hospital for the injured patients. In some cases, for example, in incidents that result in multiple injuries, some patients may not be sent to the nearest hospital. In these cases, we need to optimize the allocation of patients to hospitals. Therefore, the travel time from emergency site to hospitals as well as hospital capacities are considered in the model.

3.4 Mathematical Model

3.4.1 Notation

Based on the above assumptions and objectives, the real-time emergency response vehicle dispatching problem is formulated as an integer programming model. Though we consider three types of vehicles in this study, the model is formulated in a way that more types of vehicles can be accommodated. The following notation is defined:

V The set of emergency vehicles in the system

K The set of emergency vehicle types in the system, $k = 1,2,...,N_K$

V_k The set of available type k emergency vehicles in the system

$V_k^s(t)$ The subset of the emergency vehicles in V_k that are leaving for station at time t

$V_k^e(t)$ The subset of the emergency vehicles in V_k that are leaving for an emergency at time t

$V_k^h(t)$ The subset of the emergency vehicles in V_k that are leaving for hospitals after finishing on-site service at time *t*

j The index of vehicles in set V_k , $j = 1,2,...,N_{V_k}$

$W(t)$ The set of emergencies in the system at time *t*

N_w The size of the set of emergencies waiting for emergency vehicles

i The index of emergencies in set W , $\mathrm{i} = 1,2,...,\mathrm{N_w}$

S The set of emergency vehicle stations

N_s The size of the set of emergency vehicle stations

s The index of stations in set S , $s = 1,2,...,N_s$

H The set of hospitals in the system

N_h The size of the set of hospitals

h The index of hospitals in set H , $h = 1,2,...,N_h$

L The set of nodes in the area

l The index of nodes in N , $l = 1,2,...N$

$N_{V_{jk}}^R$ The set of nodes that can be reached by type *k* vehicle *j* within required time

Coefficients

T_{ki} The upper bound of waiting time for type k vehicle to reach emergency *i*

$t_{kji}(t)$ The predicted travel time for type *k* vehicle j to arrive at emergency i while departing at time t

$t_{kjh}(t)$ The predicted travel time for type *k* vehicle j to arrive at hospital h while departing at time t

$t_{kjs}(t)$ The predicted travel time for type *k* vehicle j to arrive at station s while departing at time t

A_{ik} The penalty associated with the type *k* vehicle dispatched to waiting emergency i whose travel time is longer than T_{ki}

B_{ik} The penalty associated with type *k* vehicle deficiency for emergency *i*

C_{kji} The cost of the type *k* vehicle *j* traveling to emergency *i*, which is a function of travel time $t_{ijk}(t)$ and related to the emergency property and vehicle type property.

C_{kjh} The cost of type *k* vehicle *j* traveling to h^{th} hospital, which is a function of travel time $t_{hjk}(t)$ and related to the property of vehicle type.

C_{kjs} The cost of type *k* vehicle *j* traveling to s^{th} station, which is a function of travel time $t_{sjk}(t)$ and related to the property of vehicle type

$CH_h(t)$ The vacancy of hospital *h* at time *t*

D_k The penalty associated with type *k* vehicle coverage deficiency for the area

M A large positive number

N_{ik} The required number of type *k* vehicle for emergency *i*

τ_1, τ_2, τ_3 The diversion criterion; when the saving of travel time from reassigning a vehicle from one emergency to another is larger than τ_1, the route change will be performed, otherwise, the original assignment will not change; when the saving of travel time from reassigning a vehicle from one station to an emergency is larger than τ_2, the diversion will be performed, otherwise, the assignment will not change; when the saving of travel time for reassigning a vehicle from one hospital to another hospital is larger than τ_3 , the diversion will be performed, otherwise, the original assignment will not change.

$NC_{lsk}(t)$ The indicator of whether of if node *l* can be covered by type *k* vehicle at station *s* at time *t*, =1 if travel time for type *k* vehicle travel from station *s* to node *l* $t_{ksl}(t) <= T_{v_k}$;

=0 otherwise

ρ_k The required coverage rate for type *k* vehicles

ω_{lk} The penalty associated with the coverage deficiency of type *k* vehicle at node *l*

A series of parameters X^0 stand for the destinations of emergency vehicles in the system in last iteration.

X^0_{kji} =1 if the type *k* vehicle *j* was dispatched to emergency *i* in the last step;

=0 otherwise

X^0_{kjh} =1 if the a type *k* vehicle *j* was dispatched to hospital *h* in the last step;

=0 otherwise

X^0_{kjs} =1 if the a type *k* vehicle *j* was dispatched to station *k* in the last step;

=0 otherwise

Decision Variables

$X_{kji}(t)$ =1 if the type *k* vehicle *j* is dispatched to an incident emergency *i* at time *t*;

=0 otherwise

$X_{kjh}(t)$ =1 if the type *k* vehicle j is dispatched to hospital *h* at time *t*;

=0 otherwise

$X_{kjs}(t)$ =1 if the type *k* vehicle *j* is dispatched to station *s* at time *t*;

=0 otherwise

Y_{kj}^{1} =1 if the type *k* vehicle *j* is re-assigned on its way to another emergency;

=0 otherwise

Y_{kj}^{2} =1 if the type *k* vehicle *j* is re-assigned on its way to a station;

=0 otherwise

Z_k =1 if the coverage rate of type *k* vehicle is lower than ρ_k.;

=0 otherwise

P_{kji} =1 if the travel time for the type *k* vehicle *j* to the emergency *i* is longer than

T_{ik} ;

=0 otherwise

Q_{ik} =the number of type *k* vehicles in deficiency for emergency *i;*

=0 otherwise

R_{lk} =1 if the node *l* can be reached by type *k* vehicles within the critical time;

=0 otherwise

3.4.2 A Basic Dispatching Model

A basic mathematical model for dispatching only, denoted as M0, is as follows:

$$Min \quad \begin{aligned} Z0 = &\sum_i \sum_j \sum_k (X_{kji}(t) \cdot C_{kji}(t)) + \sum_k \sum_j \sum_h (X_{kjh}(t) \cdot C_{kjh}(t)) \\ &+ \sum_k \sum_j \sum_s (X_{kjs}(t) \cdot C_{kjs}(t)) \end{aligned} \tag{3-12}$$

Subject to

$$\sum_i X_{kji}(t) + \sum_s X_{kjs}(t) + \sum_h X_{kjh}(t) = 1 \qquad \forall\ j \in V_k, k \in K \tag{3-13}$$

$$\sum_j X_{kji}(t) \geq N_{ik} \qquad \forall \quad i \in W, j \in V_k, k \in K \tag{3-14}$$

$$X_{kji}(t) \cdot t_{kji}(t) \leq T_{ki} \qquad \forall \quad i \in W, j \in V_k, k \in K \tag{3-15}$$

The objective function is to minimize the weighted total travel time at any time *t*, which includes the travel times to stations, travel time to hospitals and travel time to emergency sites. The two constraints (3-14) and (3-15) are to ensure that the emergency will be serviced with required number and type of emergency vehicles and within upper bound of waiting time respectively.

3.4.3 An Extension of the Dispatching Model

In real operation, when several emergencies happen during the same time interval, some vehicles may not be able to reach the emergency sites within the upper bound of waiting time, or not enough vehicles can be dispatched to certain emergencies. In that case, model *M0* may be unable to provide a feasible solution.

An easy extension of the basic model *M0* is proposed to introduce two types of penalties in the objective function.

$$Z1 = \sum_i \sum_k (\sum_j P_{kji} \cdot A_{ik}) + \sum_i \sum_k (Q_{ik} \cdot B_{ik}) \qquad (3\text{-}16)$$

Correspondingly, there are two constraints (3-17) and (3-18) that define those penalty indicators.

$$\sum_j X_{kji}(t) + Q_{ik} \quad \geq N_{ik} \qquad \forall \quad i \in W, j \in V_k, k \in K \qquad (3\text{-}17)$$

$$X_{kji}(t) \cdot t_{kji}(t) + P_{kji} \geq T_{ik} \qquad \forall \quad i \in W, j \in V_k^r, k \in K \qquad (3\text{-}18)$$

Constraints (3-17) ensure that the deficiency in type *k* vehicle for emergency *i* at time *t* will be translated into a penalty. When there are enough type *k* vehicles, no penalty will be added to the objective function because Q_{ik} is a non-negative integer. Constraints (3-18) require that the assigned vehicles reach the emergency site within an upper bound of waiting time, otherwise, waiting time penalties will be added to the objective function. P_{kji} is a binary variable.

Diversion will certainly increase the complexity of the operation, and when diversion is made too frequently, it may confuse the crew. It is necessary that each diversion brings certain benefits to the system. We defined diversions as discussed in Section 3.3.3. Therefore, they should be included in the objective function as three types of diversion penalties (3-19). The corresponding constraints are shown as (3-20), (3-21) and (3-22). Constrains (3-20) ensure that if a type *k* vehicle is diverted from emergency *i*, then a penalty will be introduced to the objective function. Constrains

(3-21) define the penalty of diversion from a station and Constraints (3-22) define the penalty of diversion from a hospital. Constraints (3-23) ensure that one vehicle can have one and only one destination at a time.

$$Z2 = \tau_1 \cdot \sum_k \sum_j Y_{kj}^1 + \tau_2 \cdot \sum_k \sum_j Y_{kj}^2 + \tau_3 \cdot \sum_k \sum_j Y_{kj}^3 \qquad (3\text{-}19)$$

$$1 - X_{kji}(t) \cdot X_{kji}^0 \le \mathrm{M} \cdot \mathrm{Y}_{\mathrm{kj}}^1 \qquad \forall \quad i \in W, j \in V_k^e, k \in K \qquad (3\text{-}20)$$

$$\tau_1 + \sum_i t_{kji}(t) X_{kji}(t) - \sum_i t_{kji}(t) X_{kji}^0 \le M \cdot Y_{kj}^1$$

$$1 - X_{kjs}(t) \cdot X_{kjs}^0 \le \mathrm{M} \cdot \mathrm{Y}_{\mathrm{kj}}^2 \qquad \forall \quad i \in W, j \in V_k^s, k \in K \qquad (3\text{-}21)$$

$$\tau_2 + \sum_i t_{kji} s(t) X_{kjs}(t) - \sum_i t_{kjs}(t) X_{kjs}^0 \le M \cdot Y_{kj}^2$$

$$1 - X_{kjh}(t) \cdot X_{kjh}^0 \le M \cdot Y_{kj}^2 \qquad \forall \quad i \in W, j \in V_k^h, k \in K \qquad (3\text{-}22)$$

$$\tau_3 + \sum_i t_{kjh}(t) X_{kjh}(t) - \sum_i t_{kjh}(t) X_{kjh}^0 \le M \cdot Y_{kj}^3$$

$$\sum_{h=i}^{H} X_{kjh}(t) = 1 \qquad \forall \quad j \in V_k^h, k \in K \qquad (3\text{-}23)$$

Now, we have an extended dispatching model, denoted as *M1*.

$$Min \quad \begin{array}{l} \sum_i \sum_j \sum_k (X_{kji}(t) \cdot C_{kji}(t)) + \sum_s \sum_j \sum_k (X_{kjs}(t) \cdot C_{kjs}(t)) + \sum_k \sum_j \sum_h (X_{kjh}(t) \cdot C_{kjh}(t)) \\ + \sum_i \sum_k (\sum_j P_{kji} \cdot A_{ik}) + \sum_i \sum_k (Q_{ik} \cdot B_{ik}) + \tau_1 \cdot \sum_k \sum_j Y_{kj}^1 + \tau_3 \cdot \sum_k \sum_j Y_{kj}^2 + \tau_3 \cdot \sum_k \sum_j Y_{kj}^3 \end{array} \qquad (3\text{-}24)$$

Subject to

$$\sum_i X_{kji}(t) + \sum_s X_{kjs}(t) + \sum_h X_{kjh}(t) = 1 \qquad \forall\ j \in V_k, k \in K \qquad (3\text{-}25)$$

$$N_{ik} - \sum_{j} X_{kji}(t) \le M \cdot Q_{ik} \qquad \forall \quad i \in W, j \in V_k, k \in K \qquad (3\text{-}26)$$

$$X_{kji}(t) \cdot t_{kji}(t) + P_{kji} \ge T_{ik} \qquad \forall \quad i \in W, j \in V_k\ , k \in K \qquad (3\text{-}27)$$

$$\sum_{s} X_{kjs}(t) = 1 \qquad \forall \quad i \in W^0, j \in V_k^s , k \in K' \qquad (3\text{-}28)$$

$$1 - X_{kji}(t) \cdot X_{kji}^0 \ \le \mathrm{M} \cdot \mathrm{Y}_{\mathrm{kj}}^1 \qquad \forall \quad i \in W, j \in V_k^e , k \in K \qquad (3\text{-}29)$$

$$1 - X_{kjs}(t) \cdot X_{kjs}^0 \ \le M \cdot Y_{kj}^2 \qquad \forall \quad i \in W, j \in V_k^s , k \in K \qquad (3\text{-}30)$$

$$1 - X_{kjh}(t) \cdot X_{kji}^0 \ \le M \cdot Y_{kj}^3 \qquad \forall \quad i \in W, j \in V_k^h , k \in K \qquad (3\text{-}31)$$

$$\sum_{h=1}^{H} X_{kjh}(t) = 1 \qquad \forall \quad j \in V_k^h , k \in K \qquad (3\text{-}32)$$

3.4.4 An Extension of the Relocation Problem

As discussed above, it is important to relocate the available vehicles to potential relocation sites in order to provide coverage to the entire region.

This relocation problem, denoted as *M2* can be formulated as the Maximal Covering Problem. And it will add another two parts to the objective function, which are the number of nodes without good coverage and the penalty of under-coverage for certain types of vehicles. The first part of objective value *Z3* represents the average coverage of a certain type of vehicle in the region while the second part depends on the individual weight of the node.

$$Minimize\ \ Z3 = \sum_{k} D_k \cdot Z_k - \sum_{k}\sum_{l} (\omega_{lk} \bullet R_{lk}) \qquad (3\text{-}33)$$

Subject to

$$\sum_{s}(NC_{lsk}(t)\cdot\sum_{j}X_{kjs}(t))-R_{lk}\geq 0 \qquad \forall \quad k\in K, l\in N, s\in S \qquad (3\text{-}34)$$

$$\rho_k\cdot N_n-\sum_{l}R_{lk}\leq M\cdot Z_k \qquad \forall \quad k\in K, l\in L \qquad (3\text{-}35)$$

$$\sum_{s}X_{kjs}(t)=1 \qquad \forall \quad k\in K, j\in V_k \qquad (3\text{-}36)$$

Constraints (3-34) identify the nodes which can be reached by any type k vehicles dispatched to stations within the critical time. Constraints (3-35) ensure a penalty will be introduced to the objective function when the total number of nodes which are covered by type *k* vehicles is less than a pre-set value. Constraints (3-36) ensure that each vehicle will be dispatched to one and only one station.

3.4.5 A Deployment Model

Based on the extensions, we can have a deployment model, denoted *M3*, which covers the dispatching and coverage concerns. The objective function is a sum of *Z0*, *Z1*, *Z2*, and *Z3*.

$$\begin{aligned} & \sum_{i}\sum_{j}\sum_{k}(X_{kji}(t)\cdot C_{kji}(t))+\sum_{s}\sum_{j}\sum_{k}(X_{kjs}(t)\cdot C_{kjs}(t))+\sum_{k}\sum_{j}\sum_{h}(X_{kjh}(t)\cdot C_{kjh}(t)) \\ Min \quad & +\sum_{i}\sum_{k}(\sum_{j}P_{kji}\cdot A_{ik})+\sum_{i}\sum_{k}(Q_{ik}\cdot B_{ik})+\tau_1\cdot\sum_{k}\sum_{j}Y_{kj}^{1}+\tau_3\cdot\sum_{k}\sum_{j}Y_{kj}^{2}+\tau_3\cdot\sum_{k}\sum_{j}Y_{kj}^{3} \\ & \sum_{k}\sum_{l}(\omega_{lk}\bullet R_{lk})+\sum_{k}D_k\cdot Z_k \end{aligned} \qquad (3\text{-}37)$$

The constraints sets include (3-25) - (3- 32), (3-34) and (3-35). Due to constraints (3-25), (3-36) is no longer necessary in model *M3*.

An important step before generating the formulation is to identify the vehicles' availability. We simplify the structure of the formulation by identifying the vehicle type, vehicle status and determine which are available for dispatching and relocating, so that we do not need to consider all the vehicles in the system at a certain time stamp in the model.

3.5 Formulation and Algorithmic Considerations

In this chapter, the nature of the problem and the important components were discussed, and the mathematical formulations were developed. This problem is formulated as an integer linear model. The model optimizes vehicle dispatching operations while taking real operational requirements into consideration. According to the literature, no similar model has been developed before. To solve the problem, we can either use existing optimization software, such as LINDO or CPLEX, or develop our own algorithms.

The nature of emergency response requires that the problem must be solved within a very short computational time. Since the model is intended for use in a real-time emergency response vehicle dispatching and routing system, it is important to provide good dispatching schemes and route guidance within a short time. It is unreasonable for the operation staff to wait five minutes or more to obtain an optimal solution. Long computational time will definitely increase the preparation time and the total response time, and deteriorate the over-all system performance. Since the General Assignment and the Facility Location Problems are NP-hard, this Emergency Deployment Problem as a combination of those two problems. The computing time increases exponentially when the problem size increases. Existing optimization software may be a good alternative for a small system that is not congested, but impractical when the system is larger and busier. The long computing time may be caused by three aspects: the network size (|N|), the combination of emergencies and vehicle fleet size (|W|, |V|) and the travel times used in the model. The size of network |N| denotes the complexity of the vehicle re-location problem for the service coverage concern. When |W| or |V| increases, the nature of the assignment problem causes the computing time to increase too. The model is updated dynamically based on the results of travel time prediction and shortest path algorithm. The computational time of the travel time prediction algorithm and the shortest path algorithm is another bottleneck for the total computational time.

With respect to a lower bound, based on the formulation, we can use Lagrangian Relaxation (LR) to relax the coverage constraints and change the problem to minimize the relaxed objective function subjects to the other constraints. By this relaxation, we avoid dealing with location problem but the dispatching problem only. Unfortunately, LR cannot provide good lower bounds in this case. Linear Relaxation is applicable as well to obtain a lower bound. Another approach may be decomposition. By decomposing the original problem *M3* to several parallel sub-problems and deriving the lower bound for each sub-problem, a lower bound for the original problem can be obtained. This will be discussed in more detail in Chapter 5.

Chapter 4: Solution Approaches and Heuristics

In this chapter, we discuss the solution approaches and heuristic algorithms. Since our objective is to solve a real-world problem, we need to address two issues:

1. Solution quality, and
2. Computational time.

We apply a rolling horizon approach to achieve an approximate model at each time stamp, which reduces the problem size. To solve the mathematical model, we start with an initial feasible solution. A good feature of this problem is that it is easy to get feasible initial solutions. Two initialization methods are suggested, which are both based on the greedy algorithm. Although more sophisticated methods could apply, we do not see any differences in the results of experiments. Several improvement methods are discussed after that. All these improvement methods are based on the three basic operations: ADD move, DROP move and SWAP move.

In this chapter, we will also discuss how to use Tabu Search strategy to do further extensive search in our improvement methods.

4.1 Rolling Horizon Approach

4.1.1 Framework

As described in Section 3.2.2, the large number of decision variables and the dynamic feature of this problem indicate that the classical dynamic programming techniques are not feasible for computing the value of the relevant functions. A rolling horizon approach is more practical and promising.

In a rolling horizontal approach, instead of considering the future dynamic features, we focus on the current assignment plan with one step look-ahead capability, that is, we relocate the available vehicles to better cover the area in order to achieve less response time for near future emergency calls.

We can solve the mathematical models described in Chapter 3 iteratively when the system is updated at each time step. This updating can be done when a new call comes in or when an emergency is dealt with the fleet, or when a certain time interval is reached and the traffic information updating is completed. The “horizon” used in this approach is not the time between fixed events or a fixed time interval, it is a combination of events and time intervals, which also can be defined as the timing advance of the system. The details will be discussed in Chapter 6.

4.1.2 Size of Variables and Computational Time

As mentioned in Chapter 3, the mathematical model may be solved by commercial optimization software, e.g. CPLEX or LINGO. Though an optimal solution may be provided, an obvious concern is the computational time. The potential factors that will affect the computational time include:

- the size of network |N|;
- the fleet size (available vehicles) |V|;
- the number of waiting emergencies in the system |W|; and

- the number of potential relocation sites |S|.

Several sets of experiments are designed here to investigate the relationship between problem size and computational time. Note the fact that if the potential relocation sites are limited to the stations, the network size will not influence the computational time when the travel times are provided independently. In real operation, for certain types of vehicles, e.g. police patrol cars, the relocation sites can be any node on arterials. Therefore, we use a network with 1757 nodes, which is abstracted from a real-world road network. The map of this network is shown as in Figure 4-1.

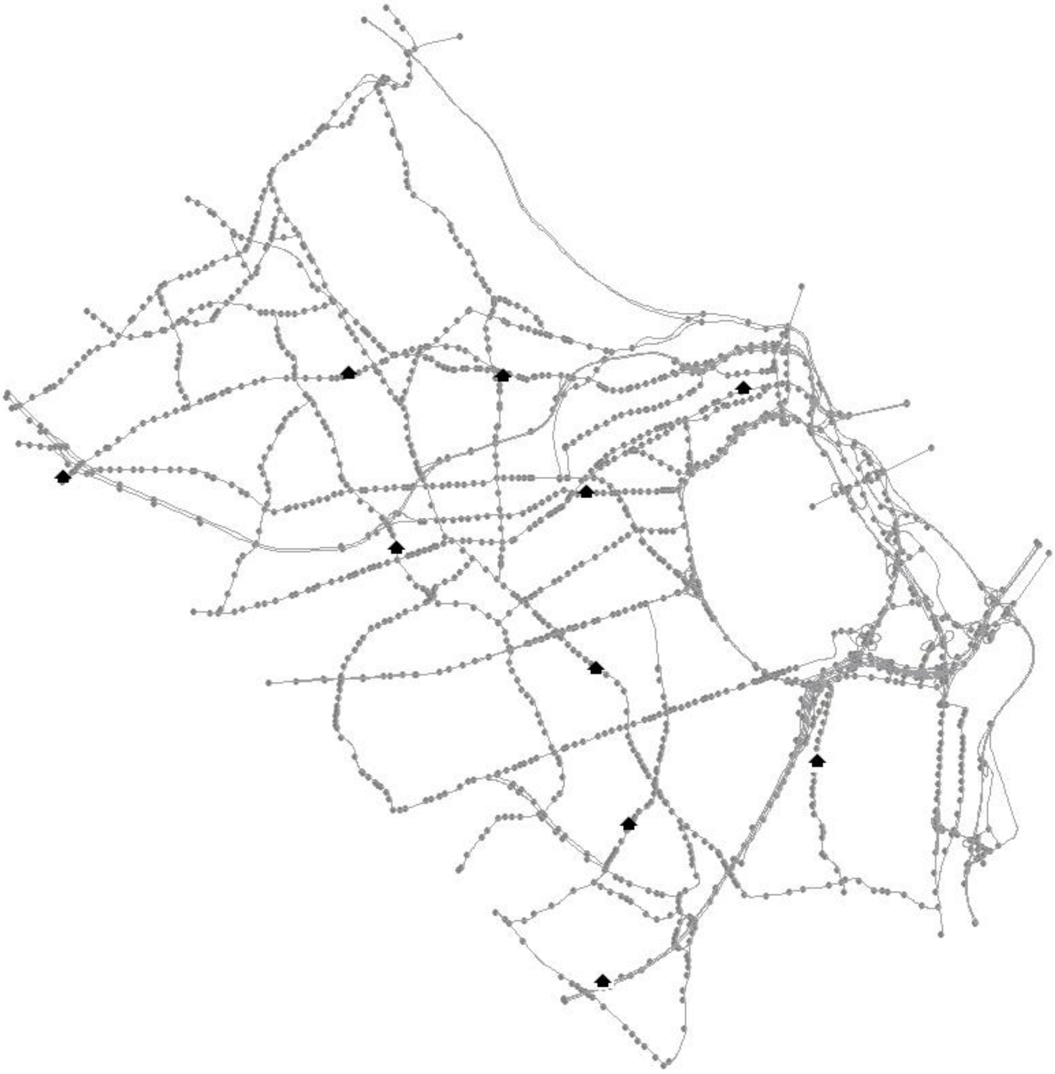


Figure 4-1: A Sample Network for Analyzing Computational Time

Table 4-1: Experiments for CPLEX Computational Times

Problem Size |V|, |W|, |S|	Dispatching without coverage concern (M1)		Deployment Problem (M3)	
	Number of Constraints & Variables	Average Computational Time (sec)	Number of Constraints & Variables	Average Computationa l Time (sec)
(10, 1, 10)	41/141	<0.005	1777/1817	0.01
(10, 5, 10)	74/204	<0.005	1831/1961	0.27
(10, 1, 500)	520/5020	0.09	2277/6777	11.47
(10, 1, 1757)	1777/17590	0.24	3534/19347	135.33
(30, 10, 10)	349/909	0.02	2106/2666	3.36
(30, 10, 50)	389/2009	0.03	2146/3866	2.13
(30, 10, 100)	439/3609	0.06	2196/5316	31.88
(30, 1, 500)	560/15060	0.26	2317/16817	51.70
(30, 1, 1757)	1817/52770	1.11	3853/55067	N/A
(100, 10, 10)	1119/3009	0.05	2876/4766	1.83
(100, 10, 100)	1209/12009	0.25	2966/13766	31.88
(100, 30, 10)	3229/16029	0.31	4896/8786	2.66
(100, 30, 500)	1609/52009	1.36	5386/57787	N/A
(100, 30, 1757)	4886/181729	5.25	N/A	N/A

We testd 14 problem sizes on a Pentium 4 PC with 2.40 GHz CPU and 1 GB of RAM. For each problem size, with different locations of vehicle and emergency sites, the computational time may vary. We solved five problems for each problem size with randomly generated vehicle locations, emergency locations and emergency types.

Table 4-1 shows the computational times for each group of experiments. The numbers of variables and constraints are listed and an average computational time is calculated for each problem size. Generally speaking, the dispatching problems are solved quickly. When there are 100 vehicles, 30 waiting emergencies and 10 stations, the average computational time is still less than 1 second. This problem size is much larger than the size of dispatching problem in real operations. Usually, the number of waiting emergencies in the system is less than five. However with the relocation/coverage concern, the computational times increase dramatically with the size of the candidate relocation set. When we limit the relocation site to be the stations only, say, |S|=10, the computational time is less than 3 seconds. When the candidate relocation set increases to 1757, with 10 vehicles, the computational time is more than two minutes. Another issue that requires attention is the reading time for CPLEX. When the number of relocation sites increases, the size of the problem file increases dramatically. For the problem size (100, 10, 100), the size of the problem file is about 100 M bytes, and it takes more than one minute for CPEX to read this file, though the solution time is only 32 seconds.

The results show that the computational time for this problem is not simply related to the numbers of variables or constraints but also to the structure of the problem. The results also indicate that for certain types of vehicle such as ambulances and fire engines, the system can rely on the existing optimization software since their relocation sites are limited to stations or hospitals in the region and the fleet size is relatively small. But for police cars, which can be relocated to any nodes on the main streets and arterials, it requires a solver that can provide quality solutions within a shorter computational time, e.g. 30 seconds.   Therefore, an efficient heuristic is in great need.

4.2 Initial solution

When a severe emergency happens, there is no time to be wasted in waiting for an improved solution. An intuitive method is to use all the available vehicles for emergency mitigation and recovery. It requires fast and good quality initial solutions.

In this section, we will discuss two different initialization methods. The first one is based on the exact solution of the dispatching problem from CPLEX and a greedy search algorithm for the relocation problem. The second one is a hierarchical greedy search algorithm.

4.2.1 Initial Solution 1: CPLEX based algorithm

As shown in Section 4.1, for police cars, the computational time of a dispatching problem in CPLEX is always less than five seconds, while the coverage constraints are the part that causes the dramatic increase in computational time.

First, we decompose the original problem into two sub-problems: dispatching problem and relocation problem and solve theses two sub-problems (*M1, M2*) sequentially, namely, dispatch first, relocate second.

We can input model *M1* into CPLEX and read the optimal solution out. Then we examine the coverage of the current dispatch scheme. If all the nodes in the network are covered by the remaining available vehicles, an optimal solution has already been achieved. This happens when the emergency call rate is low and there are enough well located available vehicles in the area.

Lemma 4.1

When the deployment problem is decomposed into two sub-problems (M1, M2) and solved sequentially, if with the optimal solution of M1, all nodes are covered then this solution is an optimal solution for the original problem (M3).

Proof:

When all the nodes are covered by the dispatching scheme from *M1*, the objective function value of model *M2* is 0. *M1* is a relaxation of *M3*, therefore the objective function value of the optimal solution of *M1* is a lower bound for that of *M3*, that is, $Z1 \leq Z3$. Since $Z1 + 0 \leq Z3$, the initial solution from *M1* is an optimal solution of *M3*.

4.2.2 Reduction of Problem Size

Before starting with the relocation algorithm, we would like to apply the rolling horizon feature to the problem to reduce the size of the candidate relocation site.

Due to the dynamic feature of this emergency vehicle response system, the status of the components in this system will be frequently updated after some small time interval. The fleet deployment and routing scheme will be optimized after each updating. For example, if a police car is assigned to a site 10 minutes away from the current location, the assignment might be changed when the next call comes or when the traffic information updating is done. As a result, for police cars, which can be relocated to any nodes on the main street network, it is not necessary to consider all the nodes in the network as the candidate set, but only those nodes which are reachable within a certain time contour. As shown in Figure 4-2, the original map contains more than 1757 nodes, while within three minutes time contour, there are only about 212 nodes for vehicles at station 101. In this way, we reduce the size of the set of candidate relocation sites significantly. As if this time step is larger than other system update time intervals, the solution is a realistic one and will not waste the mobility of the vehicle.

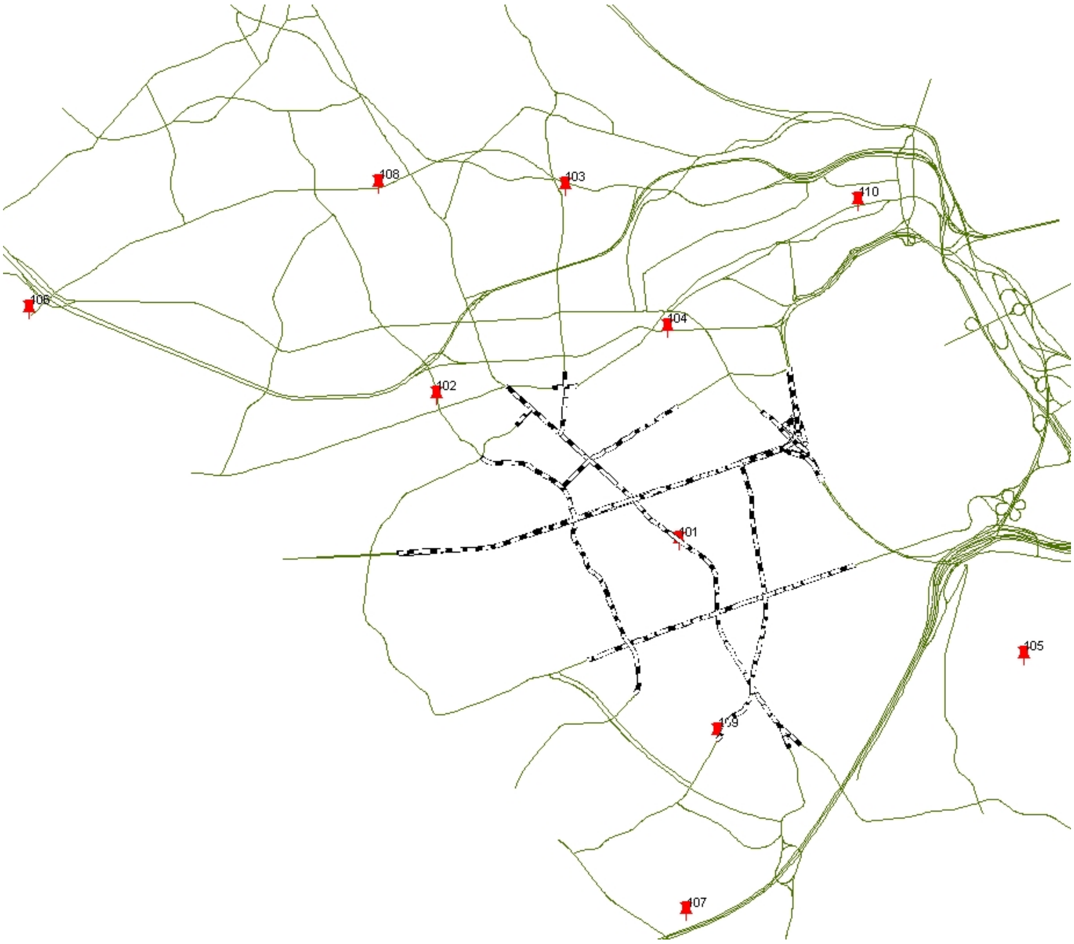


Figure 4-2: An Example of Size Reduction of Candidate Relocation Sites

With this modification, we can model the deployment problem as *M3'* which is similar to *M3* but with a slight difference compared with the original format in Section 4.2.1, that is, the assignable relocation sites are those within a time interval ε, instead of the entire set of candidate relocation sites.

It is noticeable that the file size of the deployment problem is reduced, and the computational time for the same problem size is reduced as well. Table 4-2 shows the comparison of file sizes, file reading times and computational times between the original format of deployment model and the rolling horizon deployment model.

Table 4-2: Comparison of CPLEX Computational Times of *M3* and *M3*'

Problem Size |V|, |W|, |S|	Original Deployment Problem (M3)		Rolling Horizon Deployment Problem (M3')	
	Reading Time (seconds)	Average Computationa l Time (sec)	Reading Time (seconds)	Average Computationa l Time (sec)
(10, 1, 10)	0.05	0.01	0.04	<0.005
(10, 1, 500)	2.38	11.47	1.69	3.55
(10, 1, 1757)	122.00	135.33	14.17	13.88
(30, 1, 10)	0.13	1.26	0.06	0.25
(30, 1, 500)	87.81	51.70	9.41	10.08
(30, 1, 1757)	N/A	N/A	81.66	136.22
(100, 10, 10)	2.66	1.83	0.23	0.98
(100, 10, 100)	82.67	51.88	5.06	5.91
(100, 30, 500)	N/A	N/A	87.19	212.38
(100, 30,1000)	N/A	N/A	318.17	442.53
(100, 30, 1757)	N/A	N/A	N/A	N/A

As seen in Table 4-2, for a fleet size of 100, CPLEX cannot read the problem file with 500 candidate relocation sites. But for *M3'*, CPLEX can provide optimal solution for problem sizes as large as (30, 1, 1757) and (100, 30, 1000). Although the reading times and solution times are too long for the on-line application, the solutions may be used for the comparison of heuristics developed in later sections.

