5-2014

A mathematical programming approach for dispatching and relocating EMS vehicles.

Farshad Majzoubi 1983-
University of Louisville

A MATHEMATICAL PROGRAMMING APPROACH FOR DISPATCHING AND RELOCATING EMS VEHICLES

By

Farshad Majzoubi
M.S. in Systems Engineering , Sharif University, 2009
B.S in Mechanical Engineering, Tehran Polytechnic, 2006

A Dissertation
Submitted to the Faculty of the
J. B. Speed School of Engineering. of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Industrial Engineering
University of Louisville
Louisville, Kentucky

May 2014

A MATHEMATICAL PROGRAMMING APPROACH FOR DISPATCHING AND RELOCATING EMS VEHICLES

By

Farshad Majzoubi
M.S. in Systems Engineering , Sharif University, 2009
B.S in Mechanical Engineering, Tehran Polytechnic, 2006

A Dissertation Approved on

March 21, 2014

by the following Dissertation Committee:

_____

Dr. Sunderesh S. Heragu, Dissertation Co-Director

_____

Dr. Lihui Bai, Dissertation Co-Director

_____

Dr. John S. Usher

_____

Dr. Mahesh Gupta

ACKNOWLEDGEMENTS

First and foremost, I acknowledge my advisors Dr. Sunderesh Heragu and Dr. Lihui Bai for their enormous support in this research and giving me the opportunity to work on a problem which I had been enthusiastic about and also giving me the chance to implement it in a real world application. They have graciously guided me in all steps of the project.

I thank Dr. John Usher and Dr. Mahesh Gupta for serving on my committee and for their valuable feedbacks. I also thank the rest of the faculty and staff of the Department of Industrial Engineering at the University of Louisville.

And most importantly, I acknowledge my wife for supporting and encouraging me before and after marriage. I would like to appreciate my father and mother for simply everything particularly for inspiring me to be a responsible individual and my sister for her encouragement and support.

ABSTRACT

A MATHEMATICAL PROGRAMMING APPROACH FOR DISPATCHING AND
RELOCATING EMS VEHICLES

Farshad Majzoubi

March 21, 2014

We consider the problem of dispatching and relocating EMS vehicles during a pandemic outbreak. In such a situation, the demand for EMS vehicles increases and in order to better utilize their capacity, the idea of serving more than one patient by an ambulance is introduced. Vehicles transporting high priority patients cannot serve any other patient, but those transporting low priority patients are allowed to be rerouted to serve a second patient. We have considered three separate problems in this research. In the first problem, an integrated model is developed for dispatching and relocating EMS vehicles, where dispatchers determine hospitals for patients. The second problem considers just relocating EMS vehicles. In the third problem only dispatching decisions are made where hospitals are pre-specified by patients not by dispatchers.

In the first problem, the objective is to minimize the total travel distance and the penalty of not meeting specific constraints. In order to better utilize the capacity of ambulances, we allow each ambulance to serve a maximum of two patients. Considerations are given to features such as meeting the required response time window for patients, batching non-critical and critical patients when necessary,

ensuring balanced coverage for all census tracts. Three models are proposed- two of them are linear integer programing and the other is a non-linear programing model. Numerical examples show that the linear models can be solved using general-purpose solvers efficiently for large sized problems, and thus it is suitable for use in a real time decision support system.

In the second problem, the goal is to maximize the coverage for serving future calls in a required time window. A linear programming model is developed for this problem. The objective is to maximize the number of census tracts with single and double coverage, (each with their own weights) and to minimize the travel time for relocating. In order to tune the parameters in this objective function, an event based simulation model is developed to study the movement of vehicles and incidents (911 calls) through a city. The results show that the proposed model can effectively increase the system-wide coverage by EMS vehicles even if we assume that vehicles cannot respond to any incidents while traveling between stations. In addition, the results suggest that the proposed model outperforms one of the well-known real time repositioning models (Gendreau et al. (2001)).

In the third problem, the objective is to minimize the total travel distance experienced by all EMS vehicles, while satisfying two types of time window constraints. One requires the EMS vehicle to arrive at the patients' scene within a pre-specified time, the other requires the EMS vehicle to transport patients to their hospitals within a given time window. Similar to the first problem, each vehicle can transport maximum two patients. A mixed integer program (MIP) model is developed for the EMS dispatching problem. The problem is proved to be NP-hard, and a simulated annealing (SA) method is developed for its efficient solution. Additionally, to obtain lower bound, a column generation method is developed. Our numerical results show that the proposed SA provides high quality solutions whose objective is close to the obtained lower bound with much less CPU time. Thus, the SA method is suitable for implementation in a real-time decision support system.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

A Introduction

Emergency Medical Services (EMS) is a vital component of the infrastructure that provides public safety in a city. EMS personnel serve nearly 20 million patients a year in the United States (Armstrong et al., 2006). The primary goal of EMS is to use scarce resources, physical and human, to provide critical emergency services effectively and efficiently. Of the total $5 billion EMS expenditure, over $2.5 billion is for transportation of patients in the medical program (Sayre et al., 2001). This shows that one of the most important issues in EMS is the management of the vehicles. These resources need to be used efficiently to provide the best possible response in an emergency situation. This has therefore attracted the attention of operations researchers to develop models to optimize the usage of EMS vehicles. Much research has taken place since the 1960s on EMS vehicle location and deployment.