4.2.3 Greedy Search for Vehicle Relocation

For the Maximal Covering Problem, many heuristics have been developed (Galvao, et. al. (1996, 2000); Lorena, et. al. (2001)). These algorithms use Lagrangian /Surrogate heuristics and obtain good results. However the computational time is long for large size problems. In our algorithm, we start with an easy greedy search algorithm, which provides feasible solutions very quickly.

For each type of vehicle, we define the degree of coverage at node *i* as:

$$DC_i = \sum_j NC_{ij} \tag{4-1}$$

where DC_i represents the number of vehicles that can reach node i within the upper bound of waiting time. Here the upper bounds of waiting times are only relevant to vehicle types but not emergency types.

When moving an available vehicle from its current location to a new location, the utility of the move is defined as the increased number of covered nodes.

$$U(n') = \sum_{l \in L} R_l^{'} - \sum_{l \in L} R_l \tag{4-2}$$

$R_j = 1$ when $DC_i \geq 0$, 0 otherwise.

For example, as shown in Figure 4-3, in an area with 12 nodes, the dark nodes represents locations of vehicles, and the solid circle represents the current coverage area of a vehicle. Therefore, there are seven covered nodes. If vehicle *i* at node 6 moves to node 9, it will be able to cover node set {5, 6, 8, 9, 10, 11}, shown as the dashed circle, and a total of 9 nodes will be covered. The utility of this move is 2.

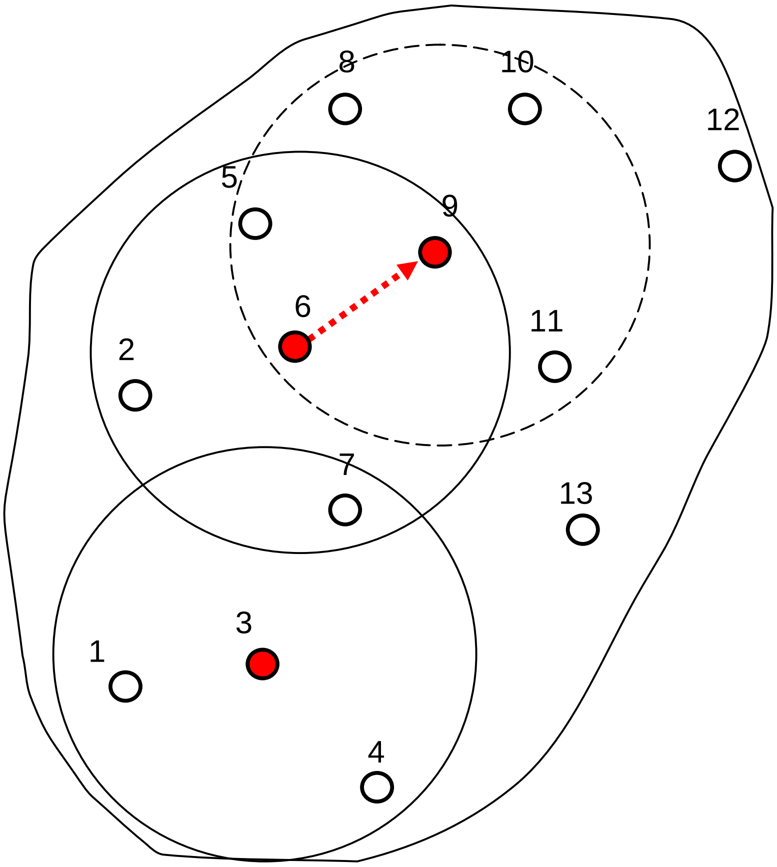


Figure 4-3: An Example of the Utility of a Move

The algorithm for this initialization method, denoted as *G1*, can be described as follows:

For all vehicles that are not assigned to an emergency call, and for each type of vehicle,

1. Identify nodes (l=1, …, N) which are not covered by vehicle type *k*, until all nodes are covered or there is no available vehicles;

2. For all nodes, do:

   Search the closest available type *k* vehicle *j* and increase the rank of vehicle Ri by 1, until all the nodes are examined.

3. Sort $R(i \in V_K)$ in an non-decreasing order,

4.a Select the unassigned vehicle with the Max(Rj),

4.b For all the nodes n' in the time contour of *i,* do:

4.c Compute U(n'), until all nodes are examined;

4.d Find Max U(n'),

4.e If U(n')>0, assign vehicle *j* to node n', remove vehicle *j* from available vehicle list and go to step 1.

Else, remove vehicle *j* and go to step 4.a.

5. Record the current solution as the initial solution and stop.

Figure 4-4 shows the flow chart for this method.

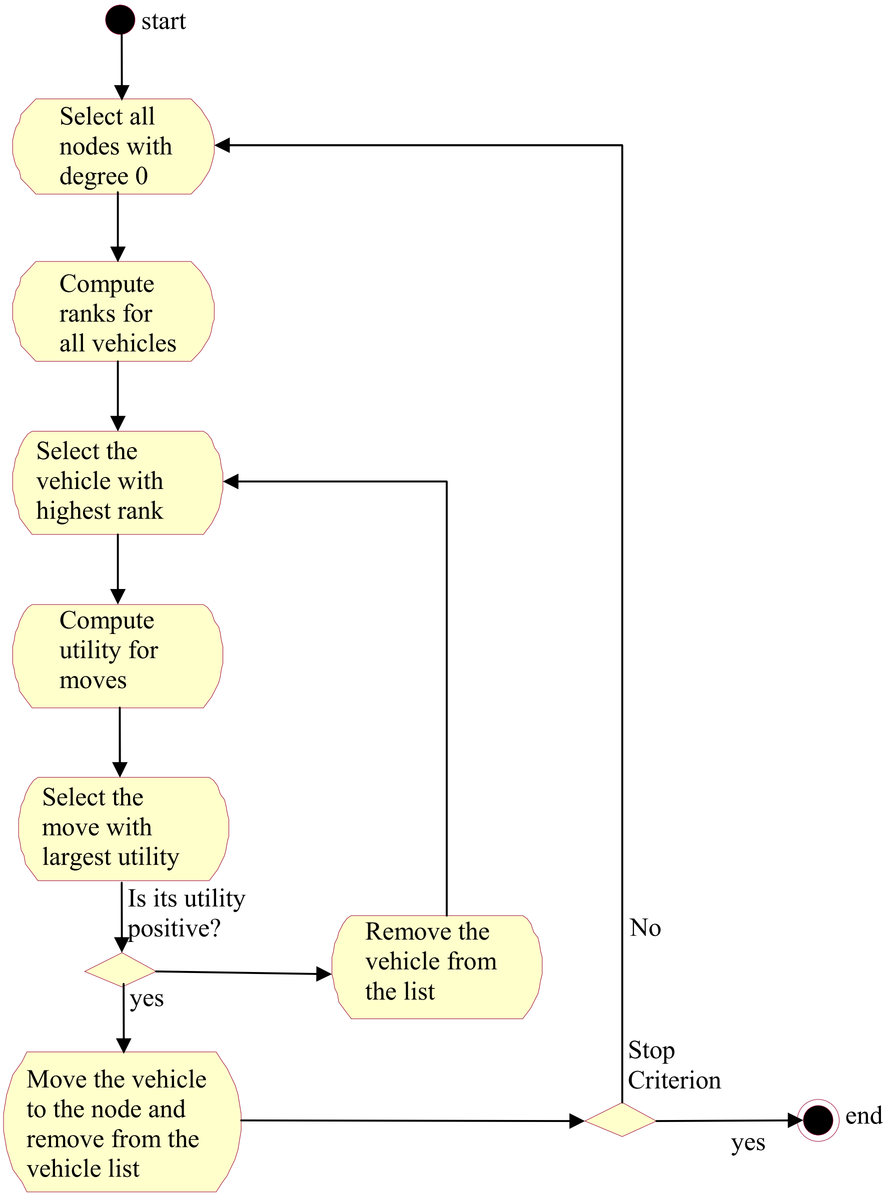

start
Select all nodes with degree 0
Compute ranks for all vehicles
Select the vehicle with highest rank
Compute utility for moves
Select the move with largest utility
Is its utility positive?
yes
Remove the vehicle from the list
Move the vehicle to the node and remove from the vehicle list
No
Stop Criterion
yes
end

Figure 4-4: Flow Chart for Initialization Solution I

Note that the solution obtained from the above procedure is already feasible, and when there is no uncovered node, this is an optimal solution.

4.2.4 Initial Solution 2: Greedy Search Algorithm

The first initial solution is based on an optimal solution of dispatching problem from existing software. Since sometimes the time for the existing software to read a large dispatching problem is too long, we suggest another greedy algorithm to solve the dispatching problem as well, and then use the same greedy search procedure in Section 4.2.3 on the relocation part.

The simple greedy search algorithm (G2) is outlined as follows:

For each type of vehicle,

1. Sort waiting emergencies by priority in descending sequence;

    If there are emergencies in the same priority, sort by their existing waiting time in system in an non-increasing order;

2.a Select the first emergency on the list, until all the vehicles are dispatched, or all the emergencies has been assigned;

2.b Calculate travel time from available vehicles to this emergency; if the vehicle was assigned to another emergency, add a penalty to the travel time;

2.c Assign the closest vehicles to the emergency and remove it from the available vehicle list, until the required number of vehicles are satisfied;

2.d Remove current emergency, go to step 2.a.

3. Call G1.

Figure 4-5 shows the flow chart for this algorithm.

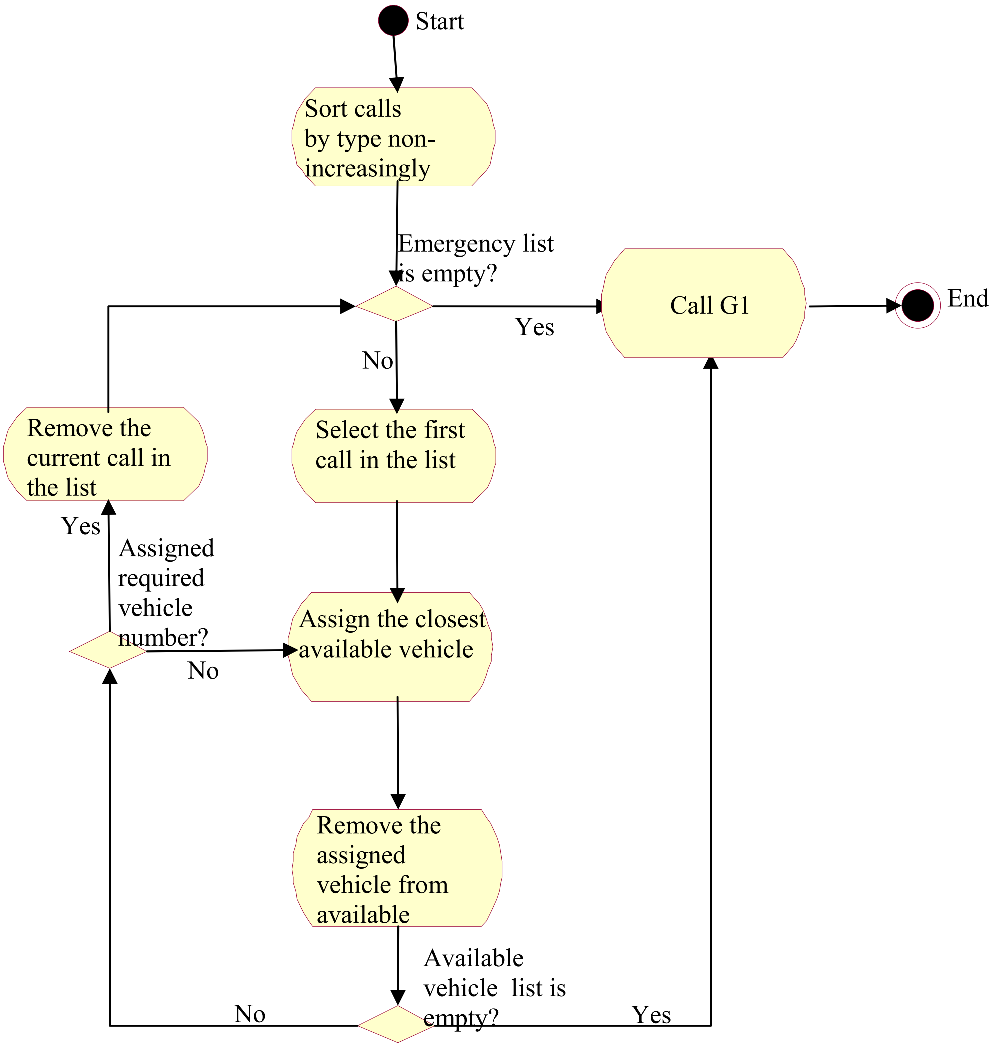

Start
Sort calls by type non-increasingly
Emergency list is empty?
Yes
No
Call G1
End
Select the first call in the list
Remove the current call in the list
Yes
Assigned required vehicle number?
No
Assign the closest available vehicle
Remove the assigned vehicle from available
Available vehicle list is empty?
No
Yes

Figure 4-5: Flow Chart for Initialization Method II

The advantage of this initialization method is that it does not require any commercial optimization software, it may provide good solutions very quickly, and the solution is feasible as well.

4.3 Improvement Methods

4.3.1 SWAP move

The idea of the SWAP move is similar to a two-opt move in the Traveling Salesman Problem. In this method, any two vehicles with different assignment will exchange their destinations and the new objective function value will be recalculated. The lower value will be kept and the corresponding deployment plan will be recorded.

The advantage of this improve method is its simplicity and it does not require much computational time. The complexity of this move is $O(|V|^2)$. This improvement method is not very effective for the CPLEX based initialization method, but is useful when we are using the greedy search for the assignment problem. The limitation of this improvement method is that it is restricted to a one-to-one exchange, and this will easily trap the search to a local optimal.

The SWAP move will also be used in the next improvement method. In the Tabu Search heuristics, SWAP can be used for intensification, and if we perform random SWAP moves which do not require the examination of all possible pairs, it can be deemed as a type of diversification. The flow chart of the SWAP improvement methods is shown in Figure 4-6.

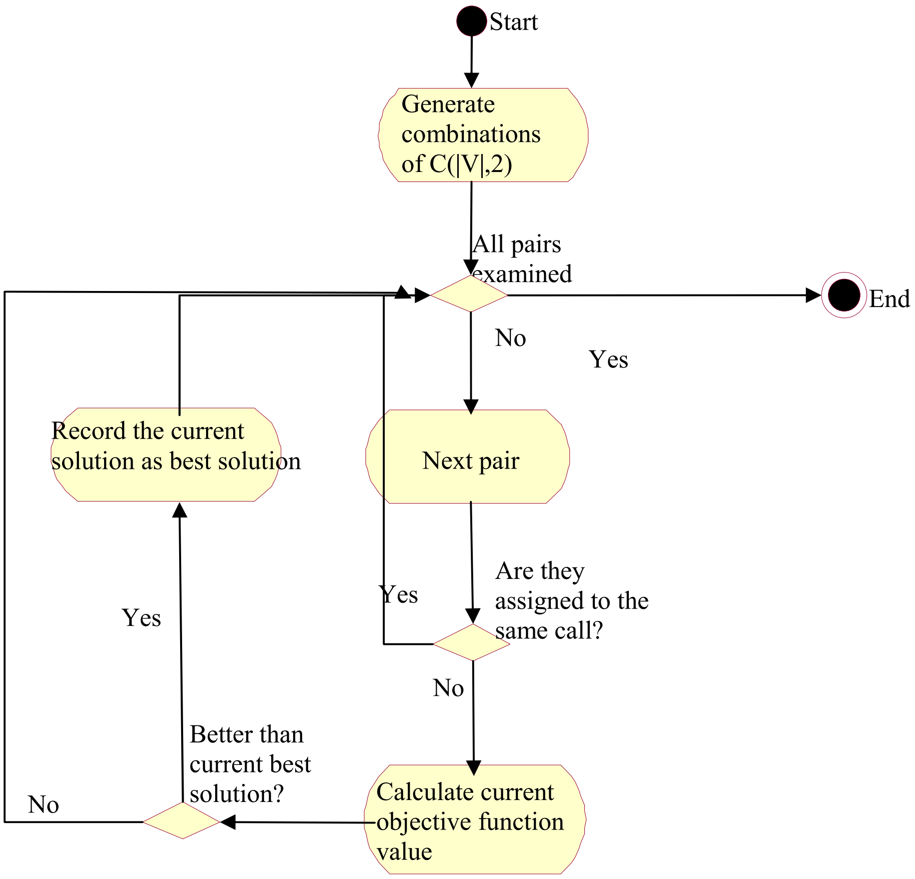

Start
Generate combinations of C(|V|,2)
All pairs examined
End
No
Yes
Record the current solution as best solution
Next pair
Yes
Are they assigned to the same call?
Yes
No
Better than current best solution?
No
Calculate current objective function value

Figure 4-6: Flow Chart for SWAP Improvement Method

4.3.2 The framework of Tabu Search

The tabu search heuristics searches the space of solution and the Tabu list helps the searching procedure avoid getting trapped in local optimal. A generic Tabu search algorithm can be summarized in Figure 4-7. In the following sections we describe in detail basic operation moves, the global search strategy, the neighborhood and tabu tenure.

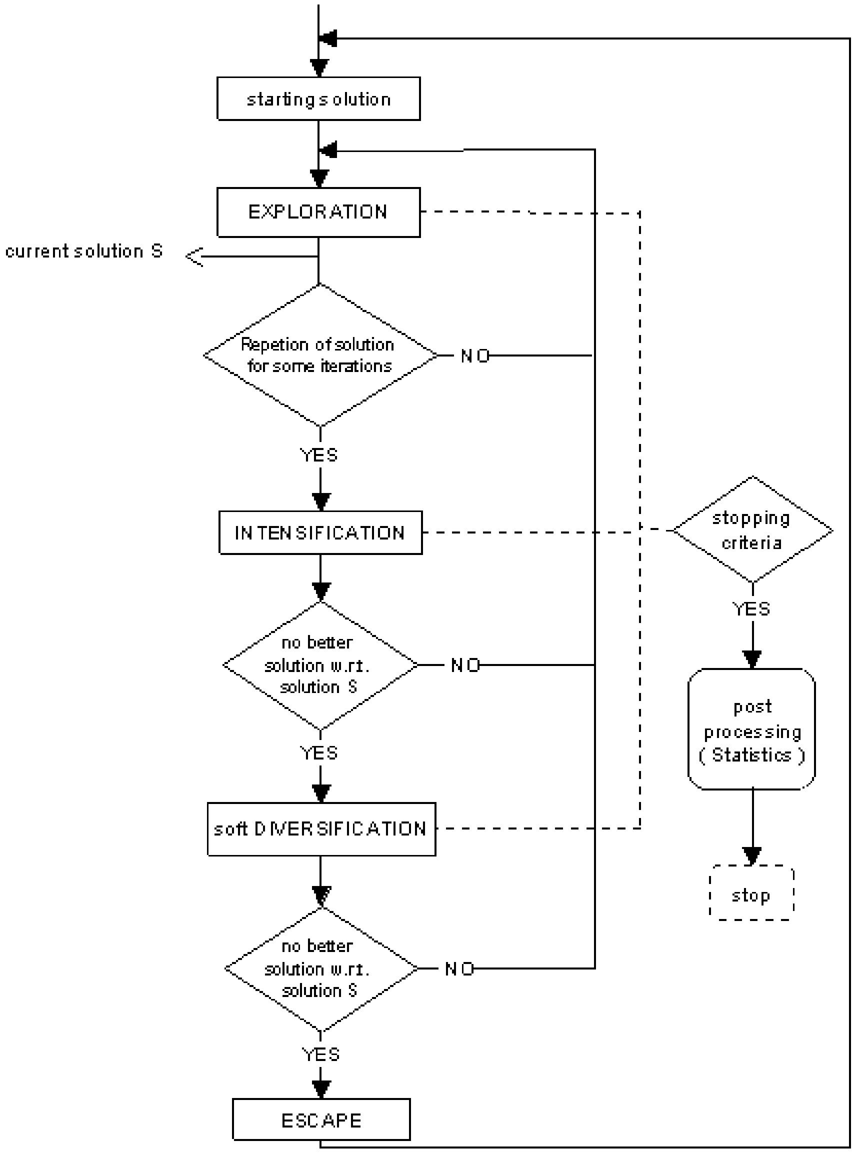

starting solution
EXPLORATION
current solution S
Repetion of solution for some iterations
NO
YES
INTENSIFICATION
no better solution w.r.t. solution S
NO
YES
soft DIVERSIFICATION
no better solution w.r.t. solution S
NO
YES
ESCAPE
stopping criteria
YES
post processing (Statistics)
stop

Figure 4-7: Generic Framework of Tabu Search Heuristics

4.3.4 ADD Move

There are two basic operators in this heuristic: ADD move and DROP move. The ADD move starts with an emergency which has the vehicle with the largest potential to increase coverage. By adding as many vehicles as possible to an emergency, opportunities are provided to balance the dispatching and coverage concerns. The procedure of an ADD move is described as follows:

Add Move

For all available vehicles which were not assigned to emergencies V':

1. Compute the rank of the assigned vehicles to the uncovered nodes as in G1;

2. Select the emergency call which has the highest rank vehicle.

3. Get the travel time from each vehicle to this emergency call;

4. Sort the travel times in non-decreasing order of their travel times;

5. Add all available vehicles with $t_{ij}<=T_{ijk}$ to the emergency call,

   If no vehicle with $t_{ij}<=T_{ijk}$, add the closest vehicle to the call

6. Update the available vehicle list;

Figure 4-8 shows the flow chart of this procedure.

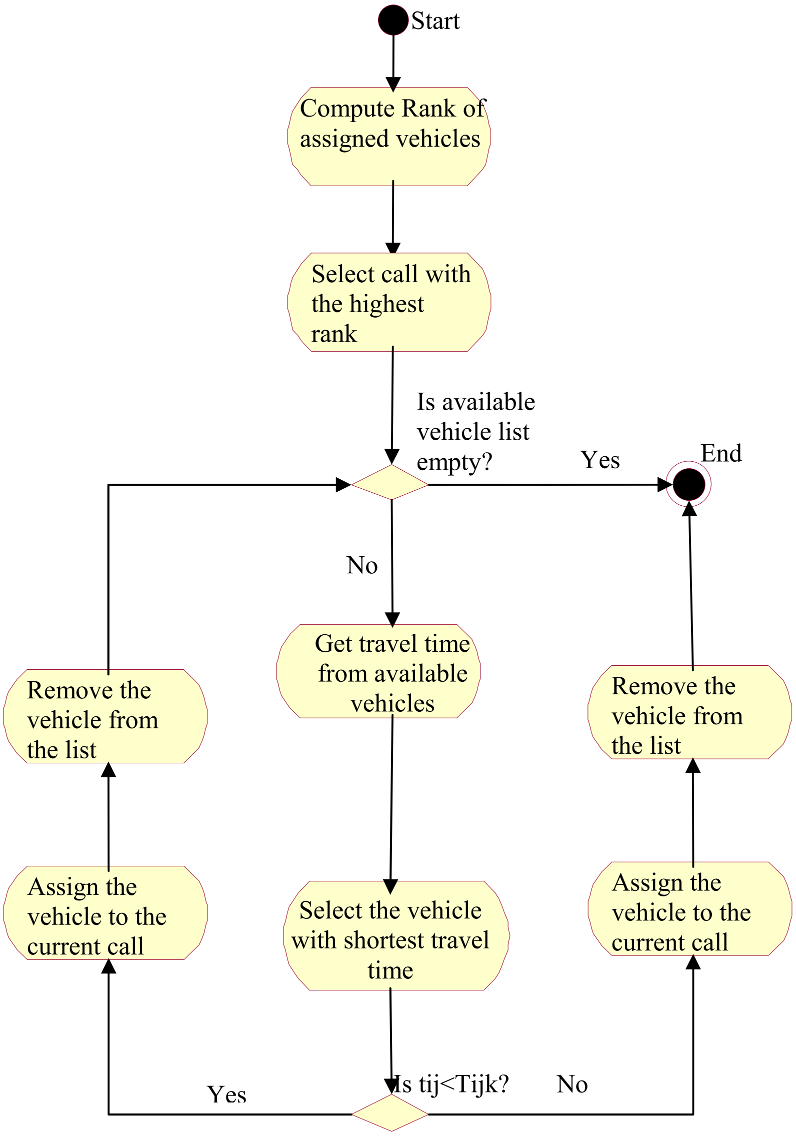

Start
Compute Rank of assigned vehicles
Select call with the highest rank
Is available vehicle list empty?
Yes
End
No
Get travel time from available vehicles
Remove the vehicle from the list
Remove the vehicle from the list
Assign the vehicle to the current call
Select the vehicle with shortest travel time
Assign the vehicle to the current call
Yes
Is tij<Tijk?
No

Figure 4-8: Flow Chart for ADD Move

4.3.4 DROP Move

The DROP move is on the counter side of the ADD move. Instead of adding all possible vehicles to an emergency call, a DROP move tries to relocate more vehicles to the candidate relocation sites when there is any benefit. Each dropping is associated with a cost. This cost can be the increase of the total travel time to a call, the penalty of breaking the constraints of required number of vehicles for a certain call, or the penalty of exceeding the upper bound for the waiting time.

The procedure of a DROP move is listed below and the flow chart of a DROP move is shown in Figure 4-9:

Drop Move

1. Get ranks (R1) for all assigned vehicles as in G1;

2. Compute their rank (R2) for the cost of dropping from assignment;

3. For each vehicle, set R=R1+R2;

4.a Select the vehicle with Max(R) and delete it from the rank list;

4.b Drop the first vehicle in the rank list from the assignment and relocate with G1 ;
    Compute the new objective function value Z';

4.c If Z'>Z0: remove the vehicle from the current assignment; Else go to step 4.a.

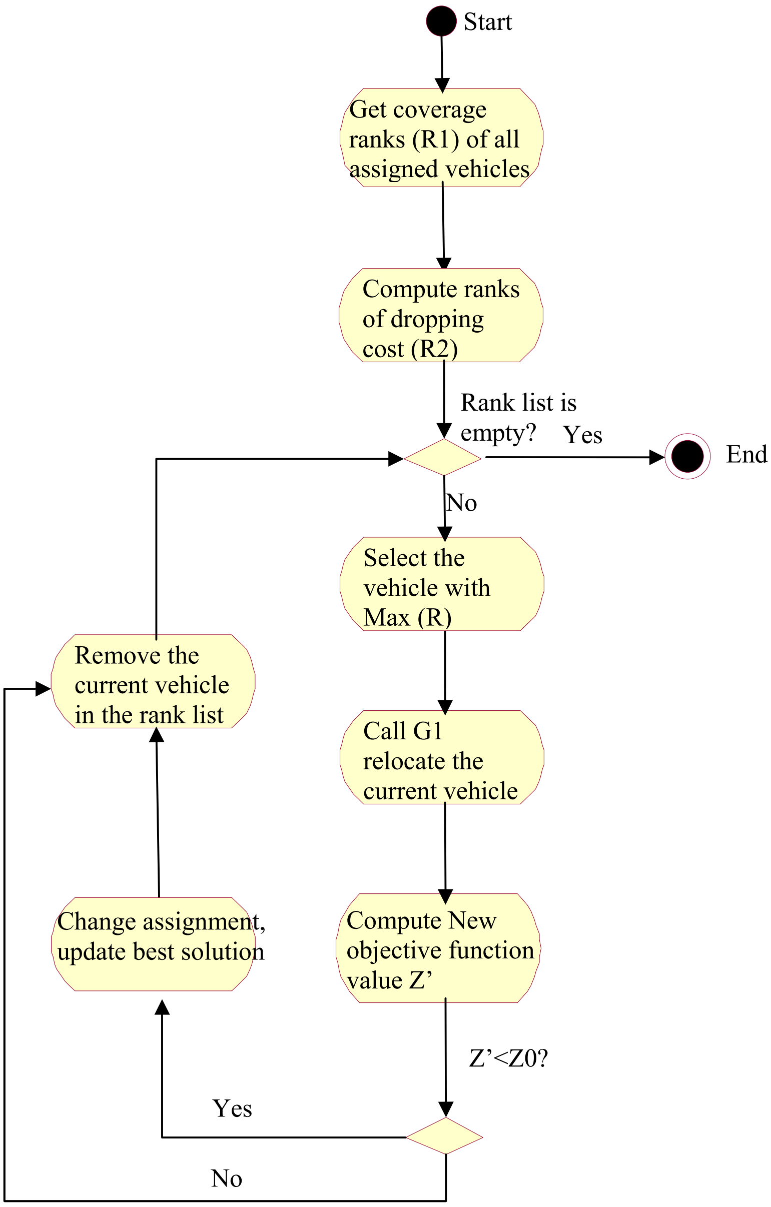

Start
Get coverage ranks (R1) of all assigned vehicles
Compute ranks of dropping cost (R2)
Rank list is empty?
Yes
End
No
Select the vehicle with Max (R)
Remove the current vehicle in the rank list
Call G1 relocate the current vehicle
Change assignment, update best solution
Compute New objective function value Z'
Z'<Z0?
Yes
No

Figure 4-9: Flow chart for DROP move

4.3.6 Search Strategy

The global search strategy mainly switches between ADD/DROP neighborhoods. . Preliminary experiments have shown that performing a tabu search with SWAP moves does not bring much improvement. But a SWAP move is still essential to the good performance of the algorithm. First, it can intensify the search in a potentially good area and it allows the method to jump out of the current configuration (diversification) and may bring a good configuration for the next round of ADD/DROP move.

The procedure of the complete algorithm is described as below:

1. Initialization (G1 or G2)

    with initial solution *S0=S'*.

    Set *S* := *S0* , *f** := *f(S0)*, *S** := *S0* , *T* := Ø;

2. While termination criterion not satisfied (CPU time limit or n1 moves are performed without improvement)

    a. Sl*=+∞;

    b. Perform ADD/DROP tabu search until n2 consecutive iterations are performed without improvement to Sl*;

    c. (Intensification) Perform SWAP until a local minimum is reached;

    d. S*=Sl*;

    e. Expire Tabu List when the move limit is reached;

    f. (Diversification) Perform random SWAP;

3. Report best solution.

4.3.7 Neighborhoods, Tabu Type and Tabu Tenure

Different from other well-known problems, such as the traveling salesman problem or the vehicle routing problem, in which the neighborhood usually are links, the neighborhood in this deployment problem are the moves made by vehicles. Two different neighborhoods are explored. The first neighborhood consists of moves where a single vehicle is dispatched to an emergency (ADD) or to a relocation site (DROP). The second neighborhood is based on SWAP moves, where two vehicles will exchange their destination. The reduction of the size of the candidate relocation site dramatically reduces the runtime of the algorithms.

If we only use an ADD move and a DROP move, it is quite possible that the procedure will be trapped in a small area of the whole region. To avoid local inferior areas, we need to know what kinds of these areas exist, namely, tabu types. We use typical tabu types which are suggested by Glovers (1993) to avoid cycling. Different tabu types are defined. Each represents different local inferior areas that we should avoid. The tabu types are defined according to the three basic operations: ADD, DROP and SWAP moves. For example, if a vehicle is assigned to a call, in the next several iterations, dropping the vehicle is forbidden. By doing this, cycling is avoided.

All tabu are stored in the tabu list. The tabu list T records the |T| last vehicles added or dropped from the solution. This prevents the reversal of their status as long as they remain in the list. Since a SWAP move might be seen as a combination of one ADD move and on DROP move, the vehicles involved in any SWAP move are also recorded in the tabu list. Each time when ADD/DROP tabu search are used, the tabu list is kept in its current state and is not reinitialized.

A tabu will not be kept in the tabu list all the time, because this may cause loss of solution space for other good solutions. After certain time or certain number of iterations, the tabu should expire so that the algorithm may search these areas and apply the improvement methods again.

The duration of certain time or the number of iterations are called tabu tenure. Tabu tenure is an important parameter in a tabu search algorithm. Too large or too small tenure duration may not provide good solution. Variable tabu tenures are suggested by some researchers in their algorithms. But those algorithms usually do not have tight computational time limits. To simplify our algorithm, we just use fixed tabu tenure. The length of the tabu tenure is a random value chosen within a pre-defined interval. We select the tabu tenure by performing preliminary computational experiments. In Chapter 5, we will compare the solutions' quality and computational times with different tabu tenures.

4.4 Conclusions

In this chapter, we started with a comparison of the problem sizes and corresponding computational times. We noticed that the size of the set of potential relocation sites dramatically increases the computational time. A rolling horizon approach results in effective reduction of size of the set o f relocation sites. Two initialization methods were suggested based on the decomposition of the deployment problem. The original problem was decomposed into two sub-problems: dispatching problem and relocation problem; and they were solved sequentially. One initialization method is based on the optimal solution of CPLEX on the assignment problem the other one is based on a hierarchical greedy search algorithm. A simple swap procedure was used to improve the initial solution. The simple improvement operations can deliver better solutions very quickly but may be trapped in local inferior solutions. Therefore, meta-heuristics were introduced to avoid local inferior solutions. Three basic operations are used in this improvement method. Tabu search strategy helps the algorithm search more solution space while keeping the memory of the ‘good’ areas. Chapter 5 shows lower bound analysis and the comparison of these algorithms.

Chapter 5: Lower Bounds

5.1 Analysis of Lower Bound Algorithms

To measure the quality of the heuristic solutions, the most straightforward way is to compare them with the optimal solutions. The Multidimensional General Assignment Problem and the Maximal Covering Problem both are NP-hard problems. The emergency vehicle deployment problem is a combination of these two problems. The computational experiments show that the computational time increases with the size of the set of the candidate relocation sites. For large size problems it is hard to obtain the optimal solution due to the large size of the problem files. Therefore, we need to develop lower bound methods for the emergency vehicle deployment problem.

Most lower bound methods in the literature are specially designed for a single problem. There is no existing lower bound method for the emergency vehicle deployment problem as formulated in this dissertation in the literature. The difficulty of finding the lower bound algorithm suggests that we focus on the characteristics of the emergency vehicle deployment problem. Although doing this will limit our lower bound method just to this specific problem, the ideas behind the method may be applied to other problems also.

There are three standard lower bound methods:

- The linear programming relaxation, in which only the integrality constraints are relaxed – the objective function remains the same as the one in the original function.
- Lagrangean relaxation, in which the feasible set is usually required to maintain 0-1 feasibility, but some constraints are moved to the objective function with a penalty term, and
- The branch and bound method, in which the feasible region is driven to several smaller feasible sub-regions.

In the following section, we will analyze the details of using various lower bound methods.

5.2 Linear Relaxation Method

In general, a lower bound on the optimal solution value can be obtained by solving a relaxation of the optimization problem. Namely, one solves another optimization problem whose set of feasible solutions contains all feasible solutions of the original problem and whose objective function value is less than or equal to the true objective function value for a minimization problem for points feasible to the original problem. Thus, we replace the "true" problem by one with a larger feasible region that is more easily solved.

For small to medium size problems, we can obtain optimal solutions from CPLEX, though the computational time might be long. For large size problems, either CPLEX cannot read the files or it cannot provide optimal solutions. For those problems which can be read but for which integer optimal solutions cannot be obtained, the linear relaxation method can be applied.

We relax four groups of variables, the coverage penalty indicators (u_{ik}), the waiting time penalty indicators (Q_{kji}), the penalty indicators (P_{ik}) required for the number of vehicles and the assignment variables (*X, Y*). As expected, because of the formulation, the solution will vary only when relaxing the waiting penalty indicators. CPELX is able to provide solutions to the relaxed problems when the problem size is (100, 30, 1000), and which cannot be solved as deployment problems. We call the lower bounds obtained by linear relaxation “type I” lower bounds.

The Linear Relaxation is a very simple lower bound technique. The disadvantage is that the lower bound obtained is loose.

5.2 Decomposition

Since the real-world size dispatching problem in this study is solvable using CPLEX, the coverage part is the bottleneck for this problem. The simplest way is to remove the coverage part. In this case, we can reduce the problem identical to the dispatching problem *M1*, and thus, we will get a “loose” lower bound by ignoring the coverage constraints. Another intuitive method is to decompose the problem into two sub-problems. Similar to the initialization method described in Chapter 4, the first sub-problem is a dispatching problem (M1), where the vehicle set includes all the available vehicles in the system; and the second sub-problem is the relocation problem (M2). Note the vehicle set in M2 is the same as that in M1, which means a vehicle might be used both for an emergency call assignment and a relocation assignment. We denote this lower bound as type II lower bound.

Lemma 5.1

By decomposing the deployment problem (M3) to two sub-problems (M1, M2), with the same vehicle sets in both sub-problems, when the two sub-problems are solved independently, the sum of the objective function values from M1,and M2 is lower than that of the original problem (M3).

Proof:

The objective function Z3 of model *M3* can be decomposed into two parts Z3(1) and Z3(2) as follows without violating the constraints:

$$\begin{aligned} Z3(1)= & \sum_i \sum_j \sum_k (X_{kji}(t) \cdot C_{kji}(t)) + \sum_k \sum_j \sum_h (X_{kjh}(t) \cdot C_{kjh}(t)) \\ & + \sum_i \sum_k (\sum_j P_{kji} \cdot A_{ik}) + \sum_i \sum_k (Q_{ik} \cdot B_{ik}) + \tau_1 \cdot \sum_k \sum_j Y_{kj}^1 + \tau_3 \cdot \sum_k \sum_j Y_{kj}^2 + \tau_3 \cdot \sum_k \sum_j Y_{kj}^3 \end{aligned} \tag{5-1}$$

$$Z3(2)= \sum_s \sum_j \sum_k (X_{kjs}(t) \cdot C_{kjs}(t)) + \sum_k \sum_l (\omega_{lk} \bullet R_{lk}) + \sum_k D_k \cdot Z_k$$

In *M1*, the relocation constraints are relaxed, there for Z1< Z3(1). Similarly, in *M2*, the decision variable X has a larger feasible region, that is, Z1 ≤ Z3(2). Since Z1 + Z2 ≤Z3, the sum of solutions from *M1, M2* is a lower bound of the solution of *M3*.

Table 5-1 shows the comparison of exact solutions and lower bounds provided by the linear relaxation on the waiting time penalty indicator (Lower Bounds I) and by the decomposition (Lower Bounds II). When there are not enough vehicles to deal with the waiting emergency calls in the system, type I lower bounds will provide tighter bounds. When vehicles in the system are more than the total required number of vehicles for calls, type II lower bounds are tighter, as shown in the last five cases.

Table 5-1: Comparison of Optimal Solutions and Lower Bounds Obtained with Linear Relaxation

Problem Size |V|, |W|, |S|	Optimal Solution Value	Lower Bounds (I)	GAP	Lower Bounds (II)	GAP
(10, 1, 1757)	37.95	37.95	0.00	37.95	0.00
(10, 3, 1757)	190.28	190.28	0.00	117.92	0.38
(10, 5, 1757)	425.22	425.22	0.00	226.80	0.47
(30, 10, 1757)	291.70	229.41	0.21	220.55	0.24
(100, 10, 100)	111.75	86.15	0.23	106.98	0.04
(100, 10, 500)	98.72	82.79	0.16	93.01	0.06
(100, 30, 100)	763.32	326.67	0.57	762.16	0.00
(100, 30, 500)	881.32	326.17	0.63	805.96	0.09
(100, 30, 800)	881.32	326.17	0.63	805.96	0.09

5.3 Lagrangian Relaxation Method

The notation and integer formulation of the Lagrangian Relaxation is shown as follows(Fisher, 1981):

Original Problem:

$$Obj: \quad Z = min\ cx \tag{5-2}$$

$$s.t. \quad Ax <= b \tag{5-3}$$

$$Dx <= e \tag{5-4}$$

$$x >= 0 \text{ and integral.} \tag{5-5}$$

Lagrangian Relaxation Problem:

$$Obj: \quad Z_D(u) = min\ cx + u(Ax - b) \tag{5-6}$$

$$s.t. \quad Dx <= e \tag{5-7}$$

where $x >= 0$ and integral, u is a non-negative vector of Lagrange multipliers.

Since $u >= 0$, and x^* is an optimal solution of the original problem, the objective function of the Lagrangian Relaxation problem can be separated into two parts. Part A is the original objective function *cx* and Part B is the relaxation part *u(Ax-b)*. The second part is always negative or zero, then $Z_D(u)$ <= *Z by observing:*

$$Z_D(u) <= \mathrm{c}\,x^* + u(Ax^* - b) <= Z \tag{5-8}$$

What we get by minimizing this new function is a certain value which depends on *u* and provides a lower bound. Therefore we want to maximize this value with respect to *u* because maximizing this value provides a better Lagrangian lower bound. However, the computational experience shows that part B is always very large and the best bound appears when *u*=0. This means that the problem is relaxed to a pure dispatching problem without relocation, which is model *M1*. Therefore, for emergency vehicle deployment problem, it is very hard to get a good lower bound using Lagrangian Relaxation.

5.4 Upper Bounds

Compared to lower bounds, upper bounds cannot provide a very good solution quality measure unless the gaps between upper bounds and optimal solutions can be defined. However, in case lower bounds are not tight enough, upper bounds may provide a reasonable benchmark for comparison.

In the emergency vehicle deployment problem, the relocation constraints caused the difficulty in obtaining good lower bounds. When the number of potential relocation sites are limited, the problem size will be smaller and become solvable. This strategy takes advantage of the fact that all vehicles can only travel to the nodes within a certain time. By reducing the time allowed, the vehicles will be restricted to fewer relocation sites, and the objective function will be an upper bound for the original problem.

5.5 Solution Analysis

To test the solution quality of the heuristics, 15 groups of computation experiments are designed. In each group, we have 5 problems with the same problem size, but randomly generated vehicle location, emergency call location and emergency types. Each problem has its own parameter set.

5.5.1 Analysis of Computational Times

The computational time is one of the most important concerns in this problem. Table 5-2 shows the computational time comparison between each group. Since in the Tabu Search algorithm we defined the total computational time as one of the stopping criteria, the upper bound of computational time of this algorithm will be a pre-set value. From the test, it is shown that with the initial solution and improvement methods, the computational time is always smaller than 30 seconds, even when there are 100 vehicles, 30 waiting emergency calls and a network that has more than 5000 nodes. The fleet size and the number of waiting emergency calls are larger than the volumes in a real system. This shows the computing speed of these algorithms can satisfy real-world operational needs.

Table 5-2: Comparison of Computational Time

Problem Size	Assignment by CPLEX	Deployment by CPLEX	G1 with Swap	G1 with Tabu Search
Size (|V|, |W|, |S|)	Computation Time (seconds)	Computation Time (seconds)	Computation Time (seconds)	Computation Time (seconds)
(10, 30, 1)	0.03	0.52	1.17	1.28
(10, 30, 10)	0.05	0.82	2.36	3.11
(500, 80, 10)	0.36	241	10.56	14.98
(1574, 100, 30)	0.82	N/A	14.49	21.34
(5496, 100, 30)	0.83	N/A	18.67	28.72

5.5.2 Comparison with Lower Bounds

In this comparison, we list the best solutions of three algorithms with the best lower bound obtained for 3 large size problems. Table 5-3 illustrates the heuristic solutions and the corresponding optimal solutions or the lower bounds when the optimal solution is unavailable. The node number of nodes represents the nodes that cannot be reached by any emergency vehicles within required time. When the problem is solvable by CPLEX, the exact solution is used as the lower bound, and when the problem size is large, we select the larger one of the type *I* and type *II* lower bounds. It is shown that the greedy search initialization with a tabu search improvement algorithm always provides the best solution among the three algorithms.

Table 5-3: Comparison of Solution Quality

Size (1757, |E|, |V|)	(1757, 10,10)			(1757,10, 30)			(1757, 10, 50)			(1757, 30,100)		
	Total Travel Time (min)	# of Node	O. F. Value /Gap	Total Travel Time (min)	# of Node	O. F. Value /Gap	Total Travel Time (min)	# of Node	O. F. Value /Gap	Total Travel Time (min)	# of Node	O. F. Value /Gap
Lower Bound	27	298	255/ 0%	57	24	361/ 0%	48	0	253 /0%	134	0	1121/ 0%
G1	25	1757	480/ 88%	57	414	460/ 24%	48	192	290/ 14%	134	113	1234/ 10%
G1 & Swap	28	1462	362/ 42%	62	278	422/ 16%	52	130	284/ 12%	146	49	1178/ 5%
G1 & Tabu	27	496	316/ 24%	58	216	408/ 13%	50	74	274/ 8%	138	33	1151/ 3%

When the fleet size is large, the computational time usually is longer but the total number of un-covered nodes and the total travel time decrease. The gaps between the lower bounds and heuristic solutions are between 5%-10%. This is because when there are more available units, the improvement algorithm has a larger space to explore better solutions. When the system has a small fleet size and more calls are waiting, the number of uncovered nodes increases dramatically. In this case, the gaps between the heuristic solutions and lower bounds increase as well.

The performance of the Tabu Search heuristic also depends on the parameters in the algorithms. In next sections, we will compare the performance of the heuristics based on the variable stopping criteria and variable time contours used in the problem size reduction.

5.5.3 Analysis of Parameters in Heuristics

The heuristic solution is greatly influenced by the parameters in the algorithm, such as the stopping criteria, and the time contour used for problem size reduction. We used the same problem sizes as in Section 5.5.2. For each problem size, demands and vehicle locations are randomly generated. Table 5-4 shows solutions with variable combinations of stopping criteria *n1, n2* and t_{cpu}, and Table 5-5 shows the results under different pre-set time contour values in the problem size reduction step. The same lower bounds as in Section 5.5.2 are used.

A 3-minute time contour is selected for the variable stopping criteria experiments. Table 5-4 indicates that the stopping criteria combination (30, 1, 2) out-performs the other combinations in most cases. In the variable time contour experiments (shown in Table 5-5), this stopping criteria combination (30, 1, 2) is used as well. It is noticed that when the time contour is large, the computational time increases so that the stopping criterion of 30 seconds cpu-time is reached more often. The results of larger time contours may provide better solutions when the computational time needed to complete the heuristic is within the pre-defined cpu-time (30 seconds). Due to the fact that the average response time in the real operation is about 3 minutes, a 3-minute time contour is selected for the algorithms, and the algorithm is used in the simulation model discussed in Chapter 6.

Table 5-4: Comparison of Solutions under Variable Stop Criteria

Problem Size (1757, |E|, |V|)	(1757,10, 30)				(1757, 10, 50)				(1757, 30,100)			
Parameters (t_{cpu}, *n1*, *n2*)	Total Travel Time (min)	Un-covered Node	O.F. Value/ Gap	Comp. Time (sec)	Total Travel Time (min)	Un-covered Node	O. F. Value/ Gap	Comp. Time (sec)	Total Travel Time (min)	Un-covered Node	O. F. Value/ Gap	Comp. Time (sec)
(15, 1, 1)	58.57	224	451/ 24.3%	14.12	48.54	88	342/ 35.2%	15	135.62	48	1211/ 8.0%	15
(15, 2,1)	58.32	224	451/ 24.3%	14.98	48.13	92	359/ 41.9%	15	135.62	48	1211/ 8.0%	15
(15, 1, 2)	57.32	216	408/ 12.7%	15	48.13	92	359/ 41.9%	15	136.83	43	1186/ 5.9%	15
(15, 2, 2)	57.32	216	408/ 13.9%	15	48.13	92	359/ 41.9%	15	135.62	48	1211/ 8.0%	15
(30, 1,1)	58.57	224	451/ 24.3%	14.12	48.54	88	342/ 35.2%	20.11	137	42	1187/ 5.9%	28.41
(30, 1, 2)	58.32	224	451/ 24.3%	14.98	49.63	74	274/ 8.3%	21.34	138.16	33	1151/ 3%	28.72
(30, 2, 1)	57.32	216	408/ 12.7%	15.05	49.63	74	274/ 8.3%	21.46	138.16	33	1151/ 3%	28.80
(30, 2, 2)	57.32	216	408/ 12.7%	15.76	49.63	74	274/ 8.3%	21.99	139.42	35	1176/ 3%	30

Table 5-5: Comparison of Solutions under Variable Time Contours

Problem Size (1757, |E|, |V|)	(1757,10, 30)				(1757, 10, 50)				(1757, 30,100)			
Time Contour (min)	Total Travel Time (min)	Un-covered Node	O.F. Value/ Gap	Comp. Time (sec)	Total Travel Time (min)	Un-covered Node	O. F. Value/ Gap	Comp. Time (sec)	Total Travel Time (min)	Un-covered Node	O. F. Value/ Gap	Comp. Time (sec)
1	58.87	351	650/ 80.1%	10.37	48.54	213	472/ 87.6%	12.13	135.03	78	1292/ 15.3%	14.32
2	57.57	247	492/ 36.3%	14.98	49.05	127	382/ 51.0%	19.34	135.67	47	1219/ 8.7%	24.75
3	57.32	216	408/ 12.7%	15.05	49.63	74	274/ 8.3%	21.46	138.16	33	1151/ 3%	28.80
4	57.32	216	408/ 12.7%	18.36	49.63	74	274/ 8.3%	28.72	136.64	42	1187/ 5.9%	30
5	56.79	208	372/ 8.2%	24.12	48.54	72	263/ 5.2%	30	136.64	42	1187/ 5.9%	30

5.6 Conclusions

The complexity of the emergency vehicle deployment problem is mainly caused by the relocation/coverage concern. In this chapter, we discussed several lower bound methods. By decomposing the deployment problem into two sub-problems, we can obtain acceptable lower bounds. This method is applicable for the situation that the system has many available vehicles. However, for the system with small vehicle fleet and high system burden (few available vehicles), the lower bound of objective function value provided by this strategy is not tight enough.

In the next chapter, we will apply these algorithms in a simulation system.

Chapter 6: Simulation Model

This chapter describes the components and details of a simulation model developed for evaluating the performance of the proposed fleet management system. The simulation model is calibrated using real-world network and operational data. A case study which uses the proposed system and simulation model is presented in Chapter 7.

6.1 System Module

6.1.1 System Framework

With the mathematical model and heuristics which solve the optimization problem at a given time, a simulation model can be used to mimic the real-world situations and evaluate the performance of this proposed system. In general, this system can be modeled as a G/G/c/∞/∞ (general inter-arrival time/general service time/c servers/unlimited waiting capacity/unlimited population) system. The simulation model should be able to accommodate different dispatching strategies and different shortest path and optimization algorithms.

The simulation model has the following five operational modules:

1. The *emergency call module* where an emergency call is generated at a node;
2. The *travel time generation module* where the all-to-all travel times are calculated;
3. The *vehicle module* where the status of each vehicle is recorded and updated;

4. The *optimization module* where the optimization model is called and solution is retrieved;

5. The *statistical module* where the important statistics value of emergency calls and vehicles' status are recorded and followed by a statistical counting process.

To be consistent with the modeling of fleet deployment problem, the following assumptions are made regarding the real-world operations:

- The distributions of emergency call inter-arrival time and service time are known; more details will be discussed in the case study later.
- The emergency calls can be grouped into several priorities. For each priority the numbers of required vehicle types and numbers are fixed, and the need for hospital treatment is known.
- Three types of vehicles are considered: ambulances, fire engines and police cars.
- The time for each type of vehicle to get charged for reuse is known.
- The emergencies only happen on the nodes.

The framework of the simulation model is illustrated in Figure 6-1. Based on this framework, different emergency response systems can be simulated based on different dispatching strategies, fleet properties and operational time properties.

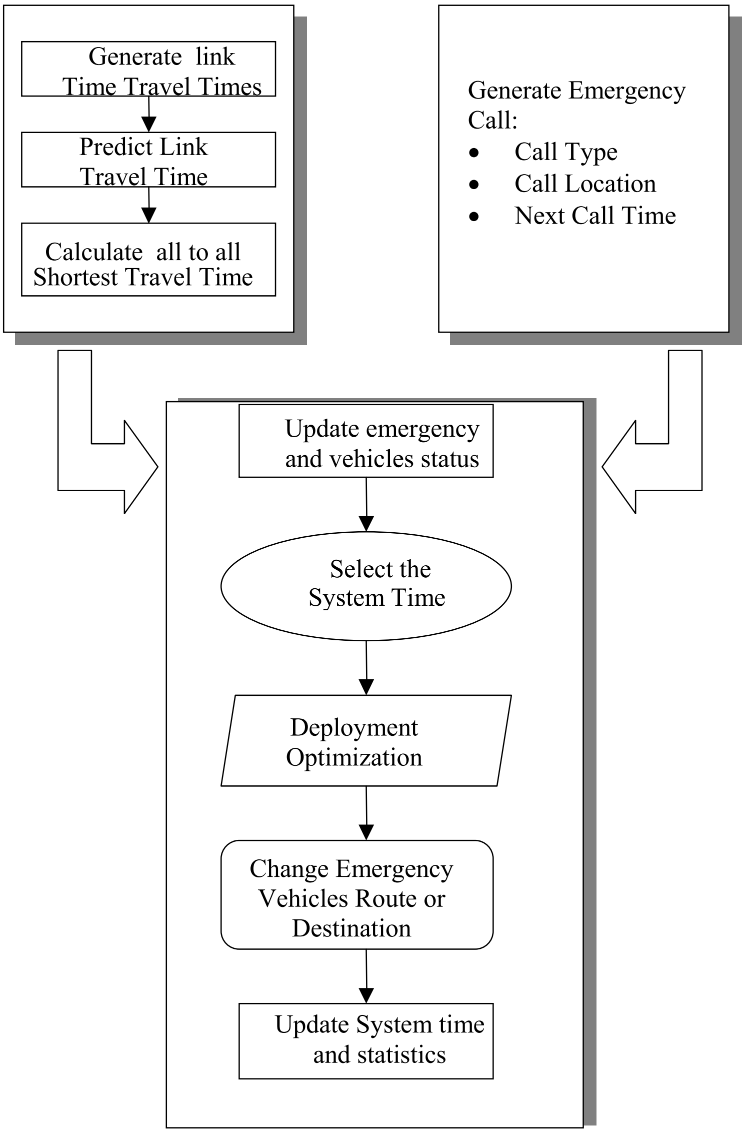


Figure 6-1: Framework of the Simulation Model

6.1.2 Dispatching Strategies

The real-time emergency vehicle deployment problem is concerned with assigning specific response vehicles to emergency calls waiting for service or to candidate relocation sites to provide better coverage for the region, and modifying existing dispatching schemes as changes happen in the system. The following four assignment strategies are tested in the simulation system:

1. First Come, First Service (FCFS)
2. Nearest Origin Assignment and (NO)
3. Flexible Assignment Strategy (FA)
4. Deployment Strategy (DP)

Under the FCFS strategy, we are not concerned with the service coverage. Emergency calls are serviced by the available vehicles in the order in which calls come in. Emergency calls are recorded in a list upon arrival. When a vehicle is available, it can be assigned the first call in the list. If one or more vehicles are available when a emergency call arrives, the vehicle that becomes available earlier will be assigned first. For fire engines, we assume they will go back to the station and get recharged before their next assignment. For police cars, it is assumed that they will be available after the on-site treatment; and for ambulances, if the hospital treatment is required, the ambulance will be available after it gets recharged at the hospital, otherwise, the ambulance has to go back to its station and get recharged before reuse.

The nearest origin strategy differs from the FCFS strategy in its assignment criterion. When a vehicle is available, it will be assigned to the nearest request instead of the one with the earliest arrival time. If an emergency call arrives when one or more vehicles are idle, it will get serviced by the nearest idle vehicle. The condition for the reuse of a vehicle is the same as in the FCFS strategy.

In the FCFS and NO strategies, the important system times are the next call time and next vehicle available time. The system keeps checking the earliest of these times and makes assignments until the end of the simulation process

Different from the NO and the FCFS strategies, the flexible assignment strategy and the deployment strategies allow vehicles on their way to an emergency call to be diverted to another destination if the diversion can bring certain benefits to the system. In the FA strategy, emergency calls enter the pool of unassigned requests when all vehicles are busy. At each simulation time point (discussed in Section 6.1.3), the dispatch center will optimize the current dispatching scheme so as to minimize the total response time according to the associated weights of different types of emergencies. Therefore, rerouting and diversion of vehicles to emergencies are allowed in these strategies. Namely, emergency response vehicles can change their current route or destination under the guidance of the dispatch center. To avoid changing the destination of vehicles so frequently that the drivers get confused and make mistakes, there are minimum required improvements associated with the classes of emergencies and vehicle types that must be satisfied when making a change. In the deployment strategy, besides dispatching vehicles to emergency calls, the relocation concern is taken care of at the same time.

In these two strategies, there are three important system times: the next call time, the next vehicle available time and the next traffic updating time. When the vehicle availability changes or the traffic information is updated, a new optimization model will be generated and solved until the end of the simulation process. More details of the simulation time control scheme are presented in the following section.

6.1.3 Time Control Scheme

In a simulation system, the time control scheme is an extremely important part. In our system, especially, there are “event series” and “time increments” driving the simulation process.

The system will update the travel time on each link after a fixed time increment, e.g. five minutes. The value of this time increment is based upon the current practice of traveler information systems. For most of the current real-time traffic information systems, the traffic information is updated every 5 minutes. This updating may result in the change of vehicle routes and vehicle deployment scheme. On the other hand, when a vehicle changes its status from busy to available, the optimization module will be re-run as well. For example, when a vehicle gets recharged after an on-site treatment and becomes available, the number of available vehicles in the system increases and a new optimization model should be formulated and solved.

To represent the system in a clearer way, we define the updating of travel time information as one type of event so that this simulation model is a pure event-driven simulation system. The time points brought by various events are ranked and the earliest one is selected as the next simulation time point. During the interval of any two simulation points, the system smoothly follows the current operation scheme in memory. This time control module controls the simulation clock to invoke the proper system operation for updating the system status. All of the timing occurrences are produced at the end of different event modules. Table 6-1 summarizes the categories of event timing and matching event. After determining the earliest timing of different occurrences, the event types as well as the status of vehicles are determined.

Table 6-1: Summary of Simulation Timing and Relevant Events

Timing	***Event***	***Entity***
Call arrival	New call arrives	Emergency Call
Call exit	Call leaves system	
Next call time	Generation	
Sit arrival time	Vehicle arrives incident site	Ambulances
Incident on-route time	Vehicle departs to site	
Depot on-route time	Vehicle departs to depot	
Hospital on route time	Vehicle departs to hospital	
Vehicle next available time	Vehicle arrives depot/hospital	
Site arrival time	Vehicle arrives incident site	Police Car
Vehicle on-route time	Vehicle departs to site	
Vehicle next available time	Vehicle departs incident site	
Depot arrival time	Vehicle arrives depot	
Site arrival time	Vehicle arrives incident site	Fire engines
Incident on-route time	Vehicle departs to site	
Depot on-route time	Vehicle departs to depot	
Vehicle next available time	Vehicle arrives depot	
Updating time arrival	Update link travel times	Travel Time Updating

6.1.4 Status Update

The simulation is an event-driven system and the events happen at both fixed time increments and dynamic time increments.

At each simulation point, the program will update the status of emergencies and vehicles. This is because the status of an emergency call and vehicle status are dependent. The information to update for vehicles includes: the current location, the route to take, the destination, the time point of next status change, current status, and next proposed status. Each vehicle in the studied network is treated as a "moving" node. If the position of vehicle has changed, the program will look up the adjacent road network nodes and find the shortest path.

The status of an emergency call is relatively simple. The information to update for an emergency call includes: call location, call type, required number of vehicle at each time, call arrival time and exit time. Some call information is also recorded by the corresponding vehicles.

The events may be caused by emergency calls and emergency vehicles. For emergency call, it can be the "arrival" of a new call; and the arrival of each required vehicle; the completion of on-site treatment. For vehicle, it can be the arrival at the emergency site, the arrival at the relocation site, the arrival at the hospital, etc. The emergencies and vehicles in the system are not independent because vehicles are always responding to emergency calls. For example, when a vehicle *j* changes its status from “driving to an emergency site” to “on-site treatment”, the corresponding emergency call (location of vehicle *j*) will change its status as follows:

1. Decrease the required vehicle number by 1;
2. If the required number of vehicles for the call before is more than 0, the status of the emergency call will not change;
3. If the required number of vehicles is 0, the next status of the emergency call will be “disappearance”.

In this way, the vehicle status and emergency call status are tightly related to each other.

Furthermore, some vehicle status changes may result in the re-optimization of the dispatching or deployment model. For instance, if a police car finished on-site treatment and changed its status to “Idle”, which means the vehicle is available at this point, we may assign it to another emergency call or a relocation site. So the optimization model will be called upon this event.

Table 6-2 lists the current status for emergency calls and the status to which they may change. Table 6-3, 6-4 and 6-5 lists the status change of three types of vehicles.

Table 6-2: Change of Status for Emergency Calls

<table>
<tr><th colspan="2">Current Status</th><th>Next Status to Change</th></tr>
<tr><td colspan="2">Waiting for first vehicle</td><td>In service</td></tr>
<tr><td rowspan="2">In service</td><td>Waiting for other vehicles</td><td>All vehicle arrived</td></tr>
<tr><td>All vehicle arrived</td><td>Disappearance</td></tr>
</table>

Table 6-3: Change of Status for Ambulance

Current Status	Next Status to Change
Idle (0)	Driving to an emergency site (1)
	Driving to an station full (2-a)
Driving to an emergency site (1)	On-Site Treatment (3)
	Driving to station full(2-a)
Driving to station full (2-a)	Idle (0)
	Driving to an emergency site (1)
	Driving to station full (2-a)
Driving to station empty (2-b)	Recharging (5)
On-site treatment (3)	Driving to hospital (4)
	Driving to station empty (2-b)
Driving to hospital (4)	Recharging (5)
Recharging (5)	Idle (0)
	Driving to emergency site (1)
	Driving to station full (2-a)

Table 6-4: Change of Status for Fire Engine

Current Status	Next Status to Change
Idle (0)	Driving to an emergency site (1)
	Driving to an station full (2-a)
Driving to an emergency site (1)	On-Site Treatment (3)
	Driving to station full(2-a)
Driving to station full (2-a)	Idle (0)
	Driving to an emergency site (1)
	Driving to station full (2-a)
Driving to station empty (2-b)	Recharging (4)
On-site treatment (3)	Driving to station empty (2-b)
Recharging (4)	Idle (0)
	Driving to emergency site (1)
	Driving to station full (2-a)

Table 6-5: Change of Status for Police Car

Current Status	Next Status to Change
Idle (0)	Driving to an emergency site (1)
	Driving to an station full (2-a)
Driving to an emergency site (1)	On-Site Treatment (3)
	Driving to station full(2-a)
Driving to station full (2-a)	Idle (0)
	Driving to an emergency site (1)
On-site treatment (3)	Driving to station full (2-a)
	Driving to an emergency site (1)

6.2 Essential Modules

An essential step in developing the simulation model is to generate different modules, such as emergency module and vehicle module. The data structures for accidents and vehicles in the program are both lists that contain the characteristics of each element.

6.2.1 Vehicle Module

Each response vehicle in the fleet represents one vehicle and a working crew which supports the service. Various classes of vehicles, with varying attributes that affect their functionality and ability to respond to particular types of request, could be represented. In this study, three types of vehicles (ambulances, fire engines, and police cars) are considered. At each simulation time point, vehicles are updated with their location, their status, the destination and the path to destination for each vehicle. The nodes representing the vehicles are “temporary” and “movable”. That is, the nodes are attached and move together with the vehicles. At any given time point, each vehicle has an associated status.

A vehicle changes status at the occurrence of certain events that mark the occurrence of an emergency call or the completion of the corresponding activity. For instance, a vehicle status changes from “idle” to “driving to emergency site” upon receiving an assignment from the dispatch center.

6.2.2 Emergency Module

The generation of an emergency call is based on the following information:

1. Spatial distribution of emergency calls;
2. Temporal distribution of emergency calls; and
3. Priority distribution of emergency calls.

Based on the analyses of operational data and historical emergency call records (see discussion in Chapter 7), we find the best fitted distributions for the spatial distribution, temporal distribution and priority distribution.

By summarizing the call types, we categorize the emergency calls into five priorities. As long as the priority of the service call is generated, the required vehicle types, the required number of vehicles in each type, the required on-site treatment time and the upper bound of response time are obtained.

The emergency calls are stored in a call list, and the attributes of each emergency call contains the following information: (a) location of the call, (b) the arrival time of the call, (c) the priority of the call, (d) the required number of vehicle in each type and (d) the maximum remaining time for service calls waiting for response and time to finish for calls in service.

6.2.3 Optimization Module

In our simulation model, when the NO or FCFS strategies are in effect, there is no need to call the optimization module. However, when the Flexible Assignment or Deployment strategies are in effect, after a vehicle changes its status or when the traffic updating is complete, the dispatch center will run the optimization module and makes decision about the movement of all vehicles. Therefore, the optimization module is the kernel of the real-time operation.

As illustrated in Chapter 4, when the FA strategy is applied, the mathematical model, which performs the vehicle dispatching only, can be solved by CPLEX in a reasonable computational time. In this case, the optimization module initializes the ILOG CPLEX environment, creates the problem object, calls the optimizer to solve the problem and retrieves the solution.

When the deployment strategy in applied, the optimization module will perform two major functions. The first function is to determine the problem size. The problem size is decided by the number of candidate relocation sites and the vehicle fleet size. For ambulances and fire engines, since there are limited candidate relocation sites, the deployment model can be solved by CPLEX as well. For police cars, if the size of problem is small enough, we can obtain an optimal solution from CPLEX as well. The optimization module will generate the formulation and will call the solver to solve the mathematical formulation. Otherwise, it will call the heuristics instead of CPLEX. The solution will be retrieved for updating system status. The sizes of problems (|V|, |S|), which can be solved by the CPLEX or heuristic solvers, are shown in Table 6-6.

Table 6-6: Reference of Solver Selection

Problem size (|V|, |S|)		Solver
|V|<=10	|S|<=500	CPLEX solver
	|S|>500	Heuristics
10<|V|<=30	|S|<=200	CPLEX solver
	|S|>200	Heuristics
30<|V|<=50	|S|<=50	CPLEX solver
	|S|>50	Heuristics
|V|>100	|S|<=10	CPLEX solver
	|S|>10	Heuristics

6.2.4 Travel Time Module

At each system time advance point, travel times are needed for the vehicle routing and dispatching. Depending on the different dispatching strategy, the system may need the travel times from the vehicles' current locations to their destinations (emergency sites, hospitals, stations) and from potential relocation sites to all the other nodes. Therefore, under the dispatching strategy without relocation, a group of one-to-one travel times are needed; while when the deployment dispatching strategy is used, we need all-to-all travel times in the network.

Based on the GIS map of the network, the head node and tail node and the distance of each link between adjacent nodes are known. Since no data of real travel times on links are available, we assume two rush periods per day. They are from 7am to 9am and from 4pm to 6pm, respectively.

The average non-peak travel time on a link is calculated by the link length and the designed travel speed of that link. We assume that the travel time in the peak hour is ω_p times the non-peak travel time and there are two peak hours per day. Since multiple vehicle types are considered in the simulation model, for each type of vehicle *k*, we assign a corresponding weight ω_k. It is assumed that the historical average travel time considers all factors that affect travel time, such as incidents, congestion, and signal controls, etc. The variance in travel time caused by these factors is relatively small since emergency vehicles always use siren when they are undertaking a task.

To represent the randomness of travel time of a type *k* vehicle $T_k(t)$ on a link, we use a normal distribution with a mean equal to the average travel time to represent the distribution of travel time on a particular link at time t. The detailed steps of generating travel time $T_k(t)$ are discussed in Appendix I.a.

During the interval $[t, t+\Delta t]$, it is possible that an incident will cause the travel time to change significantly. Therefore, as shown in Figure 6-2, the link can be segmented into very small pieces so that more precise travel time prediction can be provided.

However, it is impossible to divide the link into numerous pieces because of the computational burden. For the simplicity of the simulation model, we predict $\widetilde{T}_k(t)$, the predicted travel time for type *k* vehicles, as the average of $T_k(t)$ and $\mu_k(t+T_k(t))$, where $\mu_k(t+T(t))$ is the average travel time on the same link after a time interval $T_k(t)$. Therefore, $\widetilde{T}_k(t)=(T_k(t)+\mu_k(t+T_k(t)))/2$.