We briefly list the types of decisions made by EMS management. Upon receiving a call at the emergency call center, the following steps are undertaken.

1. The severity of call is determined and a priority is given to the call.

2. A decision is made about specific vehicle(s) that must be dispatched.

3. Vehicles are sent to the emergency scene.

4. Vehicles arrive at the scene and provide services.

5. After completion of service and/or transport to the hospital the vehicle becomes idle and proceeds to a predetermined location to wait for the next call.

Basically, there are three levels of decision making in management of EMS systems.

Strategic decisions are concerned with choosing location of fixed EMS stations and purchasing equipment and facilities.

Tactical decisions are about specifying the number of vehicles needed in a particular shift and scheduling personnel.

Operational decisions involve determining the specific set of vehicles to be dispatched, the set of locations they should be sent to after service and also procedures by paramedical staff

B   Problem Statement

This study considers the operation of EMS vehicles during a medical surge when the demand for EMS vehicles increases significantly and thus many available vehicles become busy. The national EMS Research Agenda (Sayre et al., 2001) has reported that a majority of the patients being treated by EMS do not have a major injury. Hence, a large portion of EMS budget is assigned to serving patients with low priorities who do not have life threatening conditions. We can further increase the system capacity, by allowing vehicles to serve more than one patient. Vehicles transporting low priority patients can serve another patient and this yields better usage of resources especially when a large portion of the vehicles are busy.

To our knowledge, this idea has not been considered in the past and this study models the service of two patients by an ambulance. In our discussions with EMS personnel, such a strategy is clearly being used during normal and surge situation.

The primary goal of this research is to develop models to dispatch EMS vehicles in a real time environment. The aim is to, first, assign EMS vehicles to patients, then assign them to hospitals (if required) and finally determine the base EMS station for each vehicle. We consider three different problems for this purpose.

In the first problem (Chapter III), these three different decisions are integrated in one mathematical programming model. This model uses real time information of EMS vehicles for their routing. The model is run whenever a state change occurs. A state changes whenever any of the following events occur:

- A vehicle becomes idle.
- A patient call is received at the call center.
- A vehicle must be dispatched to a hospital.

In the second problem (Chapter IV), we only consider idle vehicles and develop a model to efficiently allocate these vehicles in a real time environment. When coverage of the system drops below a given threshold, the optimization is run and it gives recommendation for relocating vehicles.

In the third problem (Chapter V), we only consider the dispatching problem to serve patients who have called and requested service from 911. Here, it is assumed the hospital of each patient is given prior to passing the call to the dispatcher. The hospitals can be selected by an external model such as the one in Sun et al. (2013) or patients choice. Here, the problem is to dispatch EMS vehicles to patients while ensuring specified time windows for serving patients are satisfied. Similar to Chapter III, serving more than one patient by a vehicle is allowed.

Two types of data are required for our model: static data and real time data. The static data must be known before running the model for the first time. This data are related to the facilities of EMS stations, geographical location of stations, and number and type of EMS vehicles. The real time data must be provided instantly at the time a decision must be made. This data includes the current location of EMS vehicles, vehicle state, and patient information. This data must be integrated with a routing tool and an optimization model to provide a decision support system for the dispatcher. Figure I.1 shows this procedure schematically.
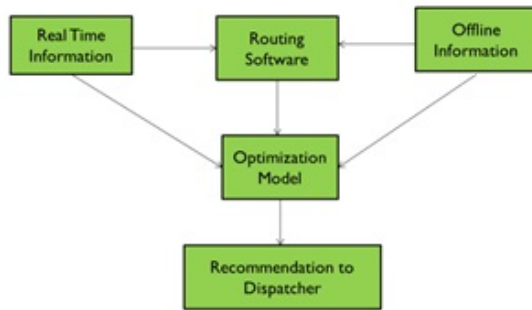
Figure I.1. Decision support system for EMS dispatcher

C Contribution

The contribution of this dissertation is as follows.

1. We develop a model for dispatching EMS vehicles that transport more than one patient.

2. We prove the EMS vehicle transporting two patients problem is NP-hard.

3. We develop a column-generation based procedure to compute the lower bound for the EMS dispatching problem with two patients.

4. We develop a simulated annealing meta-heuristic for solving the EMS dispatching problem efficiently.

5. We develop a simulation-optimization framework for relocating EMS vehicles in real time.

6. We develop a decision support systems employing advanced heuristics methods for possible adoption by EMS industry.

D Feasibility of Serving More than One Patient by an Ambulance

In this section, we discuss the practicality of serving more than one patient by an ambulance. Many counties such as Louisville KY have EMS vehicles that can

accommodate two patients, one on the bench and another on the stretcher. Thus, what we are proposing can actually be implemented in many areas of the country. It should be noted, however, that our model does not force all of the vehicles to serve more than one patient. If an ambulance is unable to transport more than one patient, it can be constrained to serve only one patient. The proposed model thus gives some flexibility to the dispatcher in reducing the service times by permitting ambulances with sufficient capacity and resources to transport more than one patient as required. If the condition of a patient deteriorates during transit, this can be considered to be a new event wherein the patient is re-classified as a high priority patient thus not allowing any more patients to be assigned to that ambulance.