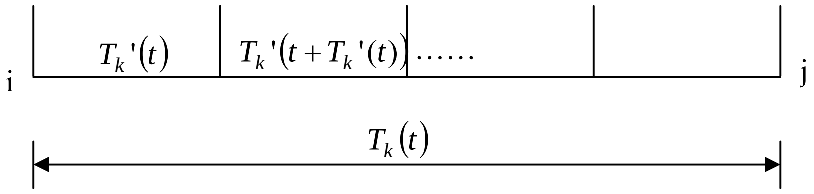


Figure 6-2: Link Travel Time Prediction

Deterministic Shortest Path Algorithm

In this study, we use the Dijkstra Shortest Path Algorithm (Dijkstra, 1959). The advantage of deterministic shortest path algorithm is its speed. In a sample network with 5000 nodes and 7000 links, the all-to-all travel times can be obtained within seconds. The main limitation of Dijkstra algorithm is that is cannot handle negative weight, since there are no negative weights in our network, this algorithm is quite effective.

Note that under certain dispatching strategies, such as the FCFS, NO and Flexible Dispatching strategies, only a group of one to all travel times are required. The all-to-all travel time is only necessary for the Deployment strategy.

Time-Dependent Dynamic Shortest Path Algorithm

To provide online emergency vehicle fleet management, the key part is to utilize real-time traffic information. As mentioned before, it is necessary to calculate all-to-all dynamic optimum shortest paths at each system time advance stamp. Thus an efficient dynamic shortest path algorithm is absolutely essential to the simulation system. In this study, the all-to-one time-dependent shortest path algorithm proposed by Ziliaskopoulos and Mahmassani (1992) is applied to implement the calculation of all-to-all dynamic shortest paths. The implementation of this algorithm is discussed in Appendix I.b.

6.2.5 Statistics Module

This module is important for collecting simulation output. Based on the different operational events, the statistical routine updates the relevant system variables. It provides not only useful information for some applications, but also critical values in checking the simulation model. Based on different events, the designed statistical functions are grouped to analyze the statistical data produced from different modules.

For an emergency call, we record its arrival time and count the number and time of the required vehicles' arrivals and departures. The corresponding response time will be calculated by the *response time* function, and the count of the number of vehicles that arrive later than the required waiting time window will be updated.

For an emergency vehicle, the arrival time at an emergency site or a station is recorded as well as the departure times. For an ambulance especially, the time it arrives at a hospital and the time it finishes recharging and becomes available is recorded.

Some common statistical functions, such as *counter* and *response time* functions, are used to capture the network flow and check the traffic behavior. For instance, due to flow conservation, the number of arrivals should be equal to the number of departures from each call. Functions, *response time*, and *over waited* provide alternatives for measuring network output.

6.3 Simulation Frameworks

With the details discussed above, the flow chart for the developed simulation program is shown in Figures 6-3 through 6-6. Figure 6-3 shows the simulation flow chart for the FCFS dispatching policy; Figure 6-4 illustrates the simulation flow chart for the Nearest Origin dispatching policy; Figures 6-5 and 6-6 indicate the simulation flow chart for the Flexible Dispatching and Deployment strategies.

The flow charts of the FCFS and the NO are similar to each other and the difference lies in the selection of each single dispatched vehicle. Under the FCFS, the vehicle which has the longest idle time will be selected, while under the NO, the vehicle that is the closest to the emergency site will be selected.

Under the Flexible Assignment and the Deployment dispatching strategies, there are three important time stamps: the time a new call comes in, the traffic updating time and the vehicle status updating time. In the simulation model, when each of these time stamps is reached, a series of operations will be performed and the corresponding optimizer module will be called.

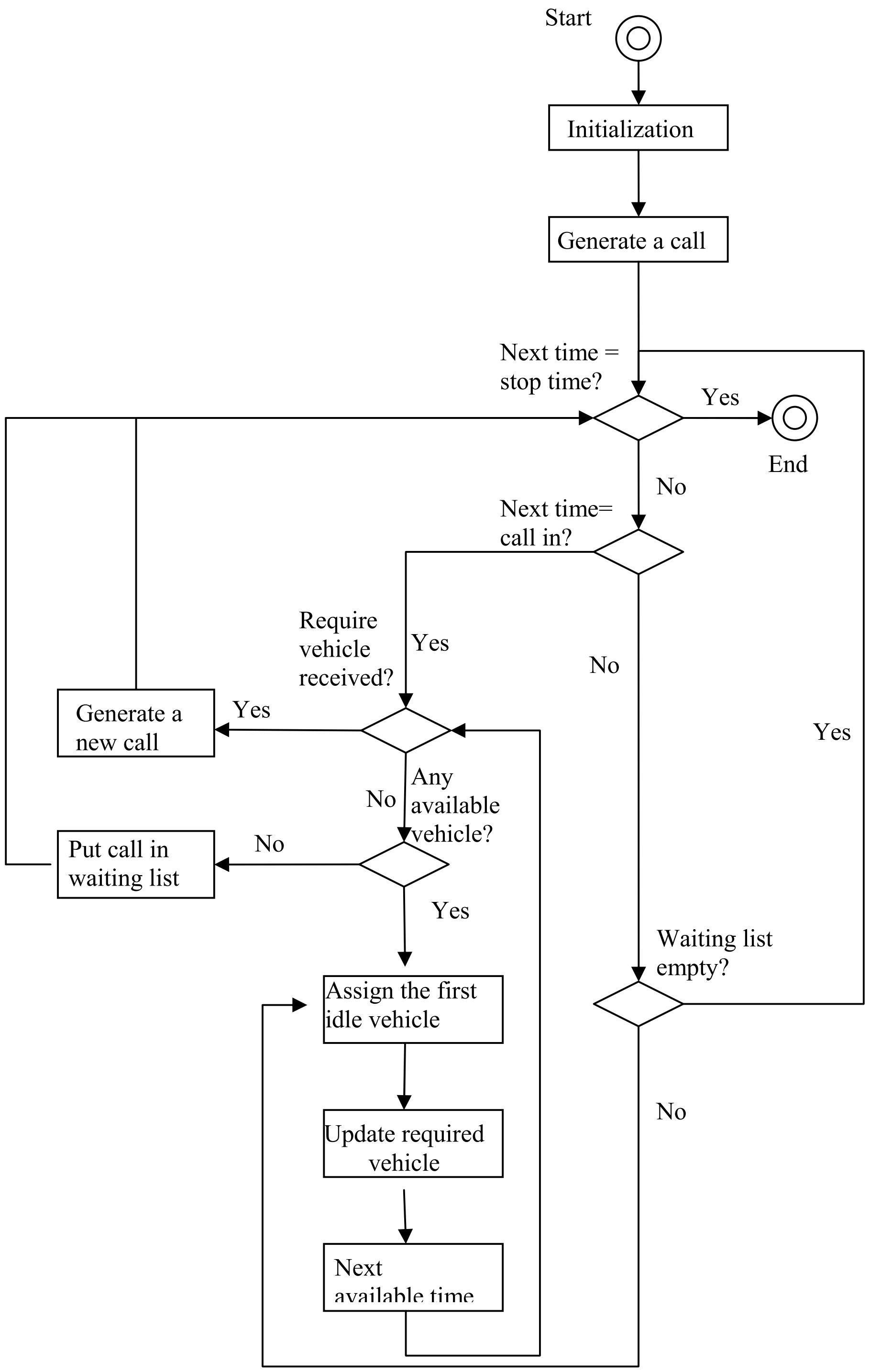


Figure 6-3: Simulation Flow Chart of FCFS

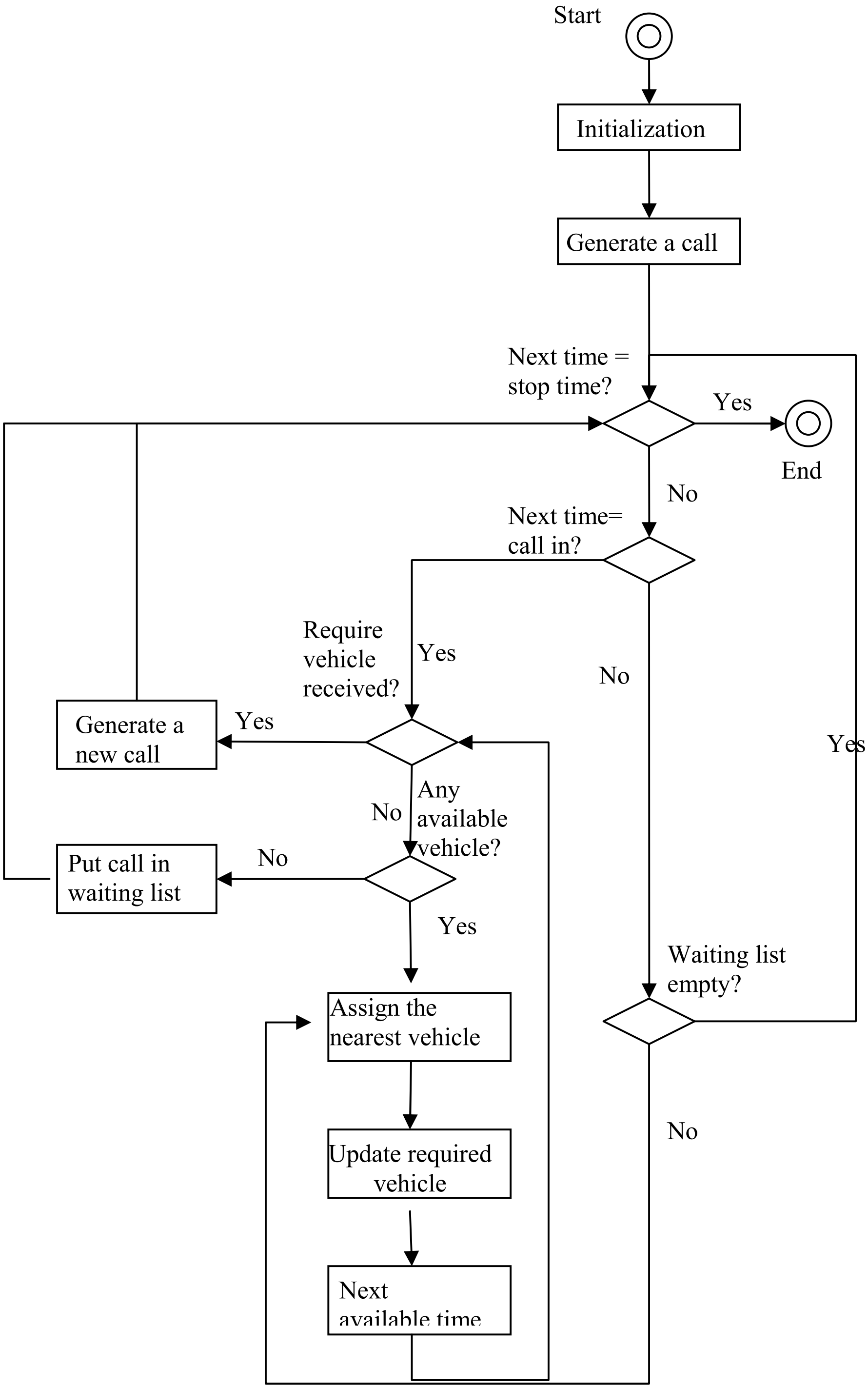


Figure 6-4: Simulation Flow Chart of Nearest Origin

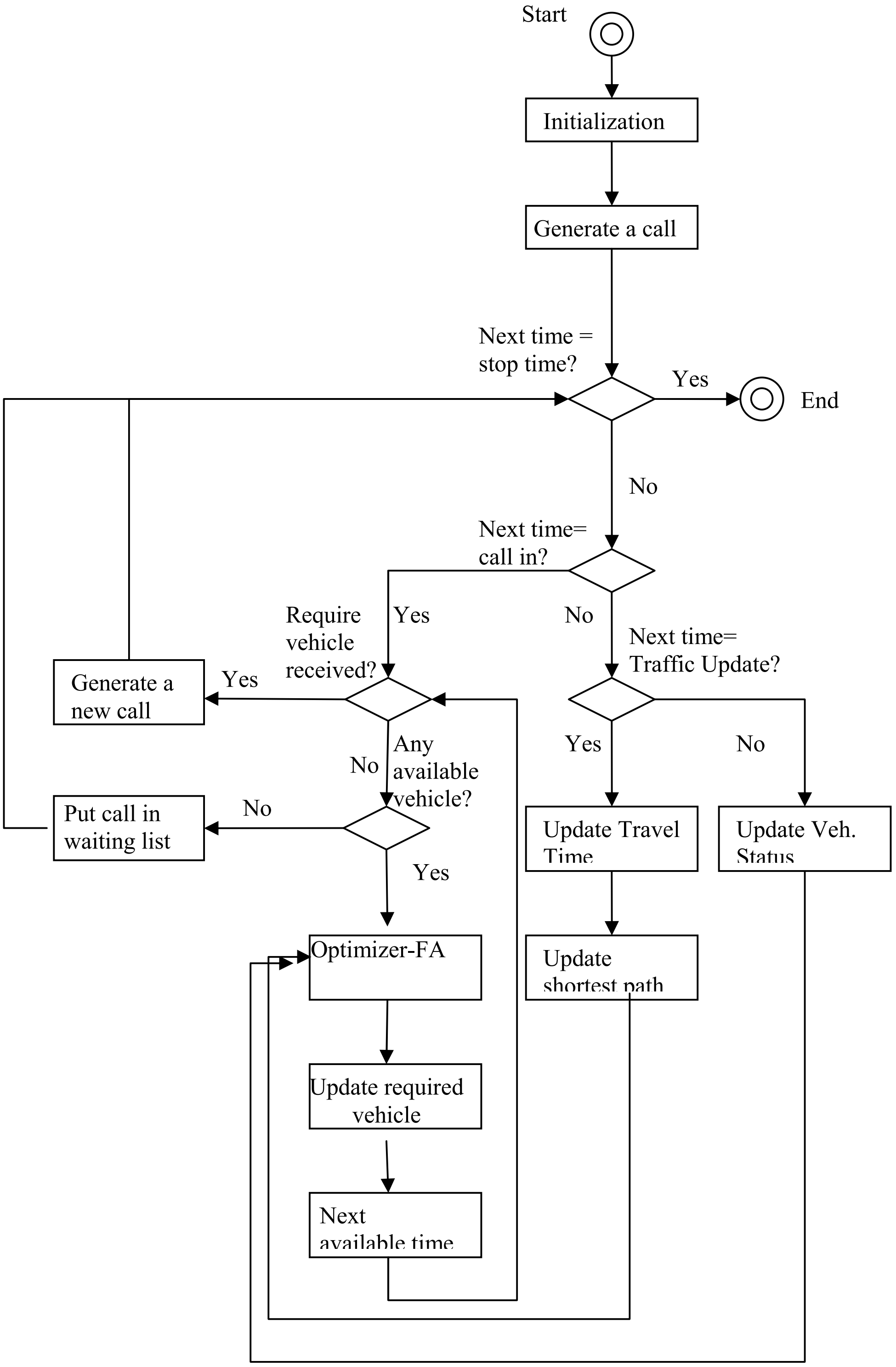


Figure 6-5: Simulation Flow Chart of Flexible Assignment

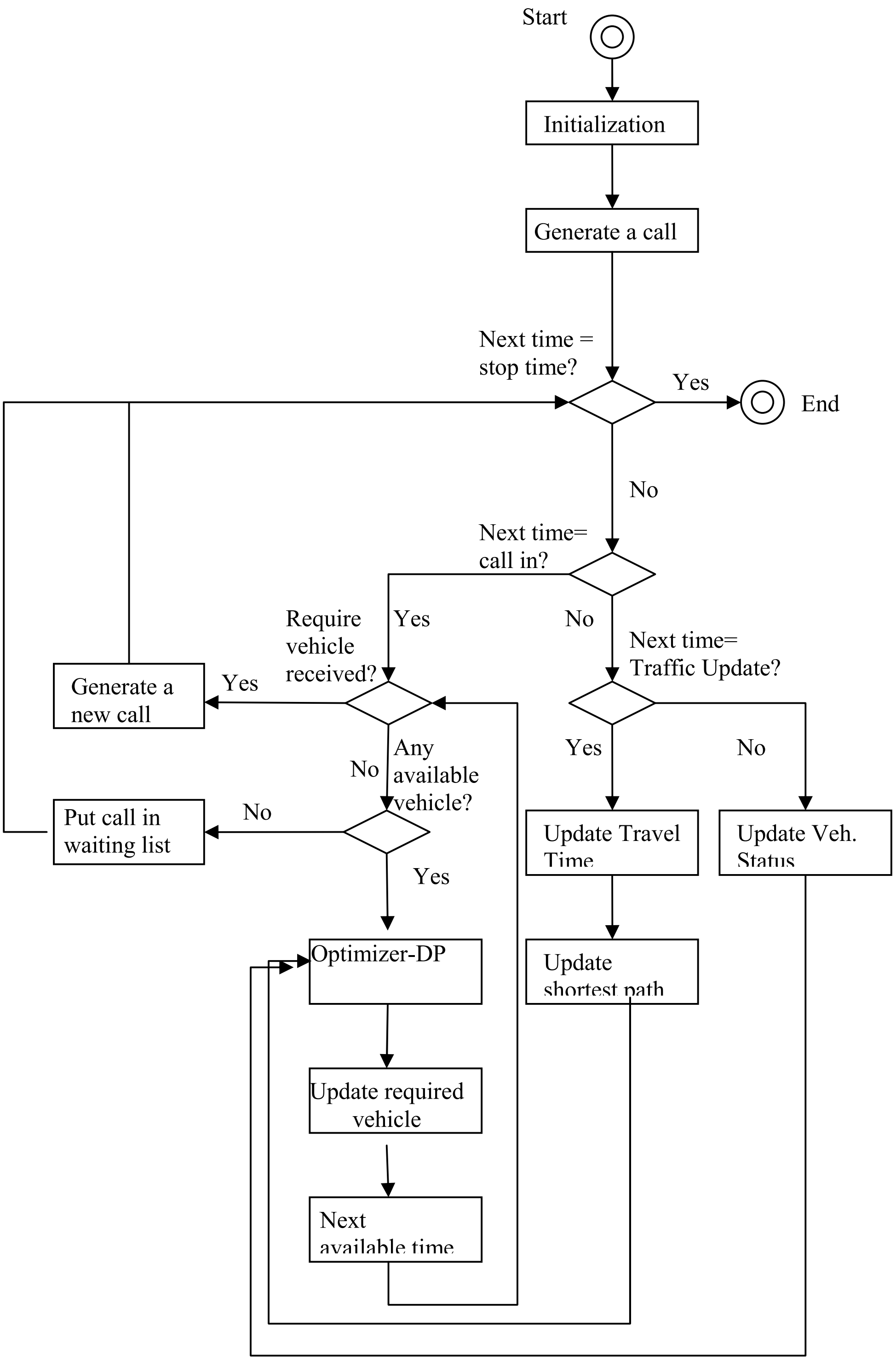


Figure 6-6: Simulation Flow Chart of Deployment Only

6.4 Input Analysis

A simulation model needs some important inputs. The system developed here can be deemed as a G/G/C/∞/∞ system. Choosing “correct” input distributions can affect the accuracy of a simulation model’s output when validating that model with real-world data. Thus the proper probability models and a reliable random number generator are important in conducting a simulation. In addition to the random feature of inter-arrival times in queuing systems, this simulation system introduces other sources of randomness such as emergency call type, spatial distribution and temporal distribution of emergency calls, and travel times of links. Some general distributions can be borrowed from established probability models, but some system randomness can be represented by empirical distributions, i.e., real-world operational data from emergency management centers. In this section we summarize the techniques used in generating the input, while the detailed calibration will be discussed in Chapter 7.

6.4.1 Selection of Probability Models

One of the most important activities in a successful simulation study is that of representing each source of system randomness by a probability distribution. If this critical activity is neglected, then simulation results are quite likely to be erroneous and any conclusions drawn from the simulation study will be suspect.

The important inputs of the system include:

1. Spatial Distribution of Emergency Calls;

2. Temporal Distribution of Emergency Calls;
3. Priority Distribution of Emergency Calls;
4. Service Time of Emergency Calls;

Since historical data are available, the most widely used technique is the density/histogram overplot. For a group of observations $X_1, X_{2,} ..., X_n$, an empirical distribution function $F_n(x)$ can be defined as follow:

$$F_n(x) = \frac{number\ of\ X_i's \leq x}{n} \qquad (6\text{-}1)$$

Where, $F_n(x)$ is the proportion of the observations that are less than or equal to x.

When fitting an empirical distribution function to a known distribution function, if the fitted distribution is a perfect fit and the sample size is very large, the plot of difference between the known distribution function $\hat{F}(x)$ and $F_n(x)$ will be a horizontal line with a height at 0. The greater is the vertical deviations from this line, the worse is the quality of fit.

Other than the eyeballing for difference or similarity which is somewhat inexact, there are several specific goodness-of-fit tests which can be use to test the following null hypothesis:

H$_0$: The X_i's are IID random variables with distribution function $\hat{F}$

The most widely used tests are Chi-square Tests and Kolmogorov-Smirnov (K-S) Tests. To compute the chi-square test statistic, we need to divide the entire range of the fitted distribution into m adjacent intervals. Denote N_j as the number of X_i's in the j^{th} interval, and p_j as the expected proportion of the X_i's that should fall in the j^{th} interval if sampling from the fitted distribution. Then, the test statistic is

$$\chi^2 = \sum_{j=1}^{m} \frac{(N_j - np_j)^2}{np_j} \qquad (6\text{-}2)$$

We reject H_0 if χ^2 is too large.

The K-S test statistic D_n is simply the largest distance between $\hat{F}(x)$ and $F_n(x)$ for all values of x and can be formally defined by

$$D_n = \sup_x \{ | \hat{F}(x) - F_n(x) | \} \qquad (6\text{-}3)$$

A large value of D_n indicates a poor fit. If D_n exceeds some constant $d_{n,1-\alpha}$, where α is the specified level of the test, we need to reject the null hypothesis H_0. The numerical value of the critical point $d_{n,1-\alpha}$ depends on how the hypothesized distribution was specified.

There are many statistical packages that have a function to select best-fitted well-known distributions for empirical data and these test statistics can be reported. These packages greatly reduce the effort in input analysis.

It is important to note that failure to reject H_0 should not be interpreted as "accepting H_0 as being true." Instead, they should be regarded as a systematic approach for detecting fairly gross differences. And if *n* is very large, then these tests will almost always reject H_0 (Gibbons, 1985). Since H_0 is virtually never exactly true, even a small departure from the hypothesized distribution will be detected for large *n*. Fortunately, it is usually sufficient to have a distribution that is "nearly" correct.

The analysis of historical emergency call records revealed that the spatial distribution of emergency calls can be represented as a uniform distribution. The analysis of real operational data describing the arrival of the emergency calls to the dispatch center revealed that the inter-arrival time of service calls can be described by an exponential distribution. Therefore, an exponential distribution with rate λ is used to fit the service call arrival rate. By summarizing the call types, we categorize the emergency calls into five priorities. The probability of a call falling into each category is directly calibrated from the historical data. As long as the priority of the service call is generated, the required vehicle types, the required number of vehicles in each type, the required on-site treatment time and the upper bound for response time are obtained.

Since a group of "random" numbers need to be generated in each simulation step, the generation of the random variates from these distributions becomes crucial.

6.4.2 Random Number Generator

The goal of choosing a "good" arithmetic random number generator is to avoid any possible correlation in a generated stream of random numbers; otherwise, the simulation's results may be completely invalid. Park and Miller (1988) provide a random number generator applicable to a wide variety of systems. It fulfills the minimum standard and can be conveniently implemented in a high-level language. The details of generating random numbers is discussed in Appendix I.c.

6.5 Output Analysis

To develop a simulation system to test various dispatching strategies and facility location/allocation plan, plenty of time and energy are spent on the conceptual model development, coding and system calibration. Actually, to get precise estimates of the system performance measures, it is important to appropriately analyze the simulation output. One simulation run is a computer-based statistical sampling experiment; each run only produces a realization of a set of random variables, which may be far from the true system characteristics. To ensure an appropriate statistical analysis from simulation results, a number of simulation replications are necessary.

In addition, when a simulation run starts at time 0, it goes through a transient period, and eventually achieves a steady state with steady demand if the system capacity is not exceeded. Because the output process from the steady-state distribution is considered, it is necessary to discard a specific transient time, which is named as the warm-up period, in which the state of system is not yet stable. The convergence rate depends on the initial condition. For instance, a larger network needs a longer time to achieve system stability. Networks with higher flow rates also require longer transition periods. Therefore, the number of replications needed and the start time for data collection are important decisions.

A simulation model with the nearest origin strategy, on a network with 1757 nodes, 10 depots and 10 vehicles that is shown in Figure 6-7, is used to demonstrate the importance of the number of replications and the initialization time.

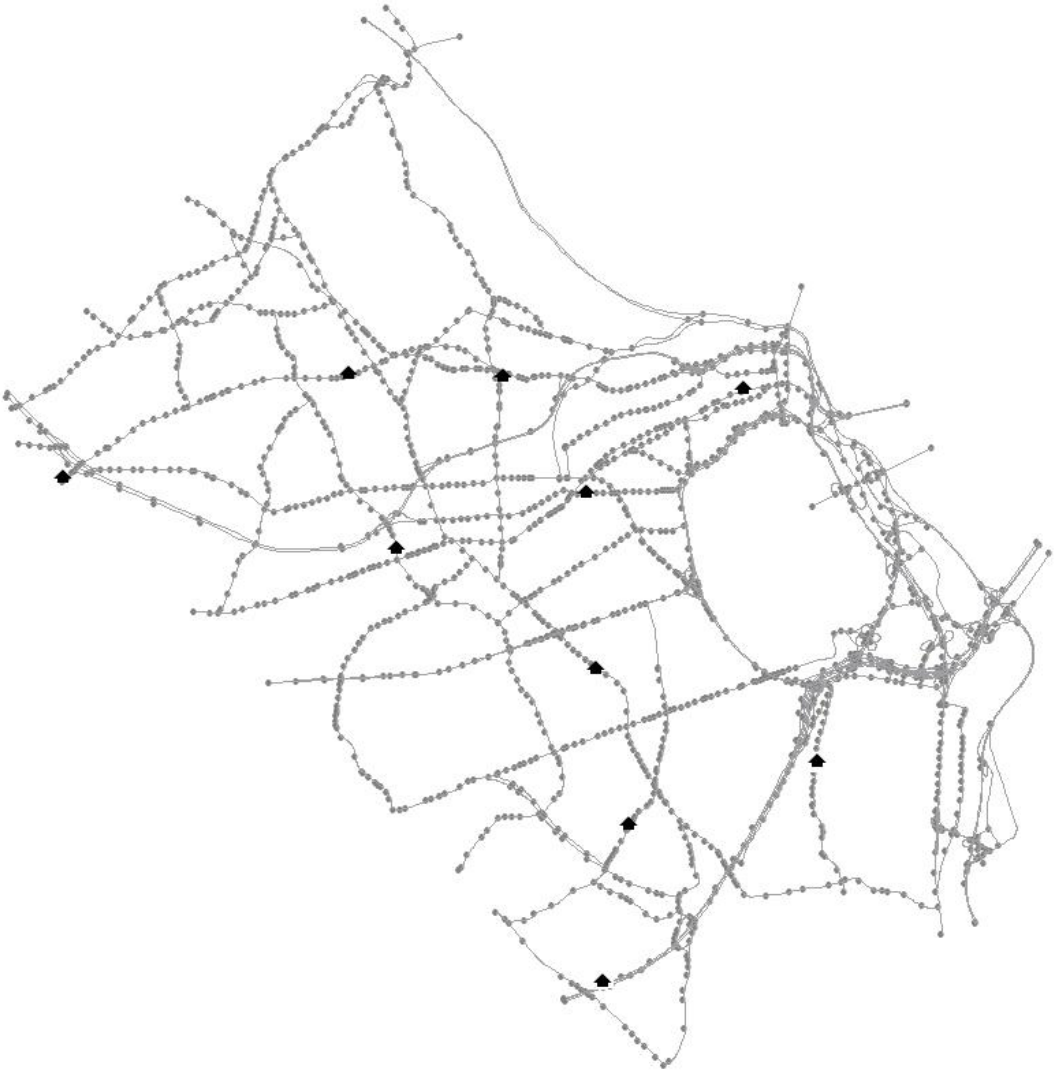


Figure 6-7: Test Example for Output Analysis.

Table 6-7 shows the typical simulation output. The simulation terminates whenever the time period reaches the pre-specified value. In 10 independent replications, with the same initial condition but different random seeds, there is considerable variance in average response time, dispatched vehicles and emergency calls. Clearly, one single simulation run cannot produce reliable estimates. As long as system reaches the steady state, it is likely to estimate statistical results from the mean of all the replications.

Table 6-7: Results for 10 Independent Replications

Replication	Total Number of Calls	Total Response Time (min)	Total Number of Vehicles	Average Response Time (min)
1	464	3295.751	851	3.873
2	428	2986.186	770	3.878
3	446	3142.61	804	3.909
4	388	2601.509	698	3.727
5	460	3222.778	842	3.828
6	402	2679.412	702	3.817
7	456	2998.432	809	3.706
8	442	3139.087	807	3.89
9	446	3085.884	788	3.916
10	455	3219.925	838	3.842

6.5.1 Length of Warm-up Period

In analyzing the output from a single simulation run, it is necessary to determine the length of the warm-up period *l* to avoid collecting outputs before a steady state is reached. The simplest and most general technique for determining *l* is a graphical procedure by Welch (1981, 1983). Its specified goal is to determine the time where the state of the system approaches a stable condition and the relevant estimates tend toward a steady-state mean.

Let Y_1, Y_2... be an output stochastic process from a single run of a simulation. Denote the steady-state mean $\mu = E(Y)$, which is generally defined by $\mu = \lim_{i\to\infty} E(Y_i)$, to determine when the transient mean curve $E(Y_i)$ flattens out at the level of μ, a time index l such that $E(Y_i) \approx \mu$ for $i > l$, where l is the warm-up period, should be identified. Usually, there exist data fluctuations in simulation throughputs. Therefore, the moving average process is applied to smooth out the high-frequency oscillations. With a series of throughputs Y_i (i= 1,2,…,m), a transformed moving average $Y_i(w)$ is defined as:

$$Y_i(w) = \begin{cases} \dfrac{\sum_{s=-w}^{w} Y_{i+s}}{2w+1} & if\ i = w+1,...,m - w \\ \dfrac{\sum_{s=-(i-1)}^{i-1} Y_{i+s}}{2i-1} & if\ i = 1,...,w \end{cases} \tag{6-4}$$

where w is the *window* and is a positive integer such that $w \le \lfloor m/2 \rfloor$ and the lower part of the transformation satisfies the boundary condition. By graphically plotting the Y_i and $Y_i(w)$, the time index l can be easily defined.

Also, from the previous test example with the NO dispatching strategy, 10 ambulances and 10 depots, consider the stochastic process D_1, D_2,…, where D_i is the selected simulation throughput per period. Figure 6-8 shows 300 original simulation throughputs of response times. Figures 6-9 indicates the averaged process for fluctuating outputs of average response time. According to the example, the warm-up period should be less than one day in this case.

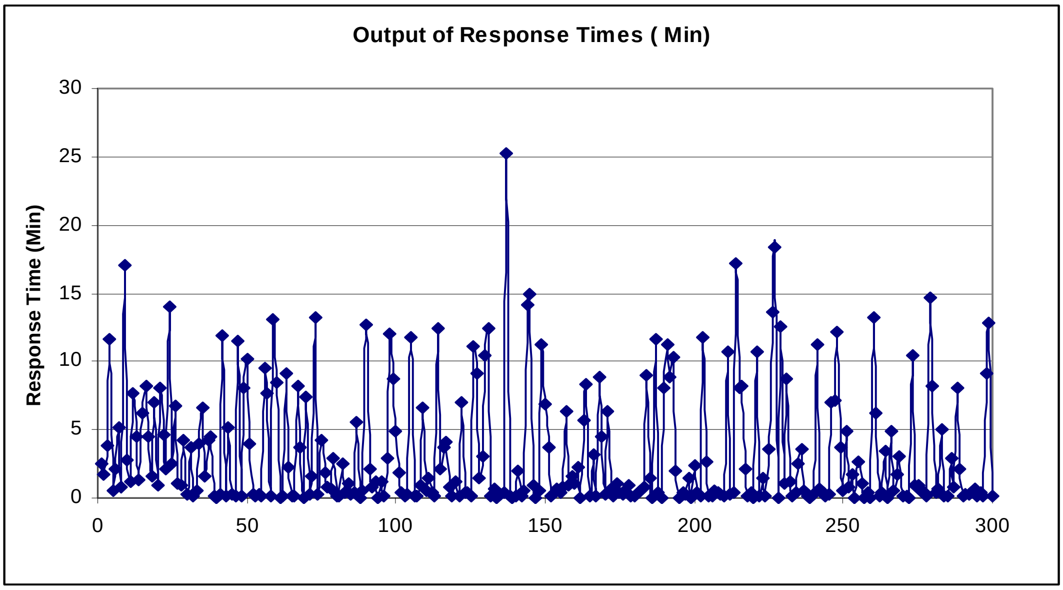


Figure 6-8: An Example of Simulation Throughput.

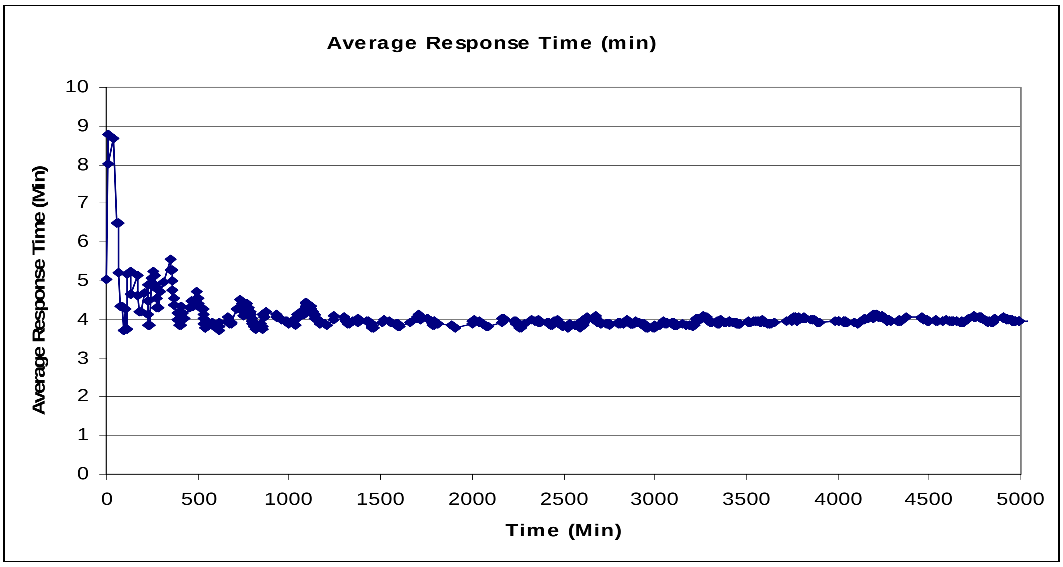


Figure 6-9: Averaged Process for Simulation Throughput

6.5.2 Independent Replications

To get an average system performance measure, e.g. average response time, a number of simulation replications provides an experimental sample set. The larger the sample set, the more precise is the estimate of parameters. The number of replications needed depends on the specified precision, degree of confidence and sample variance.

With a confidence interval (1-α) percent, for an average of performance measure *μ*, with fixed number of replications *n*, and assume that the estimate $S^2(n)$ of the population variance does not change as the number of replications increases. In order to have an estimate within an error β with *(1-α)* percent confidence, which is defined as $\left|\overline{X} - \mu\right| = \beta$, an approximate expression for the total number of replications, $n_a^*(\beta)$, is given by (Law and Kelton, 1991);

$$n_a^*(\beta) = \min\left\{ i \le n : t_{i-1,1-\alpha/2}\sqrt{\frac{S^2(n)}{i}} \le \beta \right\} \qquad (6\text{-}12)$$

Indeed, $n_a^*(\beta)$ can be approximated as the smallest integer *i* satisfying $i \ge S^2(n)(Z_{1-\alpha/2}/\beta)^2$. If $n_a^*(\beta) > n$, then $[n_a^*(\beta) - n]$ additional replications of the simulation are required.

In this simulation system, we use batch mean in the analysis. For example, the duration of each simulation is set to be 100 days, with each 10 days as an interval. With the consideration of warm-up period, the actual simulation duration is 101 days but the statistical collection starts from the second day. The statistical counter is reset at the beginning of each replication. To reach maximum independence among the batch means, each interval uses its own set of seeds which are 5,000,000 steps apart from the seed sets of the adjacent intervals. It is important to assure that the separation between adjacent seeds is far enough to avoid potential correlation brought by the overlap, in generated random sequences. Figure 6-10 shows the example of the batched mean and the corresponding seed generation.

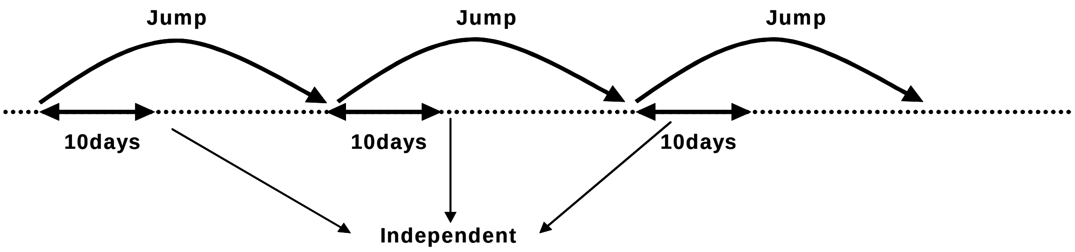


Figure 6-10: An Example of Batch Mean.

From the previous test example, the mean of average response time $\overline{X}$ is 3.839 and the variance $S^2(10)$ from 10 available replications is 0.072. Suppose the specified precision β is 30seconds and the confidence level *(1-α)* of is 95 percent. Then, the approximate number of replications $n_a^*(\beta)$ would be (0.072)·(2.262/0.5)^2 = 1.478. Therefore, the minimum number of required replications is 2 and 10 replications is good enough for the assumed confidence level and precision. We do not need additional replications.

When there are multiple measures of performance, μ_s (where s=1, 2, …*k*), and I_s is a $100(1-\alpha_s)$ percent confidence interval for the measure of performance μ_s (where s=1, 2, …*k*). In order to satisfy all *k* confidence intervals $\alpha_1, \alpha_2,\ldots, \alpha_k$ simultaneously, the overall confidence level α should be associated with $\sum_{i=1}^{k}\alpha_i = \alpha$. Since P(all *k* confidence intervals are satisfied)> $1-\sum_{i=1}^{k}\alpha_i$, α_i do not have to be equal and any α_i corresponding to more important measures could be smaller. If we have 5 performance measures and we prefer the overall confidence level to be $\alpha = 0.05$ and for each measure the confidence level is 100(1-0.01) percent, the number of replications needed is $Max(S^2(n_i)\cdot(3.25/\beta_i)^2)$, where β_i is the precision related to the i^{th} performance measure.

6.5 Conclusions

In this chapter, we described the system framework, major modules in the system and the four types of dispatching strategies and corresponding flowcharts. The simulation system is driven by fixed and non-fixed time increment caused by various types of events. These events link the status changes of vehicles and emergency calls. The details of input and output analysis were discussed as well. In the next chapter, we will test our simulation system in a real-world size case study on the street network of one of the counties in the Washington metropolitan area. The system will be calibrated using the real-world operational data as well. We will also illustrate the application of this simulation system in a long term facility location planning problem.

Chapter 7: Case Study

7.1 Background

In this case study, we first calibrate the simulation model with a real street network map and with the real operation data of one of the counties in the Washington, DC metropolitan area.

We will apply four dispatching strategies is the simulation model, which are the First Come First Service, the Nearest Origin, the Flexible Assignment strategy (*M1*) and the Deployment Strategy (*M3*). We performed sensitivity analysis for different inter-arrival times, different weight parameters, different depot locations and different fleet size and the results will be discussed in this chapter. We also applied this model to a Facility Location/Allocation Planning problem to demonstrate the potential application of the proposed system for long term planning purposes.

7.2 Current Operations

7.2.1 Street Network

A network consisting of 5496 nodes and 7325 directed links that is modeled from an existing network is used in calibration. This street network is shown in Figure 7-1. The histogram of lengths of links is illustrated in Table 7-1. Since more than 98% of the links have lengths smaller than 500 meters, it is reasonable to assume all emergency calls can be aggregated at nodes.

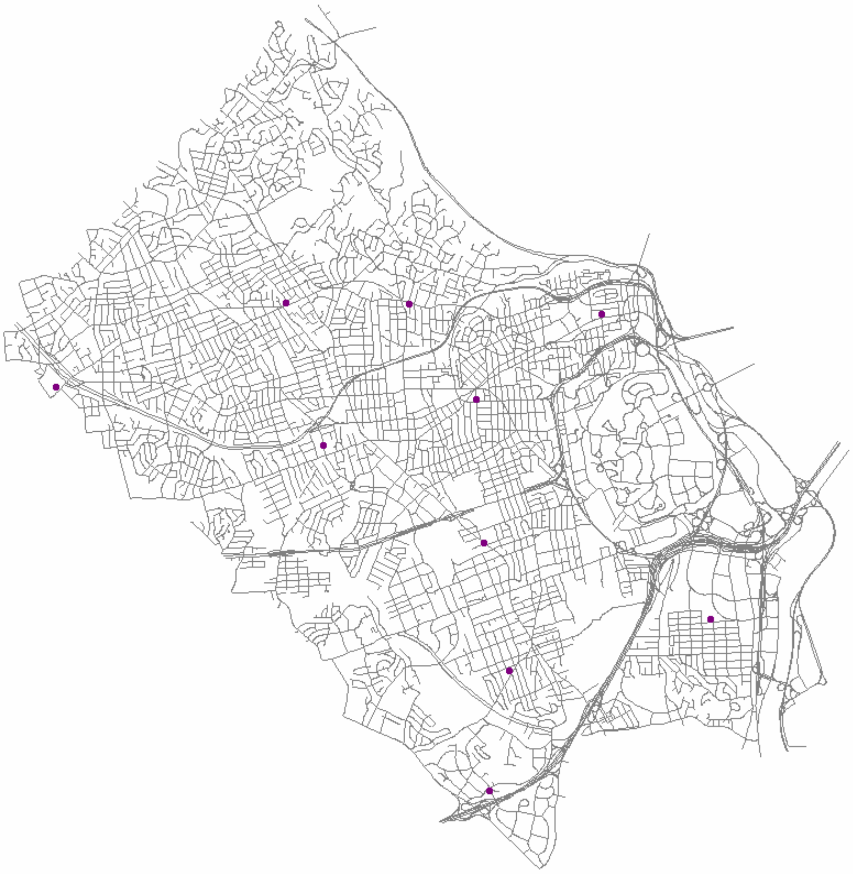


Figure 7-1: A Real Street Network

Table 7-1: Histogram of Link Lengths

Length (m)	Frequency	Cumulative %
[0, 100)	3941	53.81%
[100, 200)	2586	89.12%
[200, 300)	526	96.30%
[300, 400)	110	97.80%
[400, 500)	58	98.59%
[500, 600)	33	99.04%
[600, 700)	26	99.40%
[700, 800)	7	99.49%
[800, 900)	6	99.58%
[900, 1000)	9	99.70%
>=1000	22	100.00%

7.2.2 Operational Data

The data used in the calibration are generated from the real-world operations for the ambulances and the medical units during November and December of 2000. The data include 3029 records. Each record stands for one dispatched vehicle and has 31 variables which describe various information associated with the call including the time at which the emergency call arrived, vehicle identification number, dispatching time, arrival time and call type.

Emergency Vehicles

16 units (vehicle ids) are listed in the data, which is the total number of the EMS (ambulance) fleet. The dispatched emergency vehicle types are recorded in Table 7-2.

EMS vehicles were dispatched for Life Support for more than 87% of the cases. Table 7-2 also shows the detailed utilization of each vehicle in all types of emergencies. We can easily find that the workloads of the vehicles are uneven. For example, unit M101 was dispatched 304 times, while unit M401was dispatched only 16 times in total.

Since the operational data for police cars and fire engines were not unavailable, we assumed the fleet size for each type of vehicle. The relevant vehicle characteristics are estimated from EMS characteristics. For example, the average link travel time of an EMS unit can be 1.2-1.3 times of the travel time of a police car and 0.8-0.9 times of that of a fire engine according to the estimations of experienced staff in emergency management center.

Table 7-2: Vehicle Dispatching by Emergency Type

Unit	EMS-ALS	EMS-BLS	Fire	PS	Total
A101	20	43	N/A	N/A	63
A410	1	39	N/A	N/A	40
A428	N/A	14	N/A	N/A	14
M101	103	183	16	2	304
M102	166	230	38	2	436
M104	196	322	42	2	562
M105	140	243	36	N/A	419
M106	168	211	32	1	412
M109	170	262	49	4	485
M202	N/A	2	N/A	N/A	2
M206	2	N/A	N/A	N/A	2
M325	31	52	8	N/A	91
M401	6	9	1	N/A	16
M408	2	1	N/A	N/A	3
M410	57	73	8	N/A	138
M418	18	22	2	N/A	42
Total	1080	1706	232	11	3029

Dispatched Number of Vehicles

Among the 3029 vehicle dispatching records, there are 2647 emergency calls listed. Therefore, more than one vehicle can be dispatched to an emergency call. We categorize the emergencies into four groups based on the dispatched number of vehicles per call. Table 7-3 shows the summary of the number of dispatched vehicles.

Table 7-3: Summary of Number of Dispatched Vehicles

Number of Disp. Vehs	1	2	3	>=4	Grand Total
Subtotal	2310	299	32	6	2647
Percentage	0.873	0.113	0.012	0.002	1

Emergency Inter-arrival Time

There are 3029 dispatching time records in the file. However, since one emergency may need more than one vehicle, we use the difference between any two consecutive calls' times as the emergency inter-arrival time. 2647 emergency calls are recorded in the database. Since the inter-arrival time of the first incoming call is unknown, 2646 valid inter-arrival times are obtained.

We analyzed the input data with Arena Input Analyzer (a statistical tool available in Arena simulation software). Arena Input Analyzer is used to determine the quality of fit of probability distribution functions to the input data. It fits the original data to a set of 12 distributions. And the number of histograms for data fitting is automatically optimized. The distributions are sorted, from best to worst, according to the respective errors terms. The top five best fitted distribution are shown in Table 7-4. The result indicates that the best fitted distribution is an Exponential Distribution $Exp(0.548)$, with a squared error equal to 0.00324. Figure 7-2 shows the histograms of the real data and the fitted data.

Table 7-4: Distribution Estimation Results of Emergency Inter-arrival Time

Fitted Function	Squared Error
Exponential	0.00324
Lognormal	0.00395
Beta	0.006
Normal	0.109
Triangular	0.196

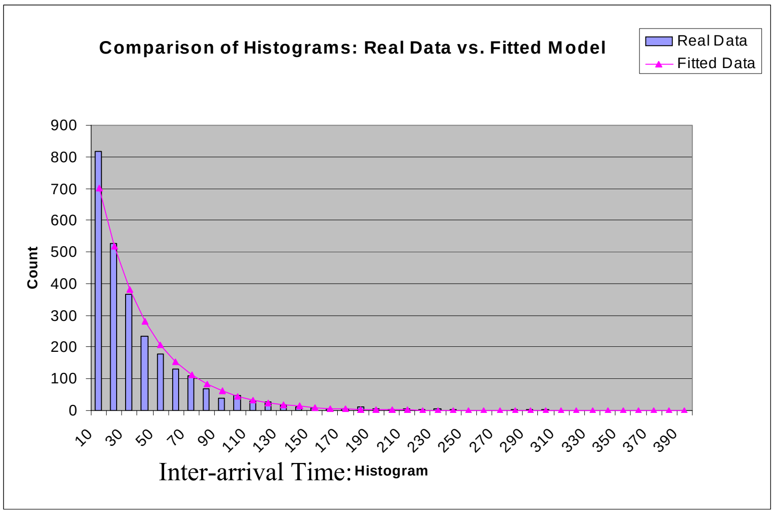


Figure 7-2: Comparison of Real Service Time and Fitted Model: Inter-arrival Time

Vehicle Travel Time

In 3029 records, 2436 records have the travel time from the station to the emergency site. According to test results shown in Table 7-5, the best distribution is Lognormal Distribution $LOGN(1.65, 0.64)$ with a squared error equal to 0.0152. The comparison of histograms of the real data and the fitted data is shown in Figure 7-3.

Table 7-5: Distribution Estimation Results of Vehicle travel time

Distribution	Squared Error
Lognormal	0.0152
Gamma	0.0341
Erlang	0.0401
Beta	0.0602
Normal	0.0981

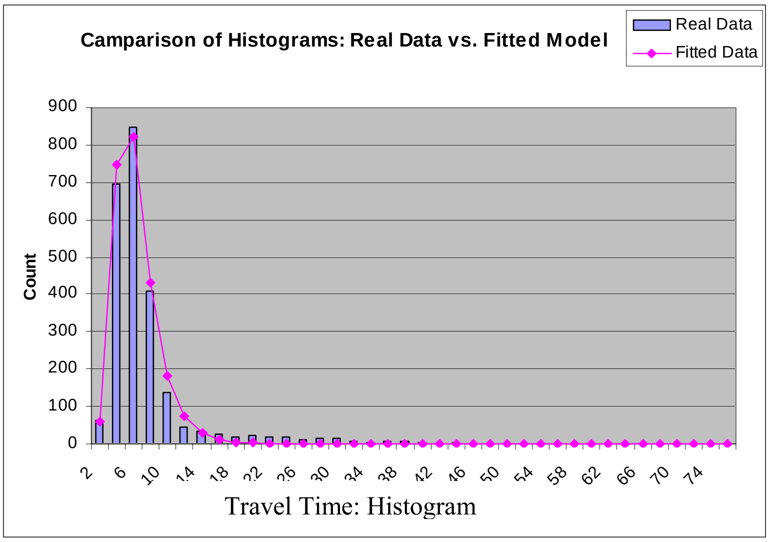


Figure 7-3: Comparison of Real Service Time and Fitted Model: Travel Time

Service Time

Since there is no on-site service time available, we use the available variables in the database to calculate the service times. The service times are calculated as the difference between the time the vehicle arrived at the site to the time the same vehicle departed from the site. Another issue is the “fake” emergency calls. In real operations, many emergencies are not as described in the calls and the emergency vehicle will depart at once or soon after arriving at the emergency site.

The histogram of service time shows two peaks (as shown in Figure 7-4). It can be represented as a combination of different distributions. It is assumed that:

1) If the call does not need further treatment, the service time follows lognormal distribution.
2) If the call does need further treatment, the service time will follow normal distribution.
3) Five type of service times are considered in the system, with the probability of each type P_i, we have:

$$\begin{array}{ll}
\textit{Type I} : \textit{LOGN}\ (2.7, 0.7) & P_0 = 0.2 \\
\textit{Type II} : N(16, 7) & P_1 = 0.13 \\
\textit{Type III} : N(57,\ 14) & P_2 = 0.56 \\
\textit{Type IV} : N(85,\ 15) & P_3 = 0.09 \\
\textit{Type V} : N(120,\ 40) & P_4 = 0.02
\end{array}$$

Based on the analysis above, we obtained the following model:

$$f(x) = P_0 * \frac{1}{\sqrt{2\pi\sigma_1^{\ 2}}} e^{-(\ln x-\mu_1)\ /(2\sigma_1^{\ 2})} + \sum_{i=1}^{4} P_i \frac{1}{x\sqrt{2\pi\sigma_i^{\ 2}}} e^{-(x-\mu_i)\ /(2\sigma_i^{\ 2})} \qquad (7\text{-}1)$$

Type I is the “fake” calls whose service time is a lognormal distribution with a mean of 2.7 minutes and a standard deviation of 0.7 minutes. Type II to V represent the emergency calls which need on-site service. The means of these types of calls vary from 16 minutes to 120 minutes and the standard deviations vary from 7 minutes to

40 minutes. The probabilities of five types of calls are tested by a computer program to achieve the smallest error.

Figure 7-4 shows the histogram of real data (all 5 types of calls) and the derived data from the fitted model; the fitted data match the real data very well.

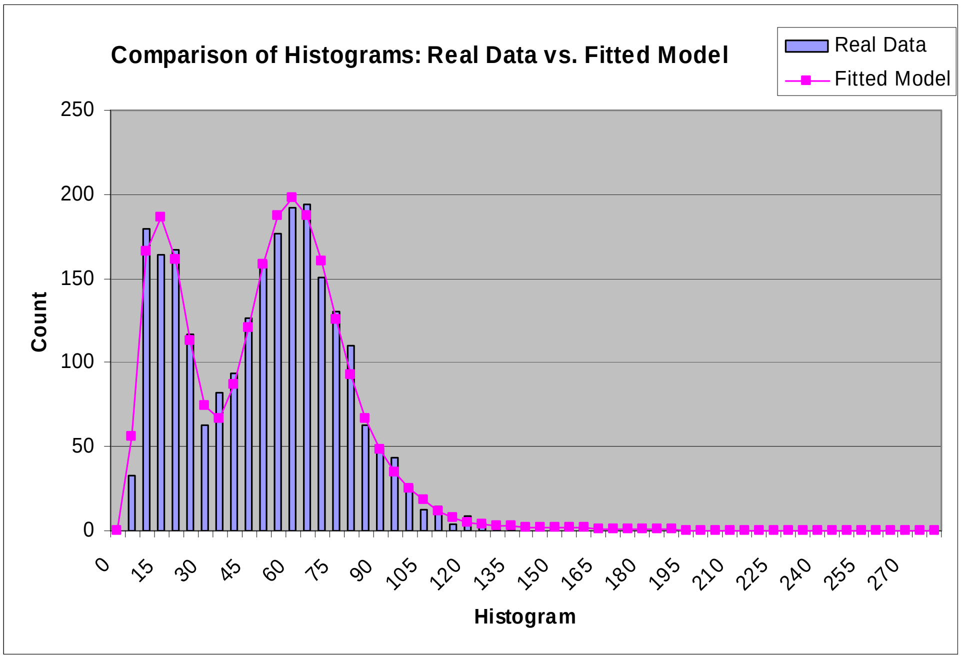


Figure 7-4: Comparison of Real Service Time and Fitted Model: Service Time

7.3 Case Study

7.3.1 Comparison of Dispatching Strategies

It is too expensive to test alternative dispatching strategies in real operations for comparison purposes. However, a precise simulator is able to provide persuading results for those promising new strategies without risking real property and lives. The dispatching strategies compared in this test include:

1. First Called, First Served (FCFS);
2. Nearest Origin Assignment (NO);
3. Flexible Assignment Strategy (FA); and
4. Deployment Strategy (DP).

The FCFS strategy assumes the service calls are assigned to available vehicles in the order in which requests are received. If one (or more) vehicle(s) is (are) idle when the call arrives, the emergency call is assigned to the vehicle that has been idle longest. In nearest origin assignment, service calls arriving when one or more vehicles are idle are assigned to the nearest idle vehicle. In the flexible assignment strategy, at each simulation time point, the dispatch center will optimize the current assignment so as to minimize the total response time according to the associated weights of different classes of emergencies. Therefore, diversion is allowed among different emergency calls and rerouting of vehicles to emergencies are allowed in this strategy as well. In deployment strategy, responding vehicles can change destination to another emergency call or another candidate relocation site under the guidance of the dispatch center.

Table 7-6: Comparison of Dispatching Strategies

Vehicle Type	Strategy	Average Response Time (minute)	Longest Response Time (minute)	% of Emergency Exceed Waiting Time Limit	% of First Arrived Unit Exceed Waiting Time Limit
EMS	FCFS	8.63	20.10	21.36	14.49
	NO	3.51	14.37	16.44	11.71
	FA	3.07	9.78	13.73	8.39
	Deployment	3.02	7.90	8.92	6.58
Fire Engines	FCFS	10.45	24.20	25.64	17.58
	NO	4.27	17.34	19.77	14.93
	FA	3.74	11.79	16.50	10.29
	Deployment	3.63	9.54	10.80	7.73
Police Cars	FCFS	7.28	16.78	17.89	12.62
	NO	2.94	12.06	12.73	9.65
	FA	2.57	8.16	10.50	7.68
	Deployment	2.60	6.61	6.51	5.96

As shown in Table 7-6, the test results for ambulances indicate that the Flexible Assignment dispatching strategy performs better than the Nearest Origin and the FCFS dispatching strategies in terms of the average response time, and the Nearest Origin dispatching strategy is better than the FCFS dispatching strategy. Under the FA and deployment strategies, the percentage of first units arriving on the scene that exceeded the waiting time limit reduced by 44% and 56% respectively compared to that under the FCFS strategy. This is an important measure in the NFPA guidelines regarding response times. When the time interval between two consecutive emergencies is small, namely, when request calls are more frequent, the advantage is more prominent. This is shown in Section 7.4.1.

7.3.2 Comparison of Shortest Path Algorithm

In another set of experiments, we used the simulation model to test two different shortest path algorithms: the Dijkstra Algorithm with deterministic traffic information and the time-dependent shortest path algorithm with the real-time traffic information. In the first scenario we always used the optimal route obtained by using a static shortest path algorithm based on the off-peak time traffic information. This is the case in real-world operations, where drivers are provided with the routes calculated by off-the-shelf mapping software which use speed limits as travel speeds all the time. In the second scenario we used the time-dependent shortest path algorithm. Table 7-7 shows the simulation results for the average response times in these two scenarios. The time-dependent shortest path algorithm takes advantage of traffic fluctuation. When travel speed on the link is stable, the travel times under these two scenarios are the same. As shown in Table 7-7, when comparing the average response times during an entire day, the difference is not very impressive. However, if we consider the average response times during peak hours, the advantage of time-dependent shortest path algorithm is clear. For some extreme cases the response times decrease around 20% by utilizing the time-dependent shortest path and online traffic information.

When analyzing real-world response times (shown in Figure 7-5), it is noticed that the response times in 24 hours have the same distribution in general. This is because the emergency vehicle has high priority on road networks. With the help of an on-board signal control system, it is quite possible to avoid the potential delay caused by congestion.

Table 7-7: Comparison of Shortest Path Algorithms Based on Average Response Time

Time Interval	Average Response Time (Fixed Route) (minutes)	Average Response Time (DSP) (minutes)	Average Improvement (minutes)	Average Improvement (%)
7:30 am-8 am	3.36	3.24	0.12	3.6
8:00 am-10 am	3.72	3.55	0.17	5.4
Non-peak Time	2.47	2.47	0.0	0.0
All Day	3.04	3.01	0.03	1.0

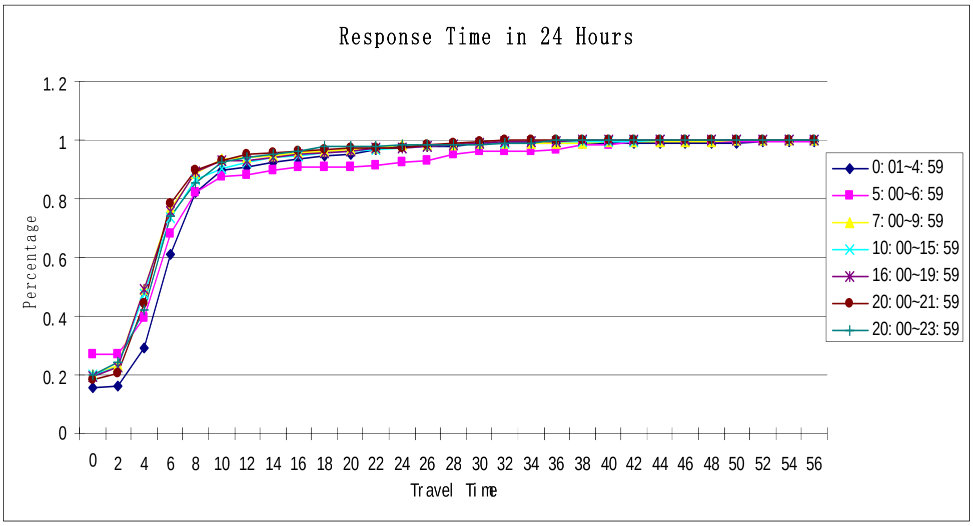


Figure 7-5: Response Time in 24 Hours

One thing to mention here is that the shortest path algorithm heavily depends on the time-dependent travel time prediction and the application of short-term emergency vehicle travel time prediction is very limited. The reasons are as follows:

1. Emergency vehicles travel in street network, and the research work on short term street network travel time prediction is limited;
2. Emergency vehicles are special type of vehicles, which have priority on the road, so the travel time prediction method for them should be different from regular vehicles;
3. It is difficult to get real data for model calibration. Most of the surveys are for the highway system, and not for the street network. It is difficult to cover the dense street network;
4. The travel time depends on numerous parameters, such as traffic volume, driver behavior, road configuration and weather conditions;
5. The time scale of prediction is small and the requirement of preciseness is high. To compute the time-dependent shortest path, we rely on precise or safe travel time prediction for the next 3-5 minutes time range.

7.4 Sensitivity Analysis

7.4.1 The System Workload

The system workload can be represented by the inter-arrival time of the emergency calls. When this inter-arrival time is long, the system is in a low workload status, when this inter-arrival time decreases, the system workload increases. From the analysis of real data, the average inter-arrival time is about 30 minutes, which is the base scenario. In this scenario, the simulation results show that the average response times for the Flexible and Deployment dispatching strategies are about 3 minutes. When we increase the workload the average response time increases dramatically. Table 7-8 shows the results of EMS units under each dispatching strategy with various system workload levels. Figure 7-6 and 7-7 illustrate the change of the corresponding average response time and the longest response time under these four dispatching strategies.

Under the FCFS strategy, the system performance is always the lowest. The NO strategy is the one used in most real operations. As mentioned in Chapter 3, in a myopic way, this can be the optimal policy. When compared with Flexible Assignment and Deployment strategies, the latter two strategies out-perform the NO in all performance measures. It is noticeable that the difference of average response times under the NO and the latter two increases when the inter-arrival time decreases (system workload increases). When the inter-arrival time is 30 minutes, the difference between NO and the other two strategies is about 18%, but this improvement is about 27% when the inter-arrival time decreases to 10 minutes. As shown in Figure 7-6, when the inter-arrival time is 30 minutes, the average response time of EMS units under DP is about 2% less than that of FA scenario. However, the number of vehicles that arrive at the emergency sites later than the maximum allowable waiting time for the emergencies decreases by 5%. This indicates that vehicle relocation allows more vehicles to reach emergency sites within the desired response time. When the system workload increases, the difference between the average response times under these two strategies is about 10%.

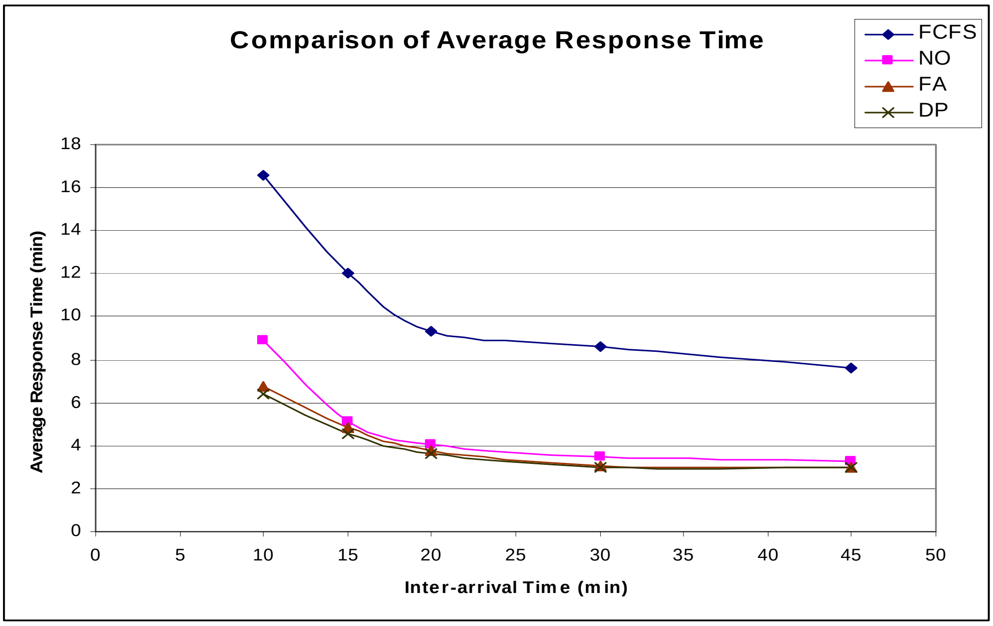


Figure 7-6: Comparison of Average Response Time

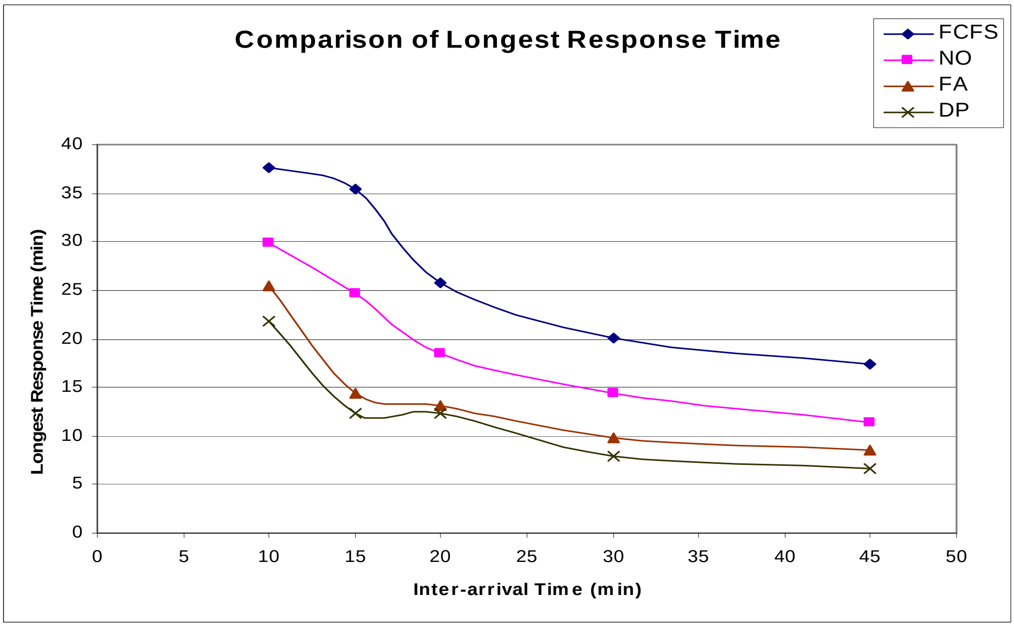


Figure 7-7: Comparison of Longest Response Time

Table 7-8: Sensitivity Analysis of System Workload

Dispatching Strategies	Inter-arrival Time (minute)	Average Response Time (minute)	Longest Response Time (minute)	% of Emergency Exceed Waiting Time Limit
First Come First Service	45	7.62	17.36	18.77
	30	8.63	20.10	21.36
	20	9.35	25.81	29.76
	15	12.02	35.41	36.94
	10	16.58	37.69	48.41
Nearest Origin	45	3.28	11.43	23.77
	30	3.51	14.37	16.44
	20	4.03	18.54	20.29
	15	5.11	24.73	22.66
	10	8.89	29.96	27.42
Flexible Assignment	45	3.01	8.48	11.47
	30	3.07	9.78	13.73
	20	3.78	13.05	15.97
	15	4.82	14.32	18.48
	10	6.73	25.39	25.16
Deployment	45	2.96	6.63	4.51
	30	3.02	7.91	8.92
	20	3.65	12.40	11.34
	15	4.56	12.40	15.41
	10	6.38	21.79	23.77

7.4.2 Penalty Parameters

For the deployment dispatching strategy, the penalty parameters in the objective function have a potential impact on the average system performance measures. When the coverage penalty is much larger than the penalty parameters for the assignment requirement, the system is more likely to dispatching more vehicles to the candidate relocation site instead of dispatching them to some less important emergency calls. Therefore, it is important to perform sensitivity analysis on these penalty parameters. Here two groups of experiments are designed, in the first group, the weight of the travel time for all types of emergency calls are the same, and we increase the coverage penalty , in the second group, we design a group of based on the ratio of these parameters. Table 7-9 summarizes the performance measures under various ratio combinations. With uniform parameter values, the shortest average response times can be achieved but the longest response time and the percentage of emergency calls exceed waiting time limits are the highest. When the value of coverage penalty parameter increases, the average response time slightly increases but the other two performance measures improve accordingly. Figures 7-8 and 7-9 illustrate the relationship between parameter ratios and the average response times and the percentage of calls with waiting time longer than the required time window for each type of emergency call, respectively. From the analysis, it is difficult to draw a conclusion for the optimal parameter values. The decision makers have to consider trade-offs between the average performance and the percentage of undesirable performance.