Additionally, our model is customized for situations (e.g., major influenza outbreak) when the demand for vehicles is high. In such a situation, equipping vehicles with more than one bed, ventilator, isolation, and other devices can be more cost effective than other options. This leads to more efficient usage of vehicle capacity. In the special case of pandemic flu, it would be helpful to design and equip vehicles with isolation areas and facilities to prevent transfer of virus from one patient to another. In addition, transporting more than one patient in an ambulance has been suggested as one way to provide better response during medical emergencies. Relevant discussions include: the Emergency Medical Services and Non-Emergent (Medical) Transport Organizations Pandemic Influenza Planning Checklist (2006), Recommended Actions for EMS Providers to Prepare For and Respond to Pandemic(2009) and Russo (2006), for example.

E   Benefit of Applying Proposed Approach

To clarify how the proposed idea can improve a real system's performance, two examples are illustrated schematically. In these examples, we compare the solutions of the proposed method with a simple greedy heuristic where the closest vehicle is always dispatched. Figure I.2 shows that applying the proposed idea of

transporting two patients in one EMS vehicle can decrease the response time, increase the coverage area and reach a patient in the desired time window. Figure I.2 illustrates this example.
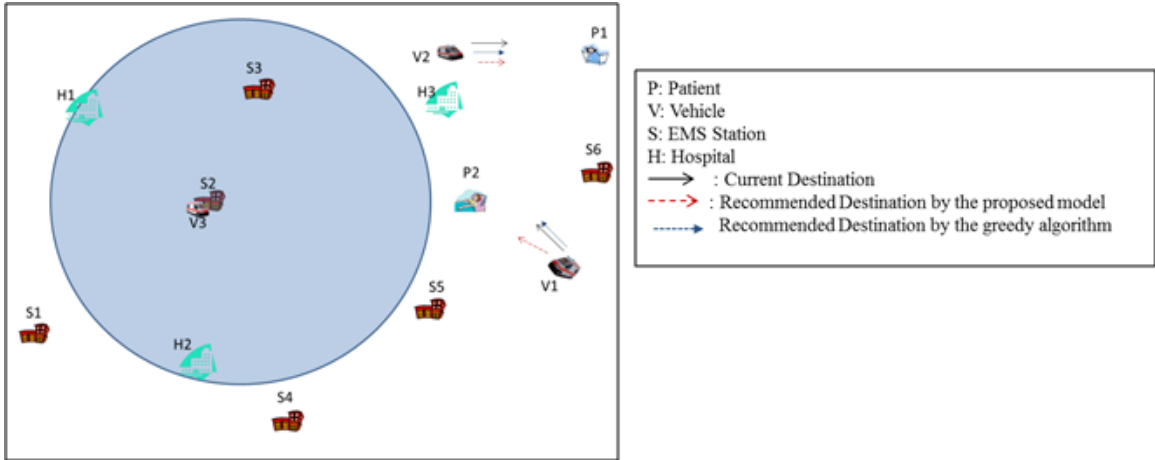
Figure I.2. Benefit of the serving more than one patient by one vehicle

In Figure I.2, if vehicle 3 is dispatched to serve patient 2 (suggested by the simple greedy algorithm) the shaded area in Figure I.2 will be left uncovered for some period of time and this decreases the coverage in the virtual city significantly. Additionally, vehicle 3 is not able to serve patient 2 in a required time window which is unsatisfactory for patient 1. While, if vehicle 1 is rerouted to patient 2, this patient can be reached in a required time window and also the shaded area in figure will be covered by vehicle 3.

In another example, shown in Figure I.3 the proposed idea of transporting two patients in one EMS vehicle proposes that vehicle 1 be rerouted to serve patient 2 and this causes both patients 2 to be served in her required time window, while in the solution of the greedy model which always assign the closest idle vehicle to the patient, patient 2 cannot be served in her required time window (8 minutes). This shows the advantage of using the proposed method for the patients. Additionally, travel time in the solution obtained by the proposed model is less than that in the solution obtained by the method which always assigns the closest idle vehicle. This

shows that patients, on average, can be serviced earlier when our proposed method is used by the dispatcher.
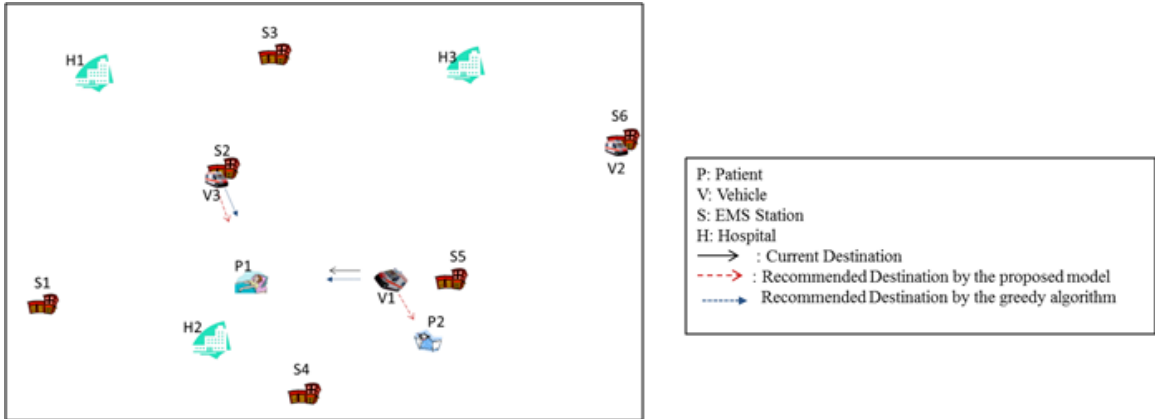
Figure I.3. Benefit of re-routing

F   Dissertation Organization

Below is the organization of the reminder of this dissertation:

Chapter II reviews previous studies that have applied operations research to EMS management. This chapter includes the literature related to the coverage and dispatching models for EMS vehicles. A brief review of the vehicle routing problem (VRP) is also presented.