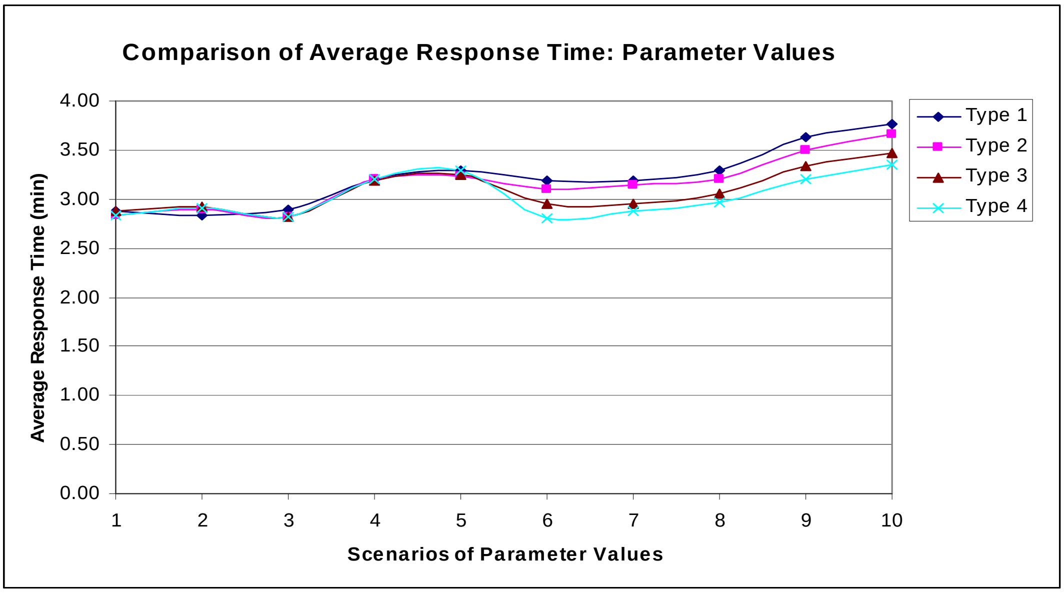


Figure 7-8: Comparison of Average Response Time under 10 Parameter Scenarios.

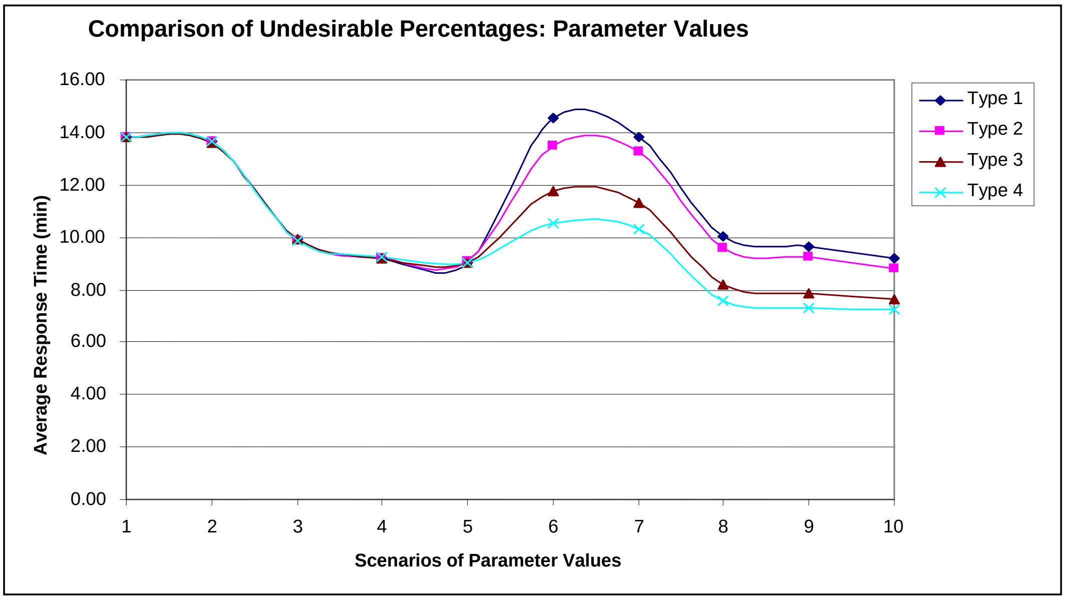


Figure 7-8: Comparison of Under-performed Percentages under 10 Parameter Scenarios

Table 7-9: Sensitivity Analysis of Parameters under the Deployment Strategy

Parameter Ratios (C_1:C_2:C_3:C_4:C_c)	Average Response Time (minute)	Longest Response Time (minute)	% of Emergency Exceed Waiting Time Limit
(1:1:1:1:1)	2.78	10.65	13.79
(1:1:1:1:5)	2.83	10.61	13.61
(1:1:1:1:10)	2.81	8.72	9.84
(1:1:1:1:20)	3.18	8.47	9.17
(1:1:1:1:50)	3.23	8.34	9.00
(1:2:3:4:1)	2.94	9.63	12.51
(1:2:3:4:5)	2.95	9.63	12.32
(1:2:3:4:10)	3.02	7.91	8.92
(1:2:3:4:50)	3.32	7.63	8.27
(1:2:3:4:100)	3.46	7.52	8.12

7.4.3 Benefit Threshold Sensitivity Analysis

The diversion benefit threshold restrains the frequency of destination change for emergency vehicles. When this threshold τ is small, the vehicle can change its destination when the saving resulting from the diversion is no less than τ. When τ increases from 30 seconds to 3 minutes, the number of diversion decreases as well. There is no significant change in the average response time. But when τ is 30 seconds, the longest response time is about 12% smaller than that of the scenario with a threshold of 3 minutes as shown in Figure 7-9.

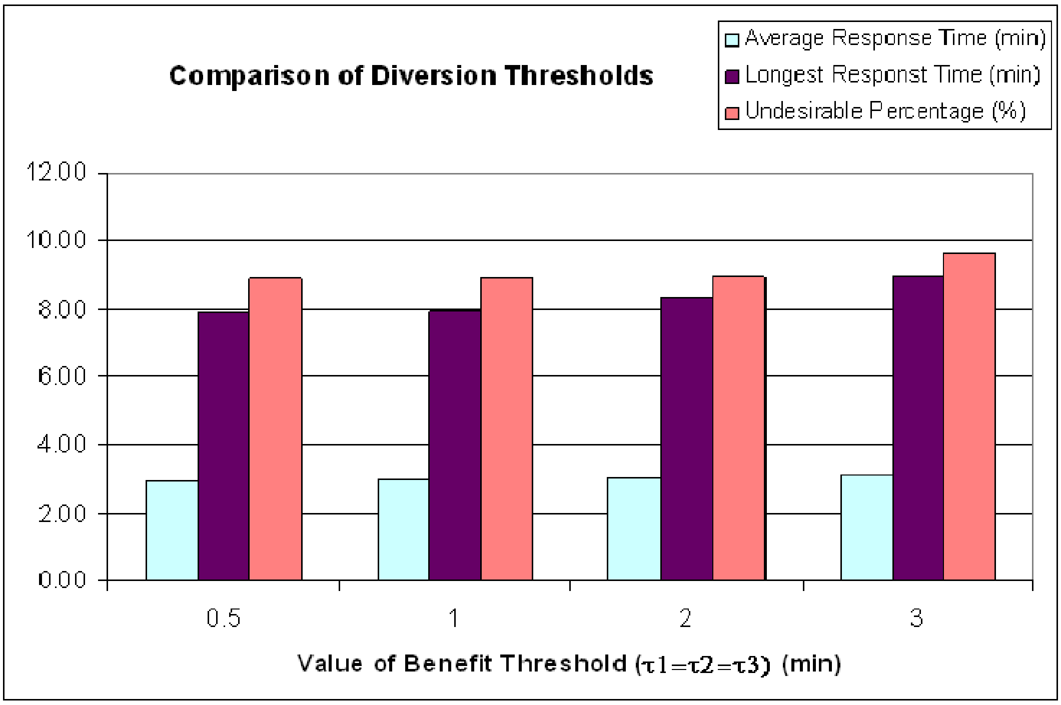


Figure 7-9: Impacts of Diversion Benefit Threshold on Performance Measures

7.4.4 Depot Locations and the Fleet Allocation

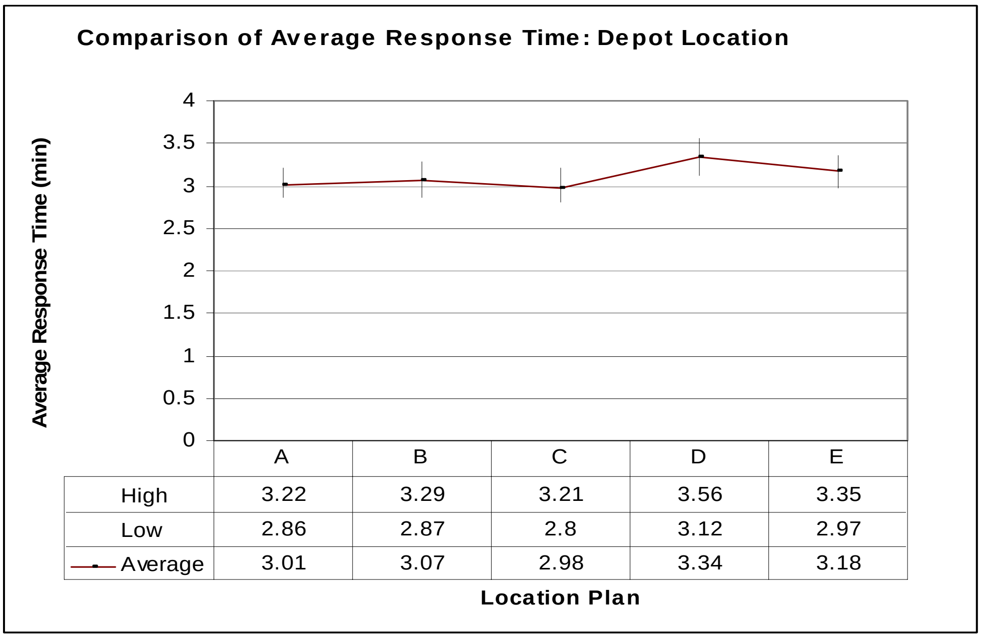
For certain types of emergency vehicles such as ambulances and fire engines, vehicles will stay in the depot when available. Therefore, the location of the depot and the allocation of the fleet have certain impacts on the system performance.

Since there are 10 depots in the network, besides the real locations (plan A), we randomly selected other location plans as listed in Table 7-10, and tested the performance of our approach. Figure 7-10 shows the average response times under each location plan. This group of experiments shows a better location plan can save significant amount of response time. In this set of experiment, the best scenario (plan

C) has an average response time of 2.98 minutes while the worst case (scenario D) has an average response time of 3.34 minutes. The difference is more than 11%.

Table 7-10: Sensitivity Analysis of Depot Locations

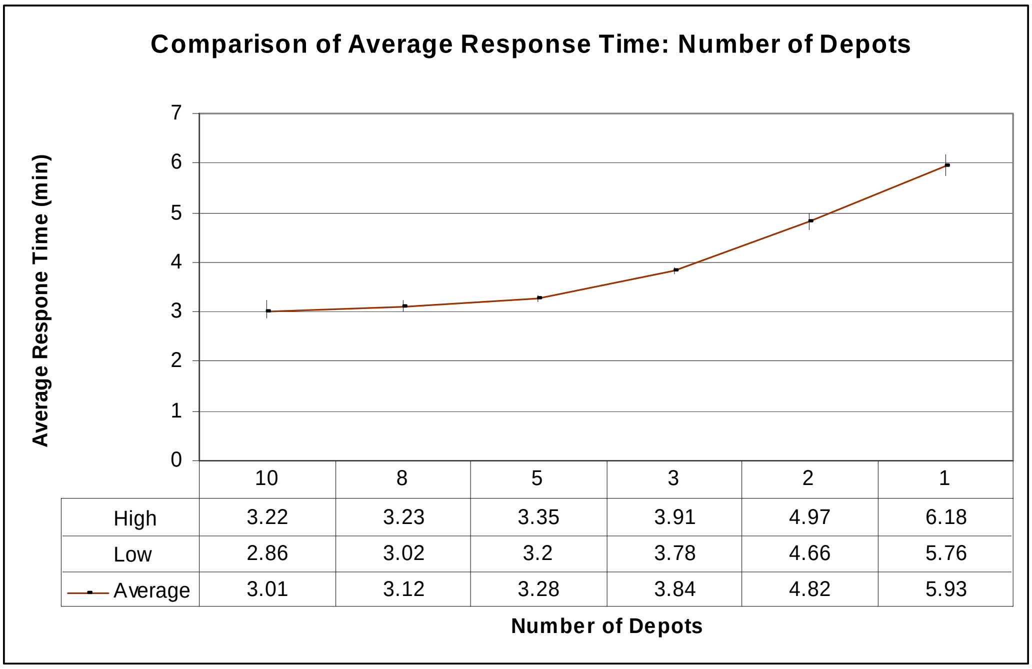
Plan	Depot Location (node number)
A	1226, 2334, 1240, 5496, 2476, 1426, 3878, 4922, 5427, 4546
B	945, 1212, 1456, 1870, 2331,5496, 4673, 3771, 4839, 3187
C	2690, 3464, 3801, 5169, 2226, 3171, 2969, 3513, 2467
D	810, 1008, 607, 76, 4846, 1619, 2823, 5234, 4331, 469
E	3957, 1294, 5407, 4451, 2706, 4439, 17, 1718, 4470, 4149



	A	B	C	D	E
High	3.22	3.29	3.21	3.56	3.35
Low	2.86	2.87	2.8	3.12	2.97
Average	3.01	3.07	2.98	3.34	3.18

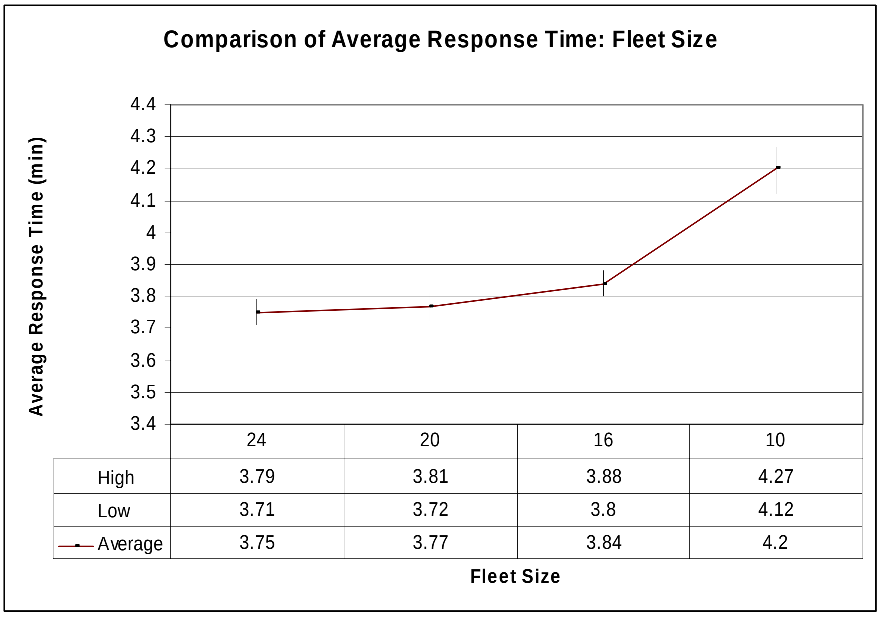
Figure 7-10: Impacts of Fleet Allocation on Average Response Time

Assume the number of depots is not given. In that case there can be numerous potential combinations of depot location plans. Figure 7-11 shows the variation of average response time with respect to different number of depots at arbitrary locations. When the number of depots reduces from ten (current situation) to five, the average response time increases by about 9%, but when only one depot can be operated, the average response time increases by 99%.



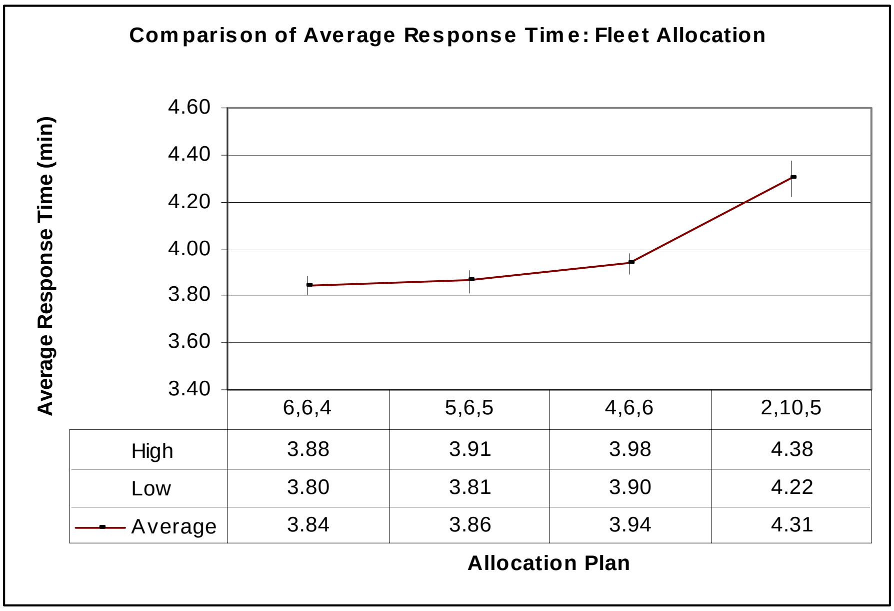
	10	8	5	3	2	1
High	3.22	3.23	3.35	3.91	4.97	6.18
Low	2.86	3.02	3.2	3.78	4.66	5.76
—▪— Average	3.01	3.12	3.28	3.84	4.82	5.93

Figure 7-11: Impacts of Number of Depots on Average Response Time



	24	20	16	10
High	3.79	3.81	3.88	4.27
Low	3.71	3.72	3.8	4.12
Average	3.75	3.77	3.84	4.2

Figure 7-12: Impact of Fleet Size on Average Response Time (3 Depots)



	6,6,4	5,6,5	4,6,6	2,10,5
High	3.88	3.91	3.98	4.38
Low	3.80	3.81	3.90	4.22
Average	3.84	3.86	3.94	4.31

Figure 7-13: Impact of Fleet Allocation on Average Response Time (3 Depots)

Similarly, with different fleet sizes the average response times will vary as well. Figures 7-12 and 7-13 show the variation of average response times when the vehicles are allocated differently, and when the fleet size is variable, respectively. As expected, when the EMS fleet size reduce from 24 to 10, the average response time increases by 12%, while with a fleet size of 16, a good allocation plan can save 10-12% in average response time as well. These analysis results indicate that it is very important to have a good facility location/allocation plan.

In Appendix II, we illustrated an example to use the proposed integrated system in a long term EMS location/allocation planning problem. The approach was tested with a real network and the results indicate that very good quality solutions can be produced. Similarly, we can use the proposed system for performance evaluation and other planning problems such as the fleet size problem.

7.5 Conclusions

In this chapter, we applied the formulations, heuristic algorithms and the simulation model that were developed in this research to real-world problems. A real-world network and its operational data were used to calibrate the simulation model. Different dispatching strategies were tested in the simulation model. By using CPLEX, we obtained optimal solution for small problems and when the problem size was large, the heuristic was called to solve the optimization of fleet deployment problem. Promising improvements can be obtained over the original Nearest Origin dispatching strategy which is widely used in real operation. More than 10% savings on average response times and better service performance measures are achieved.

Sensitivity analyses in this chapter suggested that the average performance is sensitive to the workload of the system, but not very sensitive to the coverage penalty. Though the average response time may not decrease too much but the longest response time is effectively reduced. The management can require a more balanced dispatching plan that is more operable.

In sensitivity analysis with respect to the parameters we found that there is a trade-off between the average response time and the percentage of emergency calls which exceed the waiting time limit. When the weight of the coverage penalty parameter is higher, the average response time slightly increases, while the undesirable performance percentage decreases. According to this analysis, our model and algorithm can provide the management more choices to select more balanced operational strategies.

Chapter 8: Conclusions and Future Research

8.1 Summary and Conclusions

In this dissertation, we presented our research dealing with the Emergency Vehicle Dispatching Problem. Based on the real-world operations, a system framework was proposed which integrated the information and activities of Transportation agencies, Fire Departments, Police Departments and EMS provides.

The emergency vehicle deployment problem is the kernel of the system. In this problem, we considered:

1. Real-time vehicle deployment;
2. Multiple emergency types;
3. Multiple emergency vehicle types; and
4. Service coverage.

All these extensions are required in real-world operations. In Chapter 1, we explained the field operation routing and why this integrated system is necessary. After a broad review of several relevant problems and the solution approaches in chapter 2, we presented the mathematical formulation of the emergency vehicle deployment problem in Chapter 3.

In Chapter 3, we started from a simple example which illustrated concerns in real operations. Emergency vehicle dispatching problem is not only an assignment problem, but also an allocation problem to find the best configuration in order to provide better service to the entire area. First a mathematical model for dispatching only was presented which included extensions of multiple-vehicle assignment and multiple types of vehicles. Then we expanded the dispatching formulation into a deployment problem formulation with service coverage concerns. Since the emergency vehicle deployment problem is an NP-complete problem, we tried to find good lower bounds to assess the quality of the solutions generated by the heuristic algorithms.

In Chapter 4, we presented several algorithms for solving the emergency vehicle deployment problem. Because of the nature of the problem, the computational time is an important issue when achieving the optimal solution. To effectively reduce the computational time caused by the relocation constraints in the model, a rolling horizon approach was introduced to reduce the size of potential relocation sites. Several new solution methods were discussed in this chapter. The solution approaches we discussed included:

1. Initialization methods;
2. Improvement methods; and
3. Tabu search heuristics.

All of these solution approaches can be customized for different problems for different purposes.

Before testing our research on emergency vehicle fleet deployment in a real-world operational environment, we examined different lower bound techniques. We analyzed the possibilities of using the linear relaxation and the Lagrangian relaxation method to obtain a lower bound and concluded that both methods cannot provide a good lower bound for the emergency vehicle fleet deployment problem. We used a decomposition method to produce lower bounds. We designed experiments tested the algorithms we proposed in Chapter 5. Comparing the lower bound results with those obtained from optimal solutions we conclude that the proposed algorithms can obtain nearly optimal solution within reasonable computational time.

The goal of our research was to solve a real-world problem. Since it is difficult to test the new models and the different dispatching strategies in real-operation, we developed a simulation model to test various scenarios. The simulation system is driven by two major types of events: the fixed-increment traffic information updating and the variable vehicles/emergency calls status updating. Chapter 6 discussed the essential modules in the simulation model which update the system time and status, as well update the travel time, the deployment plan updating and statistical analysis. The flow chart for each dispatching strategy was presented in this chapter as well.

In Chapter 7, we presented a thorough case study based on the traffic network of on one the counties in the Washington DC metropolitan area. First, we used the operational data to calibrate the simulation network. 4 different dispatching strategies were compared. The simulation showed that compared with the historical operations, significant savings (16.26%) of average response time can be obtained by applying the Flexible Assignment model and another (2%) can be obtained by applying the Deployment model. The comparison of different shortest path algorithm showed that with the real-time traffic information, the dynamic shortest path algorithm can bring about 3-4% savings in average response time in peak hours. To help the management improve the operation, we also conducted several sensitivity analyses for different parameters. First, we tested different system work-loads. We found that the final solution is very sensitive to the emergency call inter-arrival time as expected. Secondly, we tested different cost parameters. When the penalty parameter for the coverage requirement is high, the vehicles try to cover more nodes in the network rather than to service all the emergency calls in the most expedient way. When the coverage parameter is low, there will be more calls with long waiting times. The balance of these two issues requires a fine tuning of parameters.

Finally, we applied this simulation model in a facility location and allocation problem. In this problem, a genetic algorithm was designed in which each chromosome represented an allocation/location scheme. The simulation model was used to evaluate the performance of each plan. A set of activities were used to generate the new solutions from the population. Since the evaluation function is weighted by a set of parameters, the management may have more freedom to choose different plans based on the real consideration of service level and budget limit.

In summary, the following key contributions are made in this research:

1. A new mathematical formulation of real-time emergency vehicle dispatching problem is presented. No similar model has been formulated for emergency response system with multiple vehicle types and multiple emergency types. The model takes the assignment problem and relocation problem into consideration simultaneously.
2. Several improvement algorithms that perform well on real networks are suggested. The algorithms can provide quality solution within short computational times.
3. Different lower bound methods are developed and analyzed. Acceptable lower bounds and good improvements are obtained.
4. A simulation model is developed which incorporates the multiple types of vehicles and activities of real operations in the proposed system framework.
5. A case study is completed by applying our algorithms, formulations and lower bound methods. The simulation results indicate that significant improvements can be achieved by applying the proposed system.

8.2 Future Research

Although many achievements have been made in this research, there are still many problems that are unsolved and are left for future study.

8.1.1 Lower Bound Method

Although we obtained some acceptable lower bounds by applying decomposition strategies, the lower bounds for large networks are not yet good enough. Investigating other methods to produce good lower bounds both for the problems on large networks and large fleet size is an important area for future research.

The main problem in using the decomposition method is that the problem is a dynamic one. The quality of the dispatching scheme in the future is highly correlated to the solution at current time. More domain decomposition strategies may be explored to achieve better bounds.

In this dissertation, we conclude that the Lagrangian Relaxation method cannot produce good lower bounds for the emergency vehicle dispatching problem, however, this does not mean that Lagrangian Relaxation method cannot be applied here. Instead of relaxing the coverage constraints, some other constraints can be relaxed to produce good lower bounds too. We did not explore the opportunity in this direction, but this is possibly a good direction for future research.

Finding good lower bounds for NP-complete problem is always a challenging work for researchers. For a field application, it is more important to find a good or acceptable solution efficiently. From this aspect, it is more important to design a good algorithm.

8.1.2 Heuristic Algorithms

The algorithms are designed for the real-time emergency vehicle dispatching problem. In chapter 7, we applied our test algorithms in the simulation process and showed that we can produce good solutions that can save 18.6% in average response time, and the computational time is always within 30 seconds. However, these algorithms are not designed for efficiency for all problems. For example, if the fleet size is small, and the underlying network is very dense and the size of node set within time contour are very large, this heuristic algorithm may not be able to provide very good solution (test case 5).

Every improvement method may cause local optima. The difficulty of avoiding being trapped in local optimal is no less than the difficulty of finding a good lower bound. As seen in the tests, it is difficult to avoid the local optima caused by the initialization method. Instead of applying the adding/dropping/swapping in sequence, we may just use random node exchange improvement method by generating different random numbers. So the improvement methods can randomly be used in the whole improvement process. This can reduce the possibility of the local optima caused by reapplying one improvement method.

Control processes can be designed to guide the generation of the random numbers. For example, a random number can be generated to make sure that the add/drop improvement method or swap method is picked more frequently. In our study, tabu search algorithm is applied to deal with the difficulty of getting around the local optima. There are two aspects that the tabu search methods can be improved. The first is to find a good tabu expiration strategy and the second one is the Elite Pool. In this research fixed tabu tenure is used and different tabu tenures are tested. This method is not efficient when the tabu tenure is not good. A more robust method can be designed for the tabu tenure. Variable tabu length can be used according to the previous search experiences. However, designing a viable tabu length is difficult and somewhat arbitrary. The Elite Pool strategy can be applied to improve the solution as well. For the efficiency of the algorithm, we can limit the size of the assignments by variable pool sizes. A more advanced idea to improve the Elite Pool strategy is that instead of constructing the next initial solution from the assignment in the EP, the next initial solution can be constructed from patterns. So instead of saving elite assignment into the pool, we can save the common patterns in the Elite Pool. For example, a pattern can be defined that vehicles A and B are selected together in one scheme or vehicles A and B are not selected together. These patterns are obtained from previous computing experiences. By using patterns, the information about meritorious solutions is stored in a smaller size, so the computational time can be reduced. When an elite assignment is added to the pool, the information of the elite route is digested into patterns and stored in the EP. But how to define the patterns and find the patterns will be a challenging task. In future research, experiments and tests in this direction are also very important.

8.1.3 Others

When more information regarding police patrol services are available, incorporating the police patrol routing problem into the emergency vehicle deployment problem will have significant application potential. Instead of dispatching police cars to specific locations, it is more realistic to provide police cars with patrol routes which can provide coverage to most demands in an area.

In this dissertation, multiple objectives (minimizing cost and maximizing coverage) were considered in the models and weights were used to transform multiple objectives into a uniform objective function value. An economic analysis of the tradeoffs between the operational costs and the benefits gained from vehicle relocations and reassignments is an interesting area for future research.

Since the operation of a system is supported by both the fleet and crew, it is important to consider the corresponding crew scheduling problem, especially for the Flexible Assignment and the Deployment dispatching strategies. Since the crew will be transferred to various locations, a matching crew scheduling scheme can be as crucial as the vehicle dispatching scheme itself. Another related optimization problem is to find the proper fleet size or crew size for a system when certain performance level is designated.

How to utilize the GIS technology to support the emergency response process can be another research area. Recently, GIS technologies are widely used in environmental risk analysis. This application is extremely important for HAZMAT related emergencies, such as oil spills and gas leakage. The application of GIS can quickly provide the impact area and provide emergency response guidance. The linkage between GIS technology and the developed emergency response system requires further study.

In summary, future research should continue to find good lower bounds and improving the algorithms, as well as to look at other relevant optimization problems in this emergency response system.

Appendix I

DETAILS OF SIMULATION

I.a. Travel Time $T_k(t)$ Generation

The following steps are used to generate $T_k(t)$:

1. The polar method (see section 6.4.2) is used to generate two independent identical normally distributed random variables in pair, $X_1 \sim N(0,1), X_2 \sim N(0,1)$.

2. $T_k(t)$ is calculated from equation (I-1). $T_k(t)$ is a normally distributed random variable with mean $\mu_k(t)$ and variance $\sigma_k{}^2(t)$.

$$T_k(t) = \mu_k(t) + X_1\sqrt{\sigma_k(t)}\text{ , and } T_k(t) \sim N(\mu_k(t), \sigma_k(t)). \qquad \text{(I-1)}$$

Where, $\mu_k(t)$ is related to the average travel time and the variance/mean ratio is given. The relationships are as follows:

$$\mu_k(t) = \kappa \cdot \omega_k \cdot length / Speed \qquad \text{(I-2)}$$

$$\sigma_k{}^2(t) = \mu_k(t) / \rho_k \qquad \text{(I-3)}$$

$$T_k(t) \sim N\left(\mu_k(t), \sigma_k{}^2(t)\right) \qquad \text{(I-4)}$$

Where $length$ is the length of a link, $Speed$ is the design speed on that link, κ is a coefficient used to calculate travel times on the basis of average non-peak travel times, and ρ_k is the coefficient to represent the randomness of the travel time. $T_k(t)$ is the predicted travel time for a particular link starting at time t based on the current situation.

In a graph $G = (V, E)$, where V is the set of nodes with finite directed link set E connecting the nodes, let $d_{ij}(t)$ be the nonnegative time required to travel from node i to node j when the departure time from node i is t. $d_{ij}(t)$ is a real-valued function defined for every $t \in S$, where $S = \{t_0, t_0 + \delta, t_0 + 2\delta, \cdots, t_0 + M\delta\}$, t_0 is the earliest possible departure time from any origin node in the network, δ is a small time interval during which there is some perceptible change in traffic and M is a large integer number such that the interval from t_0 to $t_0 + M\delta$ is the period of interest, namely, the transient period.

I.b. Implementation of the Dynamic Shortest Path Algorithm

The algorithm is based on the First-In First-Out (FIFO) assumption of urban transportation networks. It is assumed that $d_{ij}(t)$ for $t > t_0 + M\delta$ is constant and equal to $d_{ij}(t_0 + M\delta)$ and after the peak hour stable travel times can be used. M is a user-defined parameter and can always be increased to include periods with variable travel times on some links. It is also assumed that $d_{ij}(\tau) = d_{ij}(t_0 + k\delta)$ for every τ in the interval $t_0 + k\delta < \tau < t_0 + (k+1)\delta$. This is not a restrictive assumption, considering that by definition δ is very small. However, with a larger M, the computational time will increase.

At each step of the simulation, the algorithm proposed by Ziliaskopoulos and Mahmassani is called to calculate the time-dependent shortest paths from every node i in the network and at every time step t to the destination node N . The algorithm for computing the dynamic shortest path between each O/D pair and each starting point is available in literature (Ziliaskoulos and Mahmassani, 1993) and the pseudo code is as follows:

```
Call Creation
Call Insertion (N)
Do 1 While (SE list is not empty)
Call Deletion (CurrentNode)
        Do 2, for (All nodes J that can directly reach Current Node)
            NextnNode=J
            InertInSEList=False
            Do 3, for (t=1, M)
                    CurrentTravelTime=TravelTime(NextNode, CurrentNode, t)
                NewLabel=LABEL(CurrentNode, t+CurrentTravelTime)+
                              Current Travel Time
                If (LABEL(NextNode,t)<=NewLabel) then
                    LABEL(NextNode,t)=NewLabel
                    InsertInSEList=TRUE
                    PathPointer(NextNode,t,1)=NodeCurrent
                    PathPointer(NextNode,t,2)=t+CurrentTravelTime
                End If
    3       continue
            If(InsertInSEList) Call Insertion (NextNode)
    2       Continue
    1       Continue
```

Procedure Creation

```
    Do, For (Node=1,V-1) Deque(Node)=0
    Deque(N)=99999
    FIRST=N
    LAST=N
```

Procedure Deletion (Current Node)

```
CurrentNode=FIRST
FIRST=Deque(CurrentNode)
Deque(CurrentNode)=-1
```

```
Procedure Insertion (Node)
    If (Deque(Node)=0) then
            Deque(LAST)=Node
            LAST=Node
            Deque(Node)=99999
    Else
        If(Deque(Node)=-1) Then
                Deque(Node)=FIRST
                FIRST=Node
            End If
End If
```

Denote $\lambda_i(t)$ the total travel time of the current shortest path from node i to node N at time t. Let $\Lambda_i = [\lambda_i(t_0), \lambda_i(t_0+\delta), \cdots, \lambda_i(t_0+M\delta)]$ be an M-vector label that contains all the labels $\lambda_i(t)$ for every time step $t \in S$ for Node i. Every finite label $\lambda_i(t)$ from Node i to Node N is identified by the ordered set of nodes $P_i = \{i = n_1, n_2, \cdots, n_m = N\}$.

$\lambda_i(t)$ is defined by the following functional equation:

$$\lambda_i(t) = \begin{cases} \min_{j \neq i} \{d_{ij}(t) + \lambda_j[t + d_{ij}(t)]\} & for\ i = 1,2,\cdots,N-1; t \in S \\ 0 & for\ i = N; t \in S \end{cases} \qquad \text{(I-5)}$$

Instead of scanning all the nodes in every iteration, a list of scan eligible (SE) nodes is maintained, containing the nodes with some potential to improve the labels of at least one other node. The proposed algorithm operates in a label correcting fashion; therefore, the label vectors are just upper bounds to the shortest paths until the algorithm terminates.

Initially the SE list contains only the destination node N . In the first iteration all the nodes that can directly reach N are updated according to Equation I-6 and inserted in the SE list.

$$\lambda_i(t) = d_{iN}(t) + \lambda_n[t + d_{iN}(t)] \qquad i \in \Gamma^{-1}\{N\} \tag{I-6}$$

Where $i \in \Gamma^{-1}\{N\}$ is the set of nodes that can directly reach N . The rest of the labels are set equal to infinity. Next, the first node of the SE list is scanned according to the following equation:

$$\lambda_j(t) = \min\{\lambda_j(t), d_{ji}(t) + \lambda_i[t + d_{ji}(t)]\} \qquad j \in \Gamma^{-1}\{N\} \tag{I-7}$$

For every time step $t \in S$, if at least one of the components of Λ_j is modified, Node j is inserted in the SE list. This scheme is repeated until the SE is empty and the algorithm terminates. Equation (I-6) and (I-7) are modifications of Equation (I-5).

At the end of this procedure, each element of the vector label is a finite number that represents the shortest path from this node and time step to the destination node since our network is a connected graph. If there is any isolated node, the vector label of that node will be infinity, which is represented by a large number.

The detailed steps of the algorithm are described in Ziliaskopoulos and Mahmassani (1992). The implementation of this algorithm is similar to the implementation of a static label-correcting algorithm. The three principal implementation issues are the network representation, the data structure of the SE list, and the path storage.

A queue structure is implemented to represent the structure of the SE list. The queue structure allows the insertion of nodes at any position of the SE list according to a predetermined strategy and removal from any position of the SE list. Two pointers are kept, one pointing to the first (FIRST) and the other to the last (LAST) node in the queue. The following operations are defined associated with this structure:

- Creation: Creation is an initialization step, which is activated just once to set Queue(i) = 0, $i = 1,2,\cdots,N-1$ and Queue(N) = ∞. Infinity is defined practically as a very large number, for example 999,999. This operation also sets the variable FIRST = LAST = N. The whole operation requires $N + 3$ computational time units.

- Insertion: Insertion involves inserting a node at the beginning or the end of the queue. To determine the insertion point, the operation checks the value of Queue(i). If it is 0, indicating that node i has never been in the queue, the node is inserted at the end of the SE list and the value of the pointer LAST is set equal to i and Queue(i) = ∞. If Queue(i) = -1, Node i is inserted to the end of the queue; Queue(i) is set to FIRST, and the value of FIRST $= i$. Otherwise, it does nothing because the node is already in the queue. The computational effort required by this step is three time units.

- Deletion: Deletion selects the first element of the queue and assigns it to the variable "Current Node". Then, it changes the value of the FIRST to the second element in the queue (which is the Queue(FIRST) node). It sets the values of Queue(Current Node) to –1. The computational effort for this operation is three time units.

An important issue is the computational time of the algorithm. The creation operation is called only at step 1 of the algorithm and does not contribute significantly to the total computation time of the algorithm. On the other hand, deletion and insertion are called repeatedly from step 2. So they are critical in the determination of the total computational effort of the algorithm.

Finally, the paths are maintained in an $M \times 2$-dimensional array of pointers for each node. These pointers point to the successor node and its label address.

I.c. Random Number Generator

Thus, based on the prime modulus multiplicative linear congruential generator (PMMLCG) algorithm with a multiplier 7^5 = 16807 and a prime modulus 2^{31}-1 = 2147483647, this random number generator has a satisfactory period for use in simulation.

In this study, the seeds are generated from the above random number generator with step of 100,000 numbers. It is important to assure that the separation between adjacent seeds is far enough to avoid overlap, i.e. correlation, in generated random sequences. The detailed steps of generating various random variates are listed below.

Uniform

The distribution function of a $U(a,b)$ random variable can be easily generated by solving $u = F(x)$ for x to obtain, for $0 \le u \le 1$,

$x = F^{-1}(u) = a + (b-a)u$

Thus, the inverse-transfer method is used to generate X:

1. Generate $U = U(0,1)$.
2. Return $X = a + (b-a)U$

Exponential

The exponential random variables with mean $\beta > 0$ can be derived by solving the following inverse-transform algorithm:

1. Generate $U = U(0,1)$.

2. Return $X = -\beta \ln U$.

Normal

The polar method is used to generate N(0,1) random variates in pairs:

1. Generate two independent identical uniformly distributed random variables U_1 and U_2 from $U(0,1)$; let $V_i = 2U_i - 1$ for $i = 1,2$; and let $W = V_1^2 + V_2^2$.

2. If $W > 1$, go to step 1. Otherwise, let $Y = \sqrt{(-2\ln W)/W}$, $X_1 = V_1 Y$, and $X_2 = V_2 Y$. Then X_1 and X_2 are independent identical normally distributed random variables, that is, $X_1 \sim N(0,1), X_2 \sim N(0,1)$.

Appendix II

AN APPLICATIOIN IN THE FACILITY LOCATION PROBLEM

II.1 Background

From the comparison in Section 7.4.3, it is noticed that the location and allocation of the facilities have a major impact on the performance measures. Better location/allocations plan can also save precious response time. In this section we focus on the location/allocation of medical service unit problem. The mathematical model and simulation models are utilized in an integrated approach which solves the location and allocation problem simultaneously.

Assume that the depots can be located at any node $n \in N$, with v_n medical service units. Since each station should have at least one medical service unit, we know that there can be $\sum_{i=1}^{v} \binom{i}{N}$ different location configurations, which is a huge number when either N or V is large. When considering the fleet assignment problem simultaneously, it becomes even more complicated. Therefore an efficient and integrated solution method is needed.

The optimal locations of the EMS units need to ensure that the total weighted cost is minimized. The total weighted cost we use in this research to determine the depot locations are average response time and cost. Capital costs are basically related to the number and the size of the stations. The annual operating cost is composed of the salary of the staff and crew, the vehicle and building maintenance costs, medicine and materials supply costs, etc. Usually, the relationship between the cost and the fleet size is nonlinear and concave which means that when the fleet size increases, the rate of increase in operating costs decreases.

II.2 Solution Approach

The approach we propose for solving for the optimal location of the depots and determination of the fleet assignment is an iterative approach that is a combination of simulation and genetic algorithm. At each iteration the location and the fleet assignment generated by a GA is imported into the simulation program and the output of the simulation program is used in the evaluation of the chromosome for the GA to generate the next population of solutions.

Genetic Algorithm

A genetic algorithm (GA) is used here because the exact solution is impossible to get by analytic approaches. Among the heuristics, GA has an efficient procedure of natural selection, which assures it can produce quality solutions in reasonable time. Here, a chromosome represents the set of nodes N and the gene represents the number of medical service units in a location. The performance is evaluated based on the fitness values that are based on the simulation results. In the process, a number of chromosomes are generated, evaluated and selected from one generation to the next, and eventually the chromosome with the best performance is chosen as the optimal depot location. The length of a chromosome is N, that is the total number of nodes in the map, a gene in the chromosome denotes the number of medical service units located at that node. Hence an integer genetic algorithm is employed, the gene is defined as 0 if it is not used as depot location; otherwise it is the number of medical service units assigned to that location. The set of genes that have integer values greater than or equal to1 is the set of depots S. The value of each gene stands for the fleet size at that station.

Assuming the network has 12 possible candidate locations and 4 available medical service units, if the initial chromosome/solution is as shown in Figure II-1, it means that there will be one depot at node 4 with one medical service unit, one depot at node 7 with 2 medical service units and another depot at node 11 with 1 medical service unit.

0	0	0	1	0	0	2	0	0	0	1	0

Figure II-14: An Example of Chromosome

The detailed procedure of the genetic algorithm is stated as follows:

Step 1: Initial population.

Randomly generate an initial population of the depot locations and their fleet assignment.

This initial population has M chromosomes. For each chromosome, we pick one random number from 1 to N for V times. Every time when one random number is selected, the value of the gene on that position is increased by 1, so that the fleet size constraint is satisfied.

Step 2: Chromosome evaluation.

Each chromosome is translated into a depot configuration, and this configuration is introduced into the simulation program. The results of the simulation are used as the input for the evaluation function. We build our evaluation model based on the fact that the sizes of the facilities located at the depots are closely related to the number of medical service units they accommodate. Therefore, the objective function can be simplified as a linear function of the number of medical service units in each location and the average response time.

$$Min\,Z = \sum_{\mathrm{i=1}}^{\mathrm{S}} n_i \cdot C_i + C_T \cdot \overline{T} \tag{II-1}$$

where $\mathrm{n_i}$ is the number of medical service unit allocated to the $\mathrm{i^{th}}$ depot, and $\mathrm{C_{i\ and}}$ $\mathrm{C_T}$ are the coefficients for the fleet size and the average response time respectively. The parameters are tested in the sensitivity analysis.

Step 3: Natural selection.

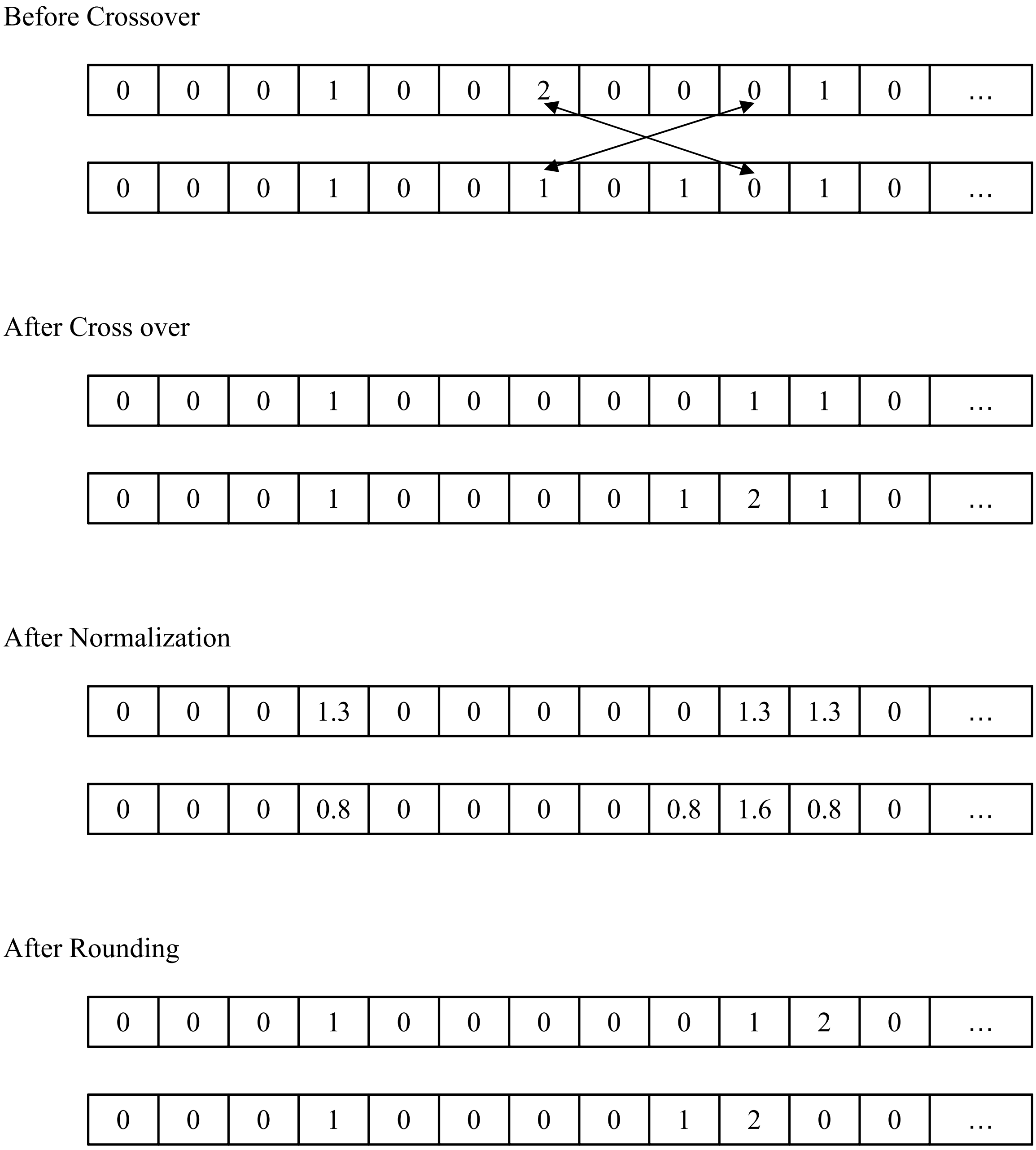
Parts of chromosomes with better performance indices are selected and the rest are discarded according to the principle of the “survival of the fittest”. The elitism is protected by saving the good solutions at each iteration.

Step 4: Crossover.

The chromosomes that survive are selected to be the parents of the next generation, and pairs are made among them randomly. The crossover operation is used to exchange genes between the parent chromosomes to generate offspring with better fitness values.

The Roulette wheel method is used here. The possibility of appearance depends on the fitness value of the chromosome. The better the fitness value is, the more possible it is that the chromosome will be chosen as a parent.

One should note that because of the fleet size constraint, in each chromosome the total value of the genes must be equal to V. When crossover and mutation operations are performed there is no guarantee that the fleet size constraint will be satisfied. Therefore, normalization is necessary. If a chromosome transfers a gene 1 to its spouse and gets a gene 0 back at one place (as shown in Figure II-2), we calculate the weight of each gene according to the current chromosome and assign the fleet to each gene based its weight. Because the number of medical service units must be integer, rounding is necessary and makes sure that after rounding, the total number of medical service unit equals the fleet size.



Before Crossover

0	0	0	1	0	0	2	0	0	0	1	0	…
0	0	0	1	0	0	1	0	1	0	1	0	…

After Cross over

0	0	0	1	0	0	0	0	0	1	1	0	…
0	0	0	1	0	0	0	0	1	2	1	0	…

After Normalization

0	0	0	1.3	0	0	0	0	0	1.3	1.3	0	…
0	0	0	0.8	0	0	0	0	0.8	1.6	0.8	0	…

After Rounding

0	0	0	1	0	0	0	0	0	1	2	0	…
0	0	0	1	0	0	0	0	1	2	0	0	…

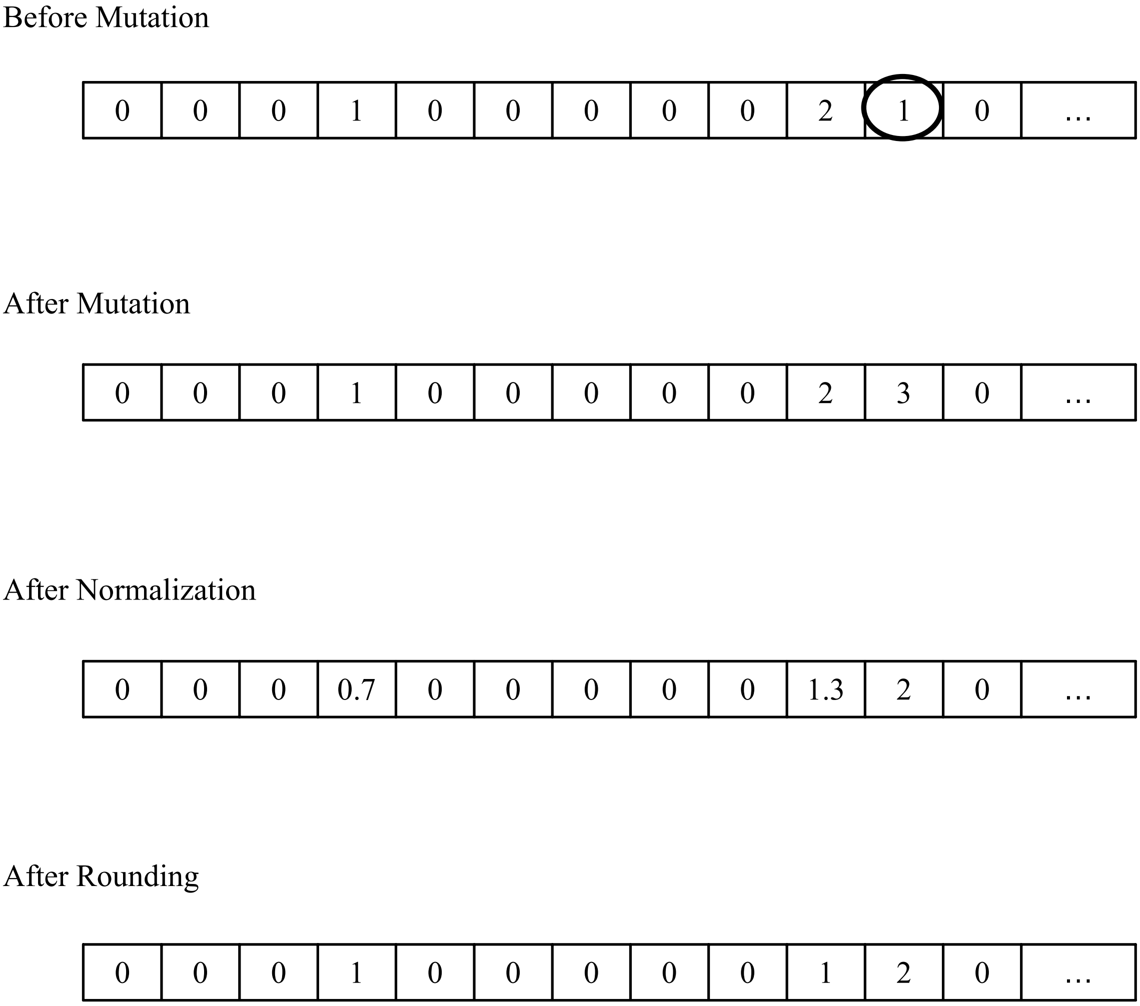
Figure II-15: An Example of Crossover Operation

The percentage of exchanged genes is called crossover ratio. Different crossover rates are tested in this program for tuning purposes.

Step 5: Mutation.

Some genes of some chromosomes are randomly selected and their values are changed. This will help the algorithm to avoid being trapped in local optima. The ratio of those selected to the total number of chromosomes is called the mutation ratio. A ratio range between 0.01-0.015 is tested. Fleet size constraint works again in mutation, and the normalization is implemented.

After crossover and mutation the pool of solutions is expanded. We use a ranking procedure to rank the chromosomes in the expanded pool. The chromosomes that have better fitness values will be kept and the others will be eliminated. The ranking is based on the evaluation value in step 2. An example is shown in Figure I-3.



Before Mutation

0	0	0	1	0	0	0	0	0	2	1	0	…

After Mutation

0	0	0	1	0	0	0	0	0	2	3	0	…

After Normalization

0	0	0	0.7	0	0	0	0	0	1.3	2	0	…

After Rounding

0	0	0	1	0	0	0	0	0	1	2	0	…

Figure II-16: An Example of Mutation Operation

Step 6: Verification of the stopping criterion.

As long as the convergence criterion has not been reached, Steps 2 through 5 are repeated to continue the next generation; otherwise the algorithm stops. The convergence criterion can be either a maximum limit on the number of iterations or maximum limit on the number of iterations during which no improvements are made, e.g. if no improvement is made in consecutive iterations, the algorithm stops.

The genetic algorithm method proposed above is able to seek the optimal EMS depot locations and the fleet assignments simultaneously for a given set of candidate depot locations.

II.3 Numerical Example

Example 1

Using the same simulation model described in Chapter 6, we test a trial network shown in Figure II-4 consists of 1757 nodes and 2144 links. We assume there are 16 medical service units available. This example is used to test the solution procedure.

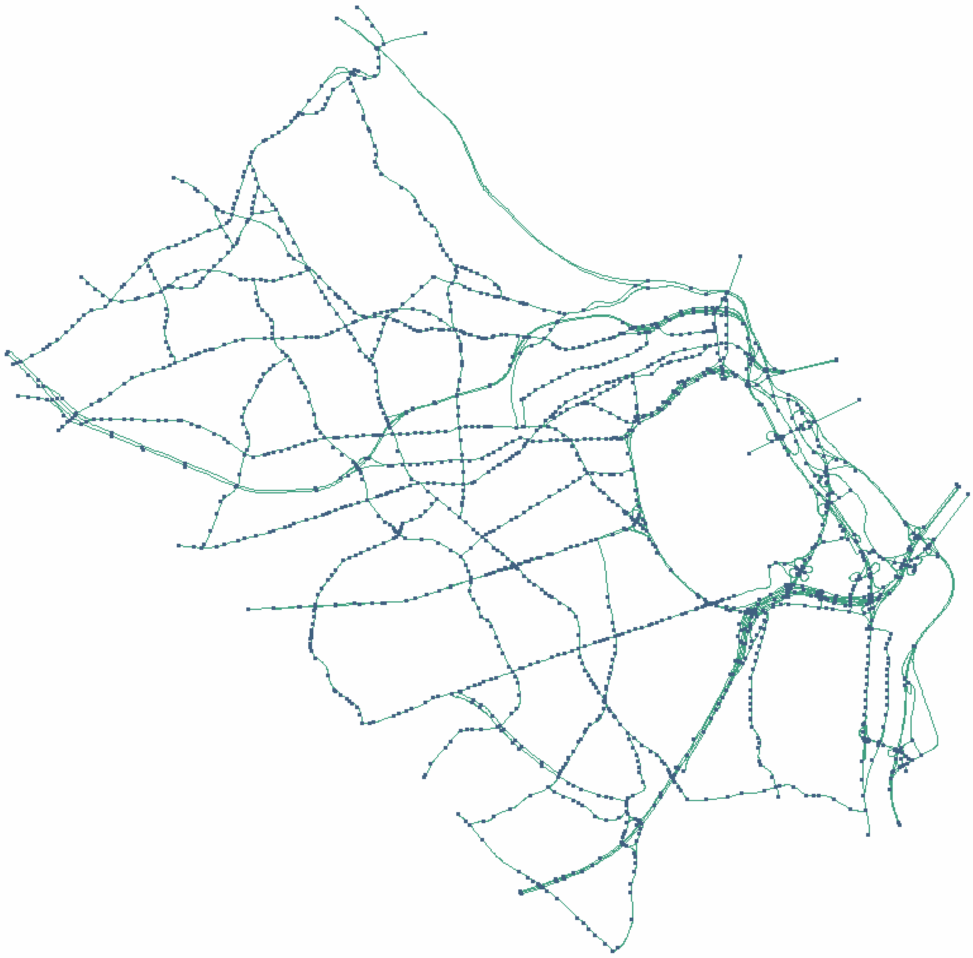


Figure II-17: The Sample Street Network

When no cost constraints are considered in the evaluation, the solution provides for 16 locations for all 16 medicate service units. These locations are very evenly distributed in the area, as shown as Figure II-5.

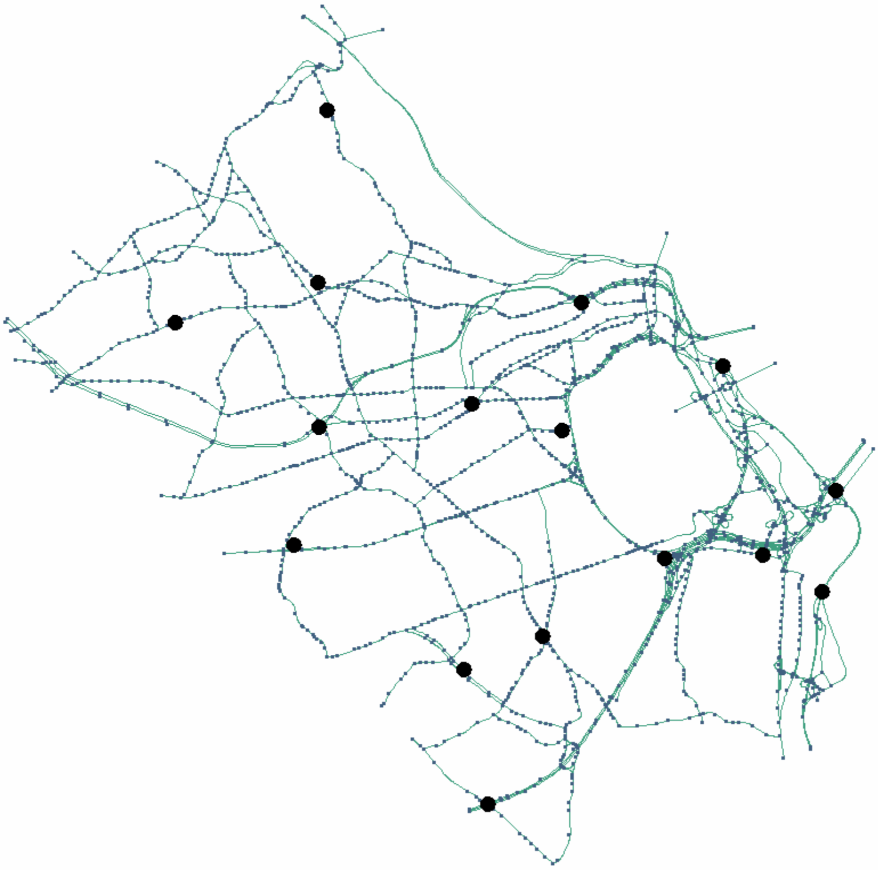


Figure II-18: Depot Locations without Cost Constraints

Example 2

The network shown in Figure II-6 is used in this example. This network consists of 5496 nodes and 7325 links and is modeled from an existing real network. The fleet consists of 16 medical service vehicles and there are 10 existing stations. Real operational data is analyzed to calibrate the parameters needed in the simulation.

In the street network that is shown as Figure II-6, the highlighted dots represent the current location of the EMS facilities. We applied the optimization approach discussed above to develop facility allocation plans for various cost scenarios.

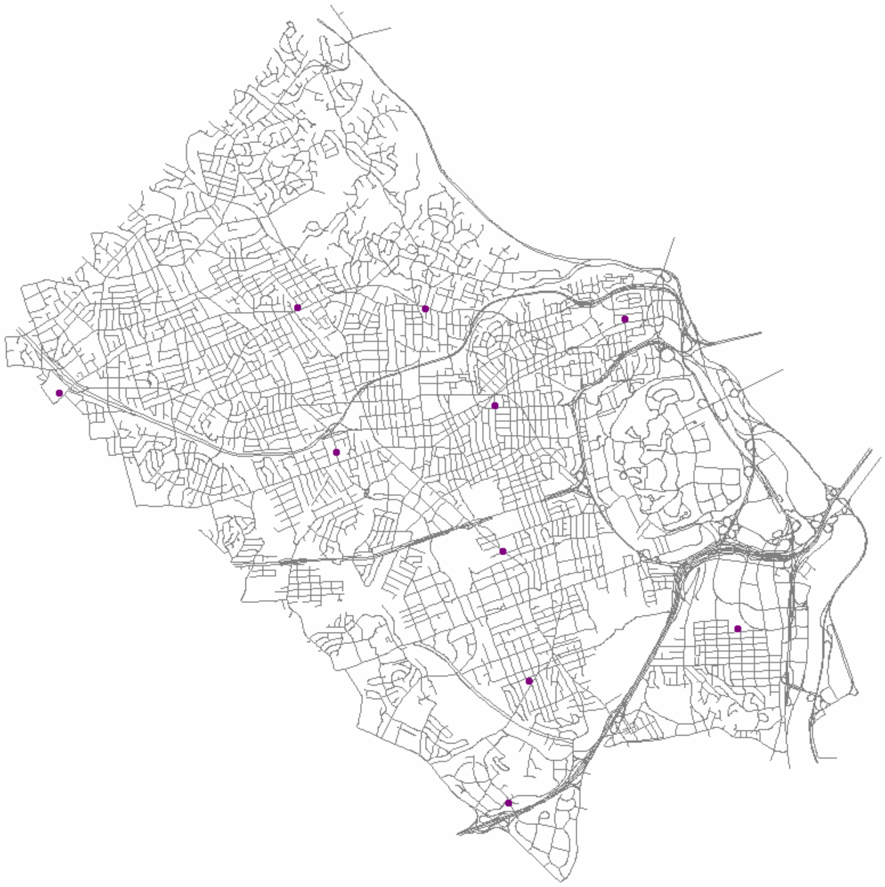


Figure II-19: A Large Street Network

In the evaluation function (II-1), when C_i and C_T change, the solutions vary. Figure II-7 shows the depot locations when cost constraints are considered and the time coefficient is set at 10. In Figure II-7, the smallest dots represent the nodes of the street network, which represent the demand distribution as well. The medium-sized dots represent the current location of depots and the largest dots represent the location of depots obtained from the heuristic solution. Similar to Figure II-5, when the evaluation function coefficient corresponding to the average response time is large, the average response time dominates the evaluation function and the locations of depot are evenly distributed according the density of nodes.

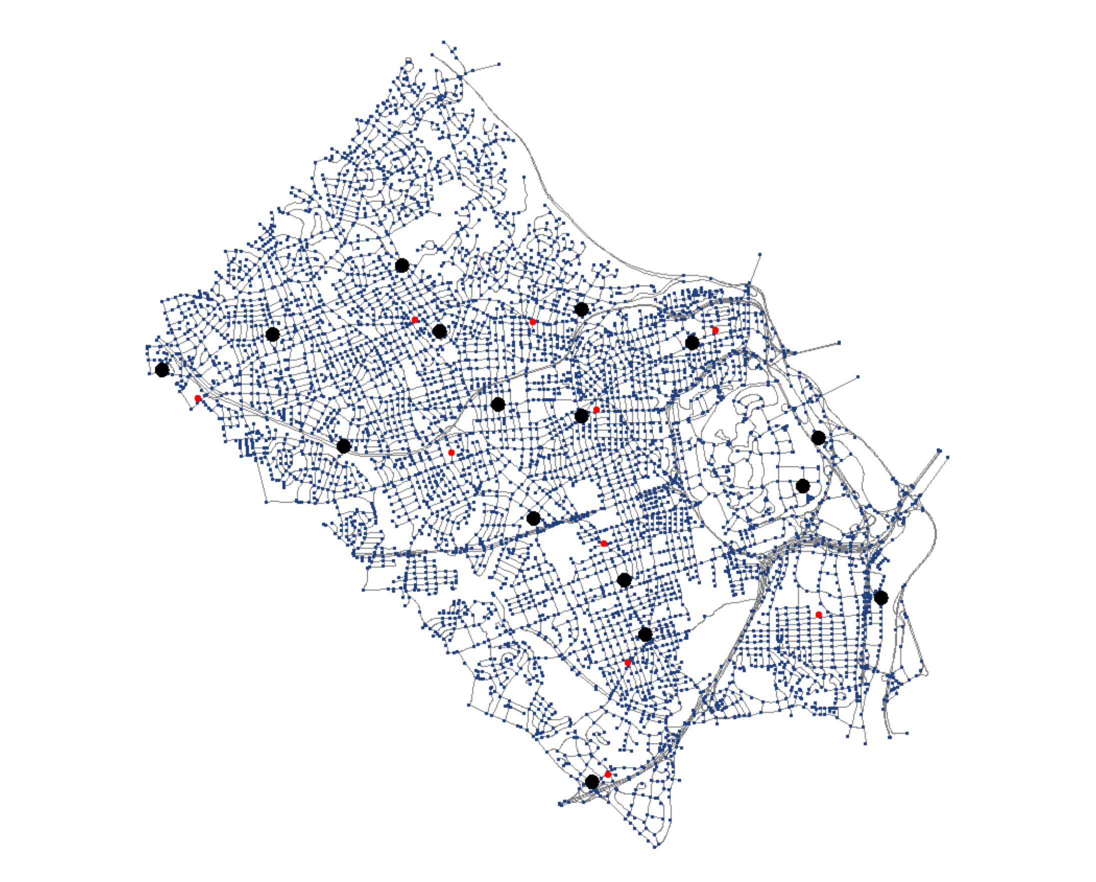


Figure II-20: Locations of Depot with Cost Constraint and Time Coefficient = 10

When the coefficient corresponding to the average response time decreases, the cost constraints result in choosing depot locations that are in more concentrated demand areas, as shown in Figures II-8 through II-11. One depot has at least 1 vehicle and it is labeled if the number of assigned vehicle to it is greater than 1.