In Chapter III, three optimization models are presented. These models consider an integrated approach for ambulance assignment when the hospital can be chosen by the dispatcher. They are customized for use during a pandemic outbreak. Additionally, experimental analysis of the proposed models is also presented. The analysis focuses on two aspects of the models: logic and performance.

Chapter IV considers the coverage problem independently. In this chapter, first an optimization model for the real time relocation of vehicles is proposed. After that, an event based simulation model is used to tune the parameters of the optimization model and evaluate its performance. The results of the simulation

show that applying the proposed optimization model will result in an increase in percentage of calls serviced in desired time window.

Chapter V considers the dispatching problem when patient hospitals are pre-determined. The problem is formulated as a vehicle routing problem. The problem is also proved to be NP-hard. Additionally an exact algorithm for obtaining lower bound using column generation method is developed.

In Chapter VI, two heuristic algorithms are presented for solving models presented in Chapters IV and V. Both algorithms are based on simulated annealing. Additionally, for dispatching problem (in Chapter V) a nested algorithm on top of simulated annealing is applied. The results of this algorithm is compared to the column generation method presented in Chapter V and it is shown that the heuristic provides high quality solutions.

Chapter VII presents a real time decision support system developed based on studies in the previous three chapters. The tools and the specialized algorithms within the decision support are described in this chapter.

Finally, Chapter VIII summarizes the dissertation and points future research.

CHAPTER II

LITERATURE REVIEW

This chapter reviews past studies related to the ambulance assignment problem. Numerous papers on this topic are available. An overview on these papers can be found in Owen and Daskin (1998), Brotcorne et al. (2003) and Goldberg (2004). The ambulance assignment problem can be categorized into two main sub-problems: coverage problem and dispatching problem. Some papers which have attempted to integrate the two sub-problems are also reviewed in this chapter, along with papers on the vehicle routing problem.

A   Coverage Problem

The aim of the coverage problem is to:

1. Determine the location of ambulance base stations.

2. Determine the number of vehicles and staff needed for each time period.

3. Redeploy vehicles to base-stations under different system states.

We first present the basic models. Then, probabilistic models are reviewed, followed by studies that incorporate multiple objectives, GIS approaches and heuristic approaches. Finally, papers which consider dynamic repositioning of ambulances are reviewed.

Set Covering Location Problem (SCLP)

Hakimi (1965) is among the first researchers to consider the problem of coverage in a network. Toregas et al. (1971) consider a set covering problem to

determine the location of emergency facilities. This problem is known as the Set Covering Location Problem (SCLP). The formulation of this problem is as follows:

$$\min \quad \sum_{j \in W} x_j \tag{II.1}$$

$$s.t. \tag{II.2}$$

$$\sum_{j \in W_i} x_j \geq 1 (i \in V) \tag{II.3}$$

$$x_j \in \{0, 1\} \tag{II.4}$$

In this model, the set of demands is denoted by V and the set of potential EMS facility locations is denoted by W, W_i is the coverage matrix, x_j is a decision variable and equal to 1 if facility is located at node j; x_j is equal to 0 otherwise. The objective function in equation (II-1) minimizes the total number of facilities used. The objective function has equal weights for the location of each facility. The constraints (II-2) ensure that each demand node is covered. The SCLP model does not consider some aspects of real life problems. First, if an ambulance is dispatched the area which it previously covered may not be covered for some time period. Also, the model assumes that a fixed number of vehicles are always available which may not be true in real situations. However, this model provides a good lower bound for the number of EMS vehicles needed for full coverage.

Maximal Covering Location Problem (MCLP)

Church and ReVelle (1974) consider a location problem and they extend the SCLP to maximize the demand coverage. The formulation is as follows

$$\max \quad \sum_{i \in V} d_i y_i \tag{II.5}$$

$$s.t. \tag{II.6}$$

$$\sum_{j \in W_i} x_j \geq y_i (i \in V) \tag{II.7}$$

$$\sum_{j \in W} x_j \leq N \tag{II.8}$$

$$x_j \in \{0, 1\} \forall j \in W \tag{II.9}$$

$$y_i \in \{0, 1\} \forall i \in V \tag{II.10}$$

In this model, y_i is a decision variable and is equal to 1 if node i is covered; is equal to 0 otherwise. Additionally, d_i denotes the demand of vertex i and N is the number of available vehicles. The remaining parameters are the same as in the SCLP model.

In this model, the objective function maximizes the covered expected demand. Constraints (II-4) impose y_i to be equal to 1, if an ambulance is located within the desired distance of node i. Constraints (II-5) restrict the total number of facilities to not exceed N.

SCLP and MCLP are complementary of each other. SCLP can be used for strategic and tactical planning to determine the number of vehicles and home stations needed while MCLP is an operational model to determine vehicle allocation to home stations.

Back Up Coverage Problem

Church and Waver (1985) develop the "vector assignment p-median model". In this work, it is assumed that x percent of demands points will be covered by the closest and y percent by the 2nd closest vehicle. Repede and Bernardo (1994) suggest repositioning vehicles based on the history of calls and demands during a

week. Gendreau et al. (1997) propose a model to maximize the double coverage for each node. In this work, a specific percentage of nodes need to be covered at least by one ambulance in a radius $r1$ and all of the nodes need to be covered in radius $r2$ ($r2 > r1$), while the objective function is to maximize the nodes covered by two or more vehicles in radius $r1$.

Probabilistic Models

Daskin (1983) is the first to consider the unavailability of a vehicle. He develops an expected covering location model, taking into consideration that a vehicle will be unable to respond to a demand. His model maximizes the number of demands that are covered. He also proposes a way to calculate the probability of availability of vehicle when it is idle. He assumes that vehicles operate independently in the model. This model is applied to a real situation in Austin, TX. In another study, Current et al. (1997) consider uncertainty in the number of vehicles. They consider the minimization of expected opportunity loss (EOL), and the minimization of maximum regret in their model.

Maximal Expected Coverage Location Model (MEXCLP)

Daskin (1983) also designs an integer programming model for the maximization of covered nodes. This model considers the probability of unavailability of vehicles and is written as follows:

$$\max \quad \sum_{i \in V} \sum_{k=1}^{N} (1-p)p^{k-1} d_i y_{ik} \tag{II.11}$$

$$s.t. \tag{II.12}$$

$$\sum_{j \in W_i} x_j \geq \sum_{k=1}^{N} y_{ik} \quad (i \in V) \tag{II.13}$$

$$\sum_{j \in W} x_j \leq N \tag{II.14}$$

$$x_j \in \{0,1\} \quad \forall j \in W \tag{II.15}$$

$$y_{ik} \in \{0,1\} \quad \forall i \in V \& k = 1.., N \tag{II.16}$$

In this model, p is the probability that an ambulance is busy, N is the maximum number of facilities, di is the demand in node i and x_j is equal to 1 if a facility is located at station j; 0 otherwise, y_{ik} is equal to 1 if node i is covered by at least k facilities; is equal to 0 otherwise. The remaining parameters are the same as in the SCLP model.

The objective function in Equation (II-6) maximizes the demand coverage. If a node is covered by $k-1$ vehicles, the expected covered demand is equal to $d_i(1-p^{k-1})$. If the kth vehicle is added to cover that node, the expected covered demand is incremented by this value $d_i(1-p^k) - d_i(1-p^{k-1}) = (1-p)p^{k-1}$. We can observe this value as the coefficient of y_{ik} in the objective function. Constraints (II-7) and (II-8) are the same as the constraints in the MCLP model. This model has been applied to some case studies. Fujiwara et al. (1987) conducts a case study for Bangkok, Thailand. In another study, Repede and Bernardo (1994) use Louisville, Kentucky data and make an extension of MEXCLP named TIMEXCLP. In this model, the variation of travel time is considered. Repede and Bernardo (1994) use simulation to assess their proposed approach and show that by applying this method, the response time can be decreased by 36% and coverage can be increased by 13%. ReVelle and Hogan (1989) construct a 0-1 integer

programming model similar to MCLP and calculate the probability of a vehicle being busy. Each node must be covered by at least b vehicles. This formulation is named Maximum Availability Location Problem (MALP I and MALP II). In MALP I, the availability percentage for all of vehicles is fixed, but in MALP II, the busy fraction varies for each vehicle. However, knowing the exact probability of being busy before allocating vehicles is unrealistic because this depends upon how vehicles are deployed to home stations.

Techniques such as hypercube model and simulation model are used to obtain the busy fraction.

Hypercube Models

Larson (1974) is the first to consider a hypercube algorithm in ambulance location. This method is used for evaluating the performance of the suggested allocation decisions. The model assumes calls are received based on Poisson process and the service time is exponentially distributed. These assumptions are used to calculate the probability of a vehicle being busy. Each patient has a priority list of locations and the first vehicle in this list which is idle will serve that demand. For simplifying the hypercube calculation, Larson (1975) proposes an approximate hypercube model.

Chelst and Jarvis (1979) develop a hypercube queuing model to estimate the probability of traveling times for emergency service systems. The hypercube models consider an EMS as a multi-server queuing system. Chelst and Jarvis (1979) calculate the travel time from each atom (the smallest unit of separation in a city) to another atom. They also consider the workload of each atom for each server and use the workload to predict the travel time.

Batta et al. (1989) implement a two-step procedure that uses Larsons hypercube algorithm. In their model, three assumptions of the TIMEXCLP are relaxed: servers operate independently, servers have the same busy probabilities,

and server busy probabilities are invariant with respect to their locations. They conclude that MEXCLP can overestimate or underestimate coverage up to 25%. This accuracy ly depends on the size of coverage matrix. If the matrix becomes larger, the problem becomes harder to solve. Problem complexity justifies the use of metaheuristics for solving coverage problems.

Queuing Models

In MALP (ReVelle and Hogan, 1989), it is assumed that server busy fractions are independent which this is not the case in Marianov and ReVelles (1996) work. In this work, Marianov and ReVelle (1996) develop a maximal availability location problem which uses queuing theory. In Marianov and ReVelle (1996), the model is connected to the MALP model for calculating the minimum number of servers required for an EMS station for a period of time.

Marianov and Serra (2001 and 2002) extend the set covering model to ensure that each vehicle is available with probability of p. In this study, queuing theory is applied and vehicles can be dependent on each other. They formulate a hierarchical queuing location set covering (HiQ-LSCP) to have complete coverage of the population while minimizing the number of servers. In HiQ-LSC, the objective is to minimize the cost of locating servers such that all demands are served from a specific distance of its origin. Each customer in this model can join the queue of servers when there are at most b other customers standing in the queue with probability of at least . In another model, they try to maximize the population covered, but serving entire population is not mandatory (HiQ- MCL). They also formulate the problem when a demand to a low level server can also be assigned to a er level server.

Multi Objective Models

Charnes and Storbeck (1980) develop a goal programming model for locating EMS vehicles to maximize coverage. Back up coordination is assured. They use multiple levels of emergency health service for basic life support and advanced life support facilities.

Badri et al. (1998) develop a multiple criteria model to locate fire stations. The criteria include minimizing fixed costs, maximizing service to those areas that have most demands, minimizing average and maximum distance travelled from station to accident sites, minimizing average and maximum time traveled, attaining targeted number of fire stations, minimizing service overlap, attaining favored area status for locating fire stations, and minimizing locations where water is not abundant.

GIS Approach

Some researchers have considered deployment of emergency vehicles from a GIS perspective. An overview on works involving facility location and GIS science can be found in Church (2002). Hillsman and Rhoda (1979) apply a GIS based method to locate EMS stations. Their focus is mainly zone structures with demand aggregated at zone levels. The most ideal case is to have as small unit as possible, but this increases the computational time significantly. Therefore, one node should represent an area for demand through a city. They specify three types of errors in structuring zones.

Type A errors: errors which measure the distance from the center of a cluster while the actual location is different from the location of aggregated calls.

Type B errors: errors caused by not knowing the location of calls Type C errors: errors in dispatching due to not knowing correct distance from bases to calls in aggregated zones.

Current and Schilling (1987) show how errors of types A and B can be

eliminated by using GIS approach. Hodgson and Neuman (1993) show how to eliminate errors of type C with GIS.

Heuristics Methods

Daskin (1983) develops a heuristic method for solving his own model (MEXCLP). This algorithm finds optimal or suboptimal solutions for all values of p (busy fraction) between 0 and 1.The algorithm guarantees optimality when p closes to 1, but when p closes to 0, optimality is not guaranteed. In this algorithm, when values of p close to 1, the optimal solution puts all facilities at the node with the est demand. Then, a search is conducted by performing single node substitutions to find the non-dominated solutions for different values of p.

ReVelle (1993) shows that the location problem can be solved by using general optimization software tools for a medium size of problem (100 demand points and 50 relocation sites). Gendreau et al. (1997) develop a tabu search heuristic for the coverage problem with the consideration of back up assurance. Aytug and Saydam (2002) develop four approaches for MEXCLP: Integer Programming, Daskins heuristic, a genetic algorithm and a genetic algorithm with local search. They conclude that GA which performs a random search can find a good solution in a reasonable time, Daskins algorithm can provide better solution but it is very time consuming and GA with local search can find high quality solutions but requires more computational time than GA.

Galvao et al (2005) develop a simulated annealing algorithm for unifying MCLP and MALP. They use the hypercube method for calculating the busy probability. Rajagopalan et al. (2007) develop an ANOVA to evaluate the performance of four met heuristics- evolutionary algorithm, tabu search, simulated annealing, and a hybridized hill-climbing algorithm - to see which of them provide better solutions. They conclude that EA (evolutionary algorithm) provide better solutions compared to others.

Dynamic Repositioning

Gendreau et al. (2001) develop a real time repositioning model using parallel computing. Their model is as follows:

$$\max \quad \sum_{i=1}^{n} \lambda_i x_i^2 - \sum_{j=1}^{m} \sum_{l=1}^{P} M_{jl}^t y_{jl} \tag{II.17}$$

$$s.t. \tag{II.18}$$

$$\sum_{j=1}^{m} \sum_{l=1}^{P} \delta_{ij} y_{jl} \geq 1 \quad \forall \nu_i \in V \tag{II.19}$$

$$\sum_{i=1}^{n} \lambda_i x_i^1 \geq \alpha \sum_{i=1}^{n} \lambda_i \tag{II.20}$$

$$\sum_{j=1}^{m} \sum_{l=1}^{P} \gamma_{ij} y_{jl} \geq x_i^1 + x_i^2 \quad \forall \nu_i \in V \tag{II.21}$$

$$x_i^2 \leq x_i^1 \quad \forall \nu_i \in V \tag{II.22}$$

$$\sum_{j=1}^{m} y_{jl} = 1 l = 1, .., P \tag{II.23}$$

$$\sum_{l=1}^{P} y_{jl} \leq p_j \quad \forall \nu_i \in V \tag{II.24}$$

Where y_{jl} is 1 if ambulance l is located at station j; 0 otherwise. x_i^1 is 1 if point i is covered; 0 otherwise. x_i^2 is 1 if point i is covered by more than ambulance. V is the set of demand points. nu_i is point i. $\delta_{ij} = 1$ if station j covers point i in radius r_1; 0 otherwise. Similarly, $\gamma_{ij} = 1$ if station i covers point j in radius r_2; 0 otherwise. M_{ij}^t is the travel time depending upon time interval t. P is total number of ambulances.

They use tabu search for determining the location of vehicles to solve this model. In this model, frequent movement of an ambulance is discouraged. Based on the history of movements between stations, each ambulance incurs a penalty. More frequent movement in the previous few hours causes the penalty coefficient to

increase. They find that applying this redeployment strategy to the city of Montreal can pre-compute 95% of strategies successfully. Comparing their solution with CPLEX, they find out that their heuristic has 2% deviation from the optimal solution in the worst case. This approach is also used in Sathe and Miller (2005) for a case study in Montreal, Canada.

Maxwell et al. (2010) use an approximate dynamic approach for redeployment of ambulances. They also design a simulation model to evaluate the performance of their model.

B   Dispatching Problem and Integrated Approach

Dispatching vehicles is another important problem for which operations research has been applied. The ambulance dispatching problem is a special type of general assignment problem which is addressed by Martello et al (1987), Ross et al (1975) and Fisher (1986).

Closest Available Vehicle

There has been a debate among researchers to see if dispatching the closest vehicle is the most favorable in every situation. Miller (1981) concludes that always dispatching the closest aircraft doesnt guarantee the best performance. Hogan and ReVelle (1986) suggest the closest vehicle dispatching for high priority patients. For low priority patients, they suggest that a vehicle should be dispatched if it can reach the patient in the required time window and if such a dispatch has minimal influence on the coverage. Cuninghame-Greene and Harries (1988) suggest the closest is good if the objective is to minimize the average response time.

Disaster Situation

Another aspect of the dispatching problem pertains to the management of fleet when there is a disaster. The aim of most papers on this topic is to serve as

many patients as possible. It is possible that vehicles may not be able to serve all the casualties.

Islam (1998) considers air tank redeployment for controlling fire in forests. His model determines the number of tanks required in each day and the locations to which they must be dispatched. Gong and Batta (2004) design a procedure when there are too many patients requesting EMS vehicles and the number of patients exceeds the number of vehicles. They categorize patients into three classes severe, moderate and mild. They propose two queuing control methods: One is the server cut off method, where they use a greedy search to determine how many vehicles to allocate to each group of patients. The other is queue length cut off. In this method, if the number of patients for priority 2 and 3 are fewer than specific thresholds, vehicles only serve priority 1 patients until the number of patients in queues 2 or 3 reaches their respective thresholds. They also define two dispatching strategies. One is the patient initiated dispatch. Under this strategy, the number of vehicles is more than the number of patients and an ambulance is dispatched which can minimize the response time for the current patient and a potential future patient. The other strategy is ambulance initiated dispatch, which is used when the number of patients is more than the number of vehicles. In this approach when an ambulance becomes idle, it is dispatched to serve the patient. This patient is selected based on the scenarios defined above for putting patients on queues.

Gong and Batta (2007) consider the growth of clusters during a disaster. They propose the relocation of ambulances in specific times. They allocate ambulances to clusters to minimize the time of serving all casualties. The earthquake in Northridge, LA in 1994 is considered as their case study. Gong and Lin (2009) develop a model for dispatching vehicles in disaster situations. They use clustering and develop a central dynamic routing based on parallel computing technology. They also use ArcGIS and the network of Changchun city for simulating an earthquake.

Jotshi et al. (2009) suggest a methodology for dispatching and routing vehicles when an earthquake has occurred and there are a large number of casualties. Simulation is used for verifying their approach.

Integrated Approach

As mentioned, if the objective is to minimize the total driving distance, dispatching the closest vehicle will provide the best solution, but when time window is considered for each patient dispatching the closest vehicle may not provide the most favorable solution. In this case, it is required to include multiple considerations to ensure that vehicle dispatch will meet these criteria.

Weintraub et al. (1999) present a model for dispatching vehicles for electric utility. These vehicles are similar to EMS dispatch in terms of random call arrivals and the traffic patterns. They geographically divide the calls into clusters and assign the vehicles to the clusters. Then, they design a heuristic tour for each vehicle to serve demands within a cluster.

Yang et al. (2004) develop an integrated approach for the dispatch of EMS, police and fire trucks. Their problem is an expansion of the multi-dimensional assignment problem. It is assumed that the number of vehicles is predetermined. This model could be used in a real time to decide the subsequent destination for each vehicle. In this model, rerouting of vehicles is allowed if it provides significantly reduces the performance of the entire system. The objective function attempts to minimize the travel time, maximize reaching a patient in the specific time window and coverage throughout the city. They also use a simulation model to evaluate the performance of the model.

Andersson and Varbrand (2007) propose new algorithms for ambulance dispatch and the dynamic relocation problems. Their work is similar to Weintraub el al. (1999) and is adopted for EMS vehicles. They categorize calls into three different categories: Priorities 1, 2 and 3. For priority 1 calls, the closest vehicle is

dispatched. For priority 2 and 3 calls, the vehicle that can reach the call in the time window is dispatched and its dispatch has the least effect on the coverage. In their algorithm, reassignment of vehicles is considered.

For dynamic ambulance relocation, Andersson and Varbrand (2007) construct a mathematical programming model whose objective is to minimize the maximum travel time of vehicles for relocating to different EMS stations. In this model, coverage of each region named preparedness cannot go under a specific value. They use a tree search algorithm for solving the dynamic relocation problem. They use a simulation model for evaluating their model.

Ibri et al. (2010) integrate dispatching and coverage problem in the same model and use combination of ant colony and tabu search algorithms for solving the model. They use the model proposed by Yang et al. (2004).

Recently, another study is dedicated to evacuation of hospitals when there is an emergency. Tayfur and Taaffe (2009) consider the problem of evacuation of hospitals. They build a mathematical model to minimize the cost of round trip of vehicles, cost of nurses and cost of not evacuating patients in a time period. The optimization model is modeled as mixed integer programming.

In this model it is decided how many patients to be evacuated in each stage, how many nurses are required and how many vehicles are required, and how vehicles should be assigned to different types of patients with their limited capacities. They discuss that their model cannot be solved using general solver optimization model solver in most cases, hence, they identify a lower bound model and use some heuristic approaches for solving the model. .

Vehicle Routing Problem

One of the problems related to ambulance dispatch is the vehicle routing problem (VRP). In ambulance dispatch, the decision maker needs to dynamically assign the fleet to demand points - the location of patients or hospitals. This makes

the dispatch problem similar to VRP which also assigns vehicles to demand points. So, studying approaches applied for solving VRP can provide good insights into solving ambulance dispatching problem. An overview on VRP models and solving techniques can be found in Toth and Vigo (2002).

The type of VRP that is more relevant to the ambulance dispatch problem is the one that vehicles can change their routes dynamically and each demand has a time window. In this type of VRP, the goal is to find vehicle routes that can service customers in the minimum amount of time. The following constraints are imposed on this problem:

Each vehicle is required to depart and terminate at a fixed depot.

Each demand needs to be served only once.

The capacity of vehicles is not exceeded and each customers demand must be satisfied.

Each demand has a service time window within which the service should be provided. The time window is a soft constraint and if violated, a penalty for wait or delay will be incurred.

Taillard et al. (1997) propose a parallel tabu search algorithm for VRP. In this method, first a search is conducted and the x numbers of the best routes found are stored in an adaptive memory. Then, the following procedure is applied for the limited number of iterations.

An initial solution is made by combining the solutions in the adaptive memory. Then, the current solution is disjoined into some subsets of routes. After that, a tabu search for each subset of routes is applied. Finally, the resulting routes are merged.

The best solutions from the above parallel tabu search are added to the adaptive memory and the procedure continues for a specific number of iterations. Finally, a post-optimization algorithm is used for each individual route.

Gendreau et al. (1999) use the Taillard et al.s (1997) method for routing and

dispatching vehicles in a real time situation. In fact, they add a dynamic aspect to the above mentioned parallel tabu search method. In this dynamic problem, if no events occur the optimization process is continued by using the previous search, but when an event occurs, the search is stopped and the best routes are added to the adaptive memory. If that event is the occurrence of a new demand, the adaptive memory is updated by inserting of the new demand in each solution and if no feasible found the demand is rejected. If the event is not the occurrence of new demand, the next destination of the driver is obtained from the best solution in the adaptive memory and the other solutions are updated. After that, a new tabu search is started using new solutions from the adaptive memory. Diversion is not allowed in this method and the drivers will choose their next destination based on the previous adaptive memory solution.

Gendreau et al.(2006) propose a neighborhood search structure for vehicle routing problem with the pickups and deliveries. In this method, they use a tabu search heuristic using parallel computing for real time dispatch of fleet in the case when a new request occurs. For the real time problem there are many considerations such as current location of vehicles, their current routes, policy of the service, travel time between points and etc.

Jung and Haghani (2001) formulate a mixed integer programing model for the time dependent vehicle routing problem (TDVRP). They consider soft time windows, real time service requests and real time variations in travel times between demand nodes. The objective function consists of the fixed cost of using a vehicle, routing cost and also some constraints related to the inconvenience of customers. They implement a genetic algorithm to solve that mixed integer programing model. Haghani and Jung (2005) compare their genetic algorithm with an exact method using lower bounds exact solutions. They show that GA can obtain very good results in this problem. Their GA can be used when the number of nodes is up to 70 and the demand locations in less than 20 minutes. They also compare static and

dynamic routing strategies and conclude that if there is uncertainty in forecasting travel time, dynamic strategy can provide superior solutions.

Malandraki and Daskin (1992) develop two heuristics for solving the TDVRP. In the first heuristic called Sequential Route Construction, a vehicle is for the node which has the shortest distance to the last visited node. This allocation continues as long as constraints on total time duration and capacity are satisfied. If these constraints are violated another vehicle is added and the procedure continues until all nodes are visited. In the second heuristic called Simultaneous Road Construction, a search is conducted for the nodes. That node which is the closest to all the available vehicles is selected. Then, this node is added to the tour of the corresponding vehicle. The procedure continues until all the nodes are visited or no more capacity is available. Additionally, cutting plane and dynamic programing methods are used for solving the traveling salesman problem which is a special case of the VRP.

In Malandraki and Daskin (1992), it is assumed that a vehicle leaving earlier than another, does not necessarily arrive sooner than the other one because the travel time is a step function and not a continuous function. In order to overcome this deficiency, they