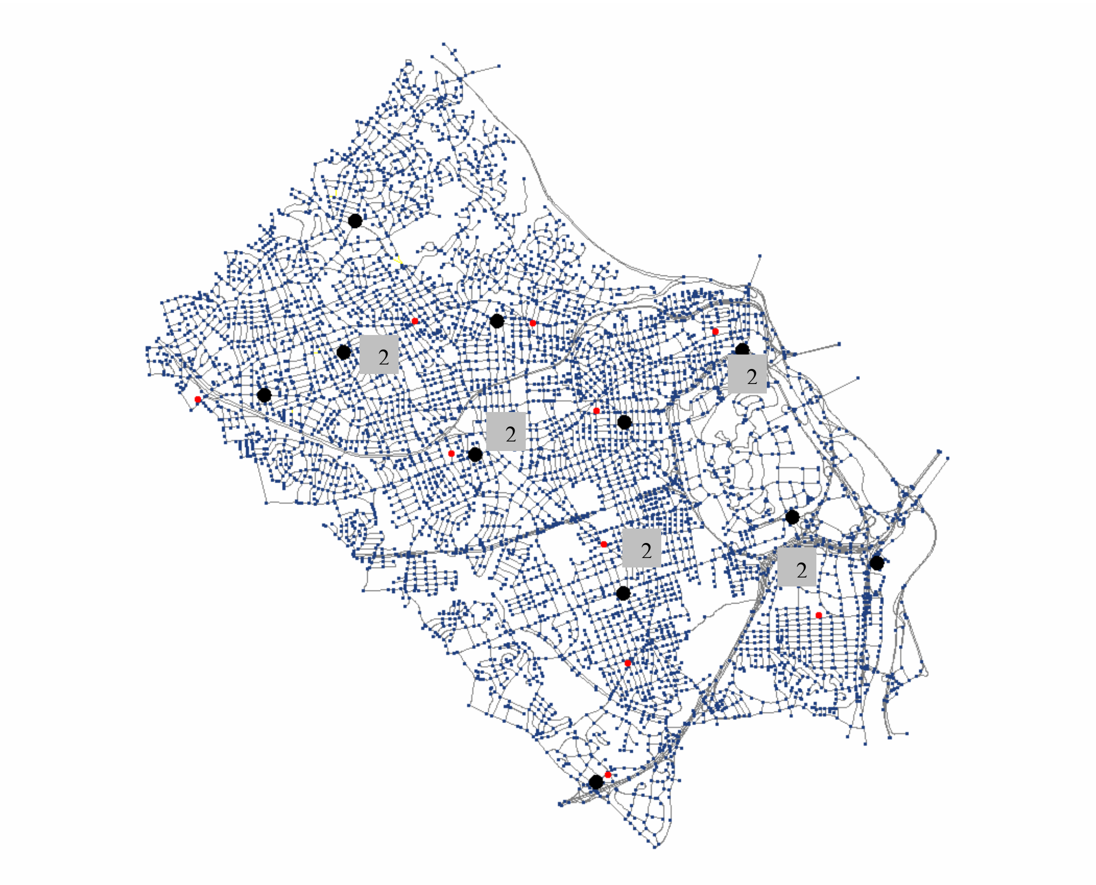


Figure II-8: Locations of Depot with Cost Constraint and Time Coefficient = 8

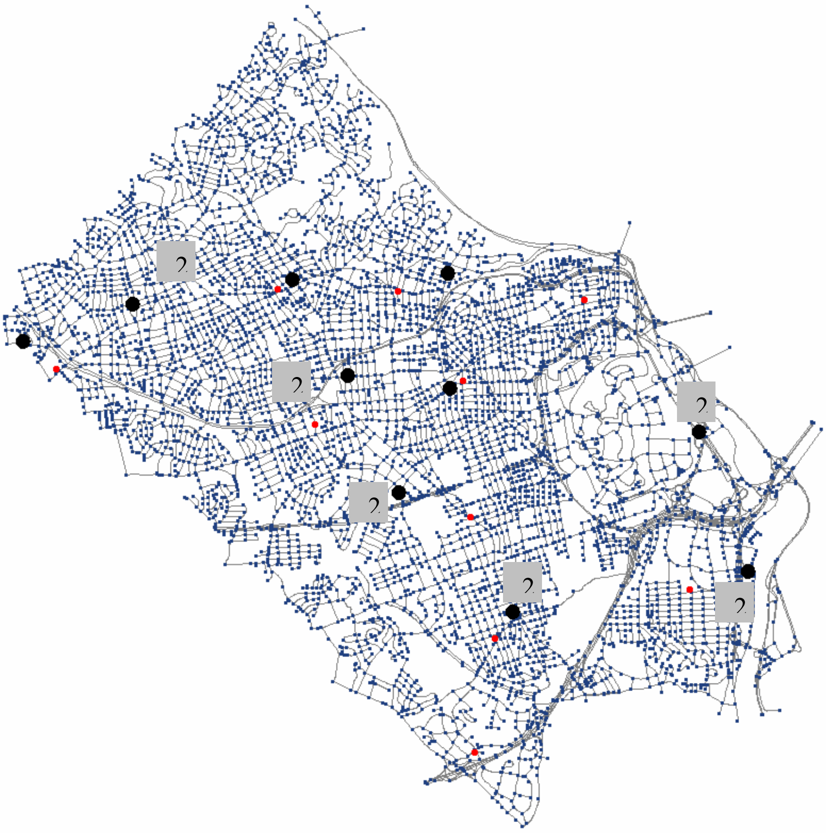


Figure II-21: Locations of Depot with Cost Constraint and Time Coefficient = 7

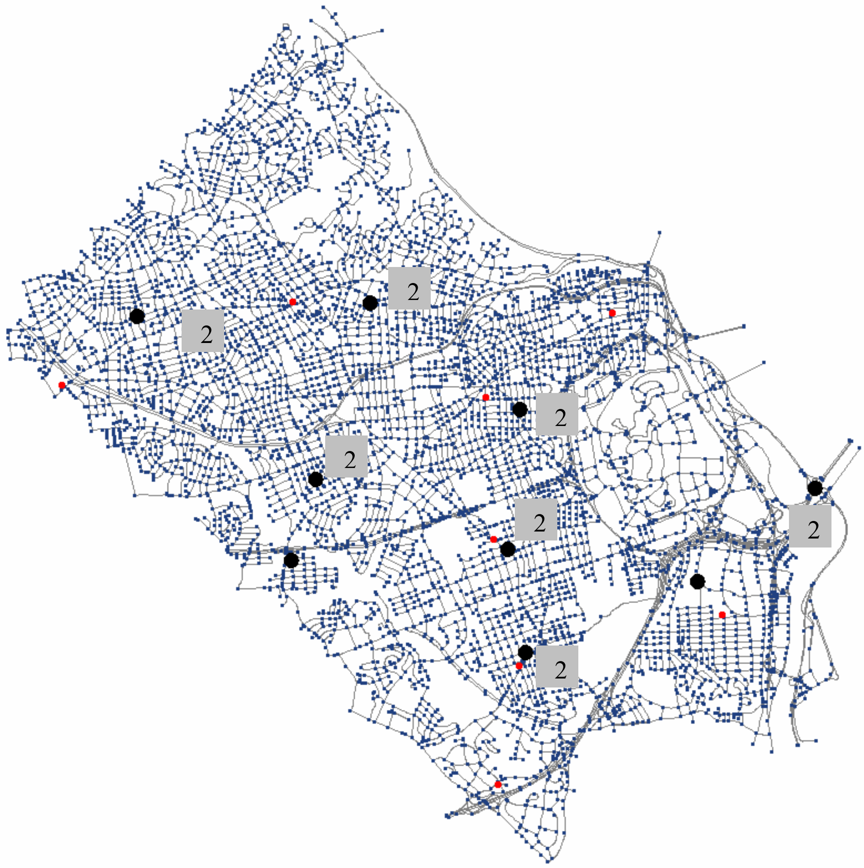


Figure II-22: Locations of Depot with Cost Constraint and Time Coefficient = 5

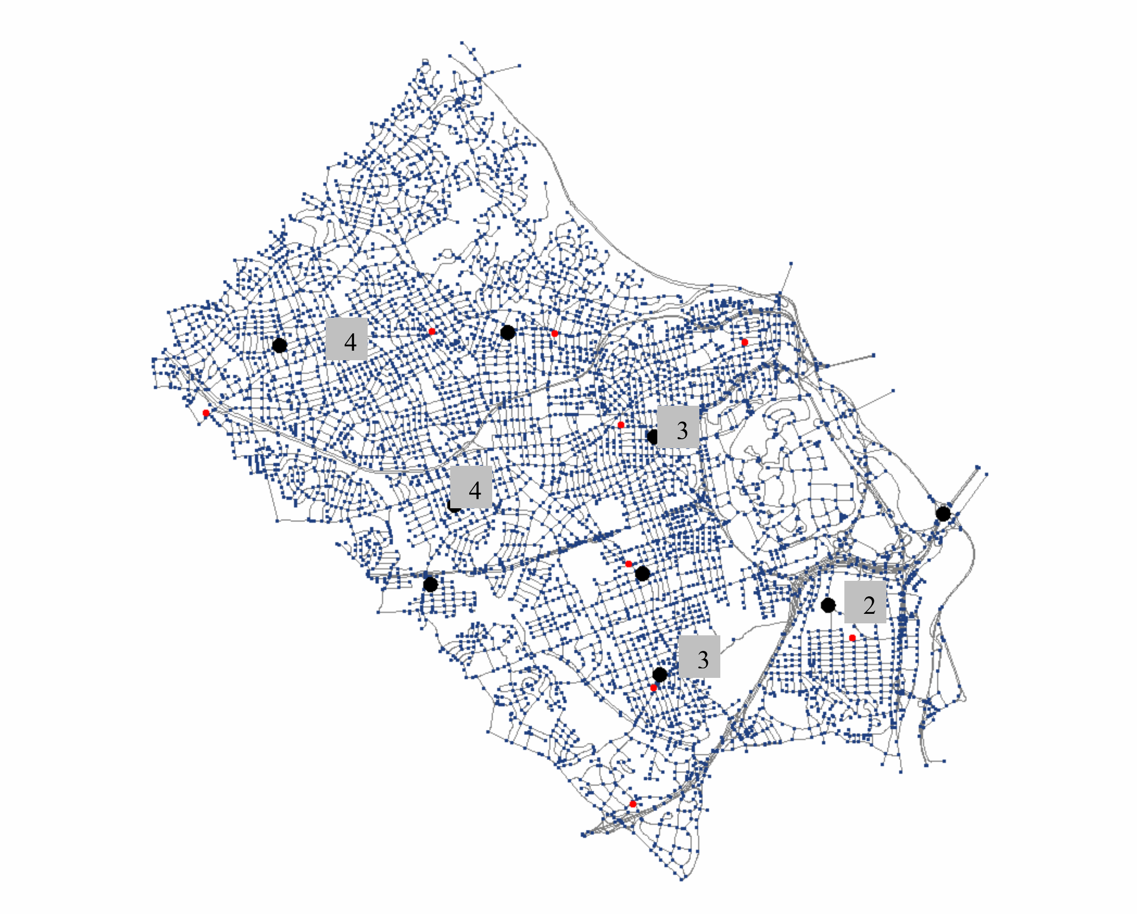


Figure II-23: Locations of Depot with Cost Constraint and Time Coefficient = 3

We compared the average response times resulting from the simulation of the emergency response operations under various depot location scenarios. Each simulation for different location configurations has 20 independent runs, and each run is on a 10-day basis. The 95 percent confidence interval of the Average Response Time for each location configuration is shown in Figure II-12. The results indicate that when the number of depots decreases, the average response time increases dramatically as expected.

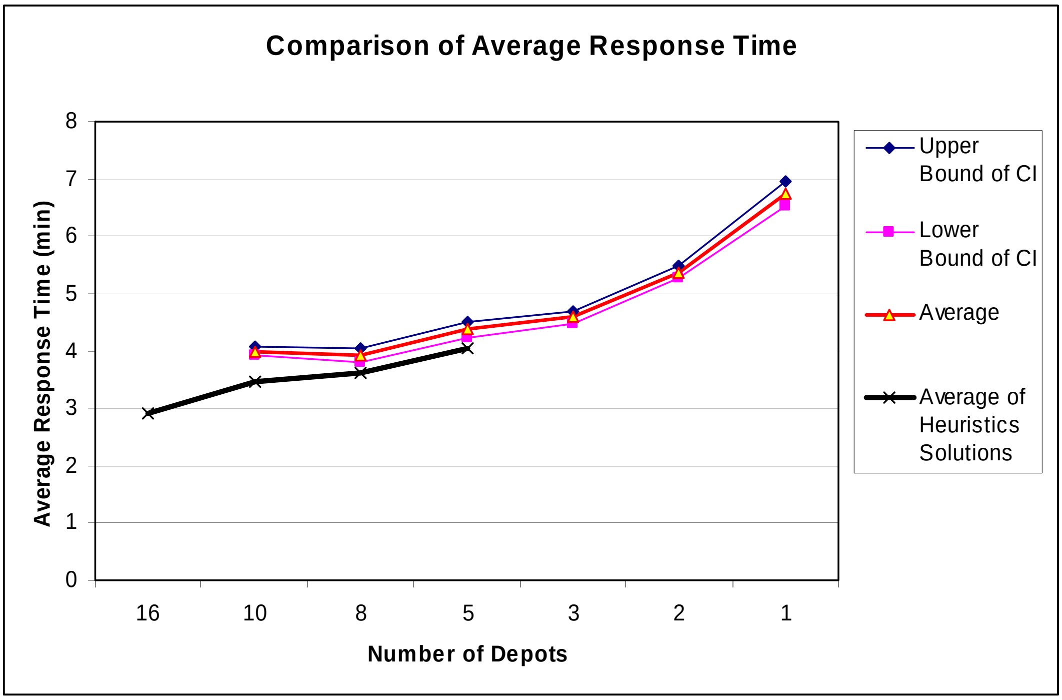


Figure II-24: Comparison of ART of Heuristic Solutions and Real Configurations

In this section, an approach was introduced to determine the locations of EMS facilities and their fleet assignment simultaneously. This approach provides the management the flexibility to select better locations and allocation plan under a known budget. A genetic algorithm was designed in which a simulation model is integrated as the tool for evaluation of solutions. The approach was tested in a real network and the results indicated that very good quality solutions are produced. This simulation system can easily be transformed for real-world applications and can assist in providing online dispatching and routing information when real-time traffic information becomes available. More research needs to be conducted on real-time travel time prediction, based on the real-time information regarding congestion, work zones, incidents and traffic conditions.

Bibliography

Al-Deek, H., D'Angelo, M., Wang, M., 1998, “Travel Time Prediction With Non-Linear Time Series”, Fifth International Conference on Applications of Advanced Technologies in Transportation Engineering.

Balas, E. and M. J. Saltzman, 1989, “Facets of the three-index assignment polytope”, *Discrete Applied mathematics*, Vol. 23, pp. 201-229.

Balas, E. and M. J. Saltzman, 1991, “An algorithm for the three-index assiggment problem”, *Operations Research*, Vol. 39, pp. 150-161.

Barker, J., Clayton, E., Taylor., B., 1989, “A Non-linear Multi-criteria Programming Approach for Determining County Emergency Medical Service Ambulance Allocations”, *Journal of Operational Research Society*, Vol.40, No.5, 423-432.

Barto, A., S. Bradtke and S. Singh, 1995, “Learning to act using real-time dynamic programming,” *Artificial Intelligence* Vol. 72, 81-138.

Batta, R. and N. Mannur, 1990, “Covering-Location Models for Emergency SituationsThat Require Multiple Response Units,” *Management Science*, Vol. **36**, 16-23.

Bell, M and K. I. Wong, 2005, “the Optimal Dispatching of Taxis: A Rolling Horizon Approach,” Proceeding of ISTTT 16, College Park, Maryland.

Ben-Akiva, M., Cascetta, E., Gunn, H., 1995, “An On-line Dynamic Traffic Prediction Model for An Inter-urban Motorway Network”, *Urban Traffic Networks: Dynamic Flow Modeling And Control.* 83-122

Benedict, J., 1983, “Three hierarchical objective models which incorporate the concept of excess coverage for locating EMS vehicles or hospitals,” M.S. Thesis, Northwestern University.

Berlin, G., 1972, “Facility location and vehicle allocation for provision of an emergency service,” Ph. D. Dissertation, the John Hopkins University, Baltimore, MD.

Berlin, G., Liebman, J., 1974, “Mathematical Analysis of Emergency Ambulance Location”, *Socio-Econ. Plan. Sci.*, Vol. 8, 323-328.

Berman, O., Larson, R.C. and Chiu, S.S., 1985, “Optimal server location on a network operating as an *M/G/*1 queue”. *Operations Research*, 33(4), 746–771.

Berman, O. and LeBlanc, B., 1984, “Location-relocation of mobile facilities on a stochastic network”. *Transportation Science*, **18**(4), 315–330.

Berman, O., Larson, R., 1985, “Optimal 2-Facility Network Districting in the Presence of Queuing”, *Transportation Science*, Vol. 19, No.3 261-277.
Berman, O., Larson, R., Parkan, C., 1987, “The Stochastic Queue p-Median Problem”, *Transportation Science*, Vol. 21, No.3, 207-216.

Bertsekas, D. P., 1981, “A New Algorithm For the Assignment Problem” *Mathematical Programming,* Vol. 21, 152-171.

Bertsekas, D. P. and J. N. Tsitsiklis, 1991, “An Analysis of Stochastic Shortest Path Problems,” *Mathematics of Operations Research*, Vol. 16, 580-595.

Bowman, R., 1997, “Deaths Expected from Delayed Emergency Response Due to Neighborhood Traffic Mitigation,” http://members.aol.com/raybowman/risk97/eval1.html.

Brown, G.G. and R. McBride, 1985, “Solving generalized networks”, *Management Science*, Vol. 20, 1497-1523.

Burkard, R. E. and R. Rudolf, 1993, “Computational investigations on 3-dimensional axial assignment problems”, *Belgian J. of Operations Research*, Vol. 32, 85-98.

Burkard, R. E.R. Rudolf, and G. J. Woeginger, 1996, “Three dimensional axial assignment problems with decomposable cost coefficients”, *Discrete Applied Mathematics*, Vol. 65, 123-169.

Burkard, Rainer and Eranda Cela, 1998, “Linear Assignment Problems and Extenstions,” Technique Report.

Carter, G. and Ignall, E., 1970, “A Simultion Model of Fire Department Operations,” *IEEE System Science and Cybernetics*, Vol. 5, 282-293.

Carson, Y.M. and Batta, R., 1990, “Locating an ambulance on the Amherst Campus of the State University of New York at Buffalo”. *Interfaces*, **20**(5), 43–49.

Catrysse, D. and Van Wassenhove, L.N, 1992, “A survey of algorithms for the Generalized Assignment Problem”, *European Journal of Operational Research,* Vol. 60, 260-272.

Chabini, I., 1998, “Discrete Dynamic Shortest Path Problems in Transportation Application”, *Transportation Research Record*, No. 1645, 170-175.

Chaiken, J., Larson, R. 1998, “Methods for allocating urban emergency units: a survey”. Management Science, 19, 110-130.

Chang, E., 1999, “Traffic Estimation for Proactive Freeway Traffic Control”, *Transportation Research Record*, No.1679, 81-86.

Chang, M. F., and D. C. Gazis, 1975, “Traffic Density estimation with consideration of lane changing”, *Transportation Science*, Vol. 9, No. 4, 308-320.

Chapman, S. C. and J. A. White, 1974, “Probabilistic formulations of emergency service facilities location problems,” ORSA/TIMS Conference, San Juan, Puerto Rico.

Chelst, K., Jarvis, J., 1979, “Estimating the Probability Distribution of Travel Times for Urban Emergency Service Systems”, *Operations Research*, Vol.27, No.1, 199-204.

Chelst, K. and Z. Barlach, 1981, “Multiple Unit Dispatching in Emergency Services: Models to Estimate System Performance,” *Management Science,* Vol. 27, No. 12, 1390-1409.

Cherkassky, B. V., A. V. Goldberg and T. Radzik, 1993, “Shortest paths algorithms: theory and experimental evaluation,” Technical Report 93-1480, Computer Science Department, Stanford University.

Christofides, N. and P. Viola, 1971, “The optimum location of multicenters on a graph,” *Operations Research Quarterly*, 145.

Chu, P.C. and J. E. Beasley, 1997, “A genetic algorithm for the generalized assignment problem”, *Computers and Operations Research*, 17-23.

Church, R. L. and C. Revelle, 1974, “The maximal covering location problem,” *Papers of the Regional Science Association,* Vol. 32 , 101-118.

Cooke, K., Halsey, E., 1966, “The Shortest Route Through a Network with Time-Dependent Internodal Transit Times”, *Journal of Mathematical Analysis and Applications*, Vol. 14, 493-498.

Corea, G. A. and V. G. Kulkarni, 1993, "Shortest paths in stochastic networks with arc lengths having discrete distribution," *Networks*, Vol. 23, No. 3, 175-183.

Cote, A. E., 1997, “Fire Protection Handbook (18th Edition)”, National Fire Protection Association.

Commission on Fire Accreditation International. 2000. *Fire & Emergency Service Self-Assessment Manual* (6th Ed.). Fairfax, Virginia: Commission on Fire

Accreditation International (p. Section 3-43)

Cragg, C. A., and M. J. Demetsky, 1995, "Final report: simulation analysis of route diversion strategies for freeway incident management," VTRC 95-R11, Traffic Research Advisory Committee, FHWA, USDOT.

Daskin, M., 1983, "A maximum expected covering location model formulation, properties and heuristic solution." *Transportation Science*, Vol. 17, 48-70.

Deo, N., Pang, C., 1984, "Shortest-Path Algorithms: Taxonomy and Annotation", *Networks*, Vol. 14, 275-323.

Dreyfus, S., 1969, "An Appraisal of Some Shortest-Path Algorithms" *Operations Research*, Vol. 17, 395-412.

Drezner, Zvi (editor), 1997, "Facility Location, a survey of applications and methods", Springer-Verlag Berlin Heidelberg NewYork.

Drezner, Zvi and Horst W. Hamacher (editor), 2002, "Facility Location, applications and theory", Springer-Verlag Berlin Heidelberg NewYork.

Eaton, D., M. Hector, V. Sanchez, R. Lantigua and J. Morgan, 1986, "Determine ambulance deployments in Santo Domingo, Dominican Republic," *Journal of the Operational Research Society*, 113.

Eldor, M., 1977, "Demand predictors for computerized freeway control systems", Proceedings of the 7th International Symposium on Transportation and Traffic Theory, Kyoto, Japan, 341-358.

Frank, H., 1969, "Shortest paths in probabilistic graph", *Operations Research*, Vol. 17-4, 583-599.

Fisher, M.L. R. Jaikumar and L.N. Wassenhove, 1986, "A multiplier adjustment method for the generalized assignment problems", *Management Science*, Vol. 32, 1095-1103.

Fisher, M. L., 1985, "An applications oriented guide to Lagrangian relaxation", *Interface*, Vol. 15, 10-21.

Fitzsimmons, J., 1973, "A Methodology for Emergency Ambulance Deployment", *Management Science*, Vol. 19, No. 6, 627.

Gafarian, A.V., J. Paul, and T. L. Ward, 1977, "Discrete time series models of a freeway density process", Proceedings of the 7th International Symposium on Transportation and Traffic Theory, Kyoto, Japan, 387-411.

Gallo, G. and Pallottino, S. 1988, “Shortest path algorithms". *Annals of Operations Research* Vol. 13, 3-79.

Gass, S., 1983, “Decision-aided models: Validation, assessment, and related issues for policy analysis”, *Operations Research*, No. 31, 603-631.

Gazis, D., Knapp, C., 1971, “On-line Estimation of Traffic Densities from Time-Series of Flow and Speed Data”, *Transportation Science*, Vol. 5, No. 3, 283-301.

Gendreau, Michel, Gilbert Laporte and F. Semet, 1999, “Parallel Tabu Search Heuristic for Real-time Vehicle Routing and Dispatching,” *Transportation Science* 33: 381-390.

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

Gendreau M, Laporte G, Semet F, 2006. “The maximal expected coverage relocation problem for emergency vehicles”. *J Opl Res Soc*, 57: 22–28.

Ghiani, G., F. Guerriero, G. Laporte and R. Musmanno (2003) Real-time vehicle routing:Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, 151, 1-11.

Goldberg, J., Dietrich, R., Chen, J., Mitwasi, M., 1990, “Validating and Applying a Model for Locating Emergency Medical Vehicles in Tucson, AZ”, *European Journal of Operational Research*, No.49, 308-324.

Glover, F., 1997, "Tabu Search,” Kluwer Academic Publishers, Boston.

Goldberg, J. and Szidarovszky, F., 1991, "Method for solving nonlinear equations used in evaluating emergency vehicle busy probabilities," *Operations Research,* Vol. 39, 903-916.

Goldberg, J., Paz, L., 1991, “Locating Emergency Vehicle Bases when Service Time Depends on Call Location”, *Transportation Science*, Vol. 25, No.4, 264-280.

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

Hadas, Y. Ceder, A., 1996, “Shortest Path of Emergency Vehicle Under Uncertain Urban Traffic Conditions”, *Transportation Research Record*, No.1560, 34-39.

Haghani, A, 1996, "Capacitated Maximum Covering Location Models: Formulations and Solution Procedures*", Journal of Advanced Transportation*, Vol. 30, No.3, 101-136.

Haghani, A., H. Hu, and Q. Tian, 2003, "An Optimization Model for Real-Time Emergency Vehicle Dispatching and Routing", Presented at the 82nd annual meeting of the Transportation Research Board, Washington, D.C.

Haghani, A., Q. Tian, and H. Hu, 2003, "A Simulation Model for Real-Time Emergency Vehicle Dispatching and Routing", Presented at the 82nd annual meeting of the Transportation Research Board, Washington, D.C.

Hakimi, S., 1964, "Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph", *Operations Research*, Vol. 12, 450-459.

Hakimi, S., 1965, "Optimum Distribution of Switching Centers in a Communications Network and Some Related Graph Theoretic Problems", *Operations Research*, Vol. 13, 462-475.

Hall, R., 1986, "The Fastest Path through a Network with Random Time-Dependent Travel Times", *Transportation Science*, Vol. 20, No. 3, 182-188.

Halpern, J., 1977, "The accuracy of Estimates for the Performance Criteria in Certain Emergency Service Queuing Systems", *Transportation Science*, Vol.11, No.3, 223-241.
Hansen, P. and L. Kaufman, 1973, "A primal-dual algorithm for the three-dimensional assignment problem", Cahiers du CERO, Vol. 15, 327-336.

Heller, M., J. Cohon and C. ReVelle, 1989, "The use of simulation in validating a multi-objective EMS location model", *Annuals of Operational Research*, Vol. 18, 303-322.

Hoffman, C. and Janko, J., 1988, "Travel time as a basis of the LISB guidance strategy," *Proceedings of IEEE Road Traffic Control Conference,* IEEE, New York, 6-10.

Hogan, K. and C. ReVelle, 1986, "Concepts and applications of backup coverage," *Management Science*, Vol. 32, 1434-1444.

Huisken, G., 2003, "Soft-computing techniques applied to short-term traffic flow forecasting," *Systems Analysis Modelling Simulation*, Vol.43-2, 165-173.

http://www.capwinproject.com/

Ignall, E.D., P. Kolesar, and W.E. Walker, 1978, "Using Simulation To Develop and Validate Analytic Models: Some Case Studies", *Operations Research*, Vol. 26, No. 2, pp. 237-253.

Jornsten, K. and Bjorndal, M., 1994, “Dynamic location under uncertainty”. *Studies in Regional and Urban Planning*, **3**, 163–184.

Karp, R. M., 1980, “An algorithm to solve the m \Theta n assignment problem in expected time O(mn log n)”, *Networks,* Vol. 10, 143-152.

Kaysi, I., M. Ben-Akiva and H. Koutsopulos, 1993, “An Integrated Approach to Vehicle Routing and Congestion Prediction for Real-Time Driver Guidance,” *Transportation Research Record*, Vol. 1408, 66-74.

Kolesar, P. and Walker W. E., 1974, “An algorithm for the dynamic relocation of fire companies,” Operations Research 22: 249–274.

Kuhn, H. W. 1955, “The Hungarian Method for the Assignment Problem”. *Naval Research Quarterly* 2, 83.

Kulkarni, V., 1986, “Shortest Paths in Networks with Exponentially Distributed Arc Lengths,” *Networks* Vol.16, 255 –274.

Larson, R., 1974, “A Hypercube Queuing Model for Facility Location and Redistricting in Urban Emergency Services”, *Comput. & Ops. Res.*, Vol. 1, 67-95.

Larson, R., 1975, “Approximating the Performance of Urban Emergency Service Systems”, *Operations Research*, Vol.23, No.5, 845-868.

Law, A.M. and W. D. Kelton, 2000, “Simulation Modelling and Analysis (Third Edition)”, McGraw-Hill International Series-Industiral Engineering Series, McGraw-Hill higher Education, USA.

Lorena, L.A.N. and M.G. Narciso, 1996, “Relaxation heuristics for a generalized assignment problem”, *European Journal of Operational Research*, Vol. 19, No. 3, 600-610.

Lorena, L.A.N., M.G. Narciso and J. E. Beasley, 2002, “A constructive genetic algorithm for the generalized assignment problem”, *working paper.*

Lubicz, M., Mielczarek, B., 1987, “Simulation Modelling of Emergency Medical Services”, *European Journal of Operational Research*, No. 29, 178-185.

Marinov and Revelle,C, 1995, “Siting Emergency Services, in Facility Location: A Survey of articles, applications and methods”, Drezner, Z(editor), Springer Series in Operations Research, 199-222.

Miller-Hooks, E., 1996, “Optimal Routing in Time-Varying, Stochastic Networks: Algorithms and Implementation,” Ph.D. Dissertation, University of Texas, Austin.

Miller-Hooks, E. and H. Mahmassani, 1998, “Least possible time paths in stochastic time-varying networks,” *Computers and Operations Research*, Vol. 25-12, 1107-1125.

Minieka, E., 1970, “The m-center problem,” *SIAM Review*, Vol. 12, 138-139.

NFPA 1710, Standard for the Organization and Deployment of Fire Suppression Operations, Emergency Medical Operations, and Special Operations to the Public by Career Fire Departments, 2001 Edition, 1 Batterymarch Park, PO Box 9101, Quincy, MA.

Nicholson H. and C. D. Swann, 1974, “The prediction of traffic flow volumes based on spectral analysis”, *Transportation Research Record*, Vol. 8, 533-538.

Nulty, W. G. and M. A. Trick, 1988, “GNO/PC generalized network optimization system”, *O.R. Letters,* Vol. 2, 101-112.

Osman, I.H.. 1995, “Heuristics for the generalized assignment problem, simulated annealing and tabu searcha pproaches,” OR Specktrum.

Park, D., Rilett, L., 1998, “Forcasting Multiple-Period Freeway Link Travel Times Using Modular Neural Networks”, *Transportation Research Record*, No.1617, 163-170.
Piershalla, W. P., 1967, “The multidimensional assignment problem”, *Operations Research*, Vol. 16, pp. 422-431.

Piershalla, W. P., 1968, “The tri-substitution method for the three-multidimensional assignment problem”, *Canadian ORS Journal*, Vol. 5, pp. 71-81.

Pirkul, H., Schilling, D., 1988, “The Siting Emergency Service Facilities With Workload Capacities and Backup Service”, *Management Science*, Vol. 34, No.7, 896-908.

Psaraftis, H., and J. N. Tsitsiklis, 1993, “Dynamic Shortest Path in acyclic Networks with Markovian Arc Costs”, *Operations Research*, Vol. 41, No. 1, 91-101.

Rao, A, 1974, “Counterexample for the location of emergency service facilities”, *Operations Research*, Vol. 22-6, 1259-1261.

Regan, A., Mahmassani, H., Jaillet, P., 1998, “Evaluation of Dynamic Fleet Management Systems: Simulation Framework”, *Transportation Research Record*, No. 1645.

Revelle, C., 1997, “A Perspective on Location Science”, *Location Science*, Vol. 5, No.1, 3-13.

Revelle, C., 1989, “Extension and Prediction in Emergency Service Siting Models”, *European Journal of Operational Research*, Vol. 40, 58-69.

Revelle, C. and K. Hogan, 1989, “The maximum reliability location problem and α-reliable p-center problem: derivatives of the probabilistic location set covering problem,”, *Annuals of Operations Research*, 155-174.

Ross, G. T. and M. S. Soland, 1975, “A branch and bound algorithm for the generalized assignment problem”, *Mathematical Programming*, Vol. 8 91-103.

Sathe, Aamod and Miller-Hooks, Elise, 2005, “Optimizing Location and Relocation of Response Units in Guarding Critical Facilities”, *Transportation Research Record,* 1923, pp. 127-136.

Savas, E., 1969, “Simulation and Cost-effectiveness Analysis of New York’s Emergency Ambulance Service”, *Management Science*, Vol. 15, No.12, B608-B627.

Schilling, D. A., J. Vaidyanathan and R. Barkhi, 1993, “A review of covering problems in facility location,” *Location Science*, Vol. 1, 25-55.

Schilling, D. A., D. Elzinga, J. Cohon, R. Church and C. Revelle, 1979, “The TEAM/GLEET models for simultaneous facility and equipment siting,” *Transportation Science*, 167.

Shantikumar, J.G., and R.G. Sargent, 1983, “A unifying view of hybrid simulation/analytic models and modeling”, *Operations Research*, Vol. 31, No. 6, pp. 1030-1052.

Sivanandan, R., Hobeika, A., Ardekani, S., Lockwood, P., 1987, “A Heuristic Shortest-Path Method for Emergency Vehicle Assignment-A Study on the Mexico City Network”, *Transportation Research Record*, No. 1168, 86-91.

Smith, B., Demetsky, M., 1995, “Short-Term Traffic Flow Prediction: Neural Network Approach”, *Transportation Research Record*, No. 1453, 98-104.

Stephanedes, Y. J., P. G. Michalopoulos, and R. A. Plum, 1981, “Improved estimation of Traffice Flow for Real-Time Control,” *Transportation Research Record*, Vol. 795, 28-39.

Sutton, R. and A. Barto, 1998, “Reinforcement Learning: An Introduction,” MIT Press.

Tian, Q., 2002, "A Simulation Approach for Real-Time Emergency Vehicle Dispatching", Master's Thesis, University of Maryland, College Park, MD.

Toregas, C., Swain, R., ReVelle, C., Bergman, L., 1971, "The Location of Emergency Service Facilities", *Operations Research*, Vol. 19-6, 1363-1373.

Toregas, C., Swain, R., ReVelle, C., Bergman, L., 1974, "Reply to Rao's note on the location of emergency service facilities," *Operations Research*, Vol. 22-6, 1262-1267.

Trick, M. A., 1992, "A linear relaxation heuristic for the generalized assignment problem", *Naval Research Logistics*, Vol. 39, 137-152.

Weintraub A, Aboud, J. Fernandez, C., Laporte, G. And Ramirez, E., 1999. "An emergency vehicle dispatching system for an electric utility in Chile". *J Opl Res Soc,* 50: 690–696.

Ziliaskoulos, A., Mahmassani, H., 1993, "Time Dependent, Shortest-Path Algorithm for Real-Time Intelligent Vehicle Highway System Applications", *Transportation Research Record*, No. 1408, 94-100.

Zografos, K., Douligeris, C., Lin. C, 1993, "Model for Optimum Deployment of Emergency Repair Trucks: Application in Electric Utility Industry", Transportation *Research Record*, No. 1358.

Zogfrafos, K., Douligers, C., Lin, C., 1995, "Simulation Model for Evaluating the Performance of Emergency Response Fleets", *Transportation Research Record*, No. 1452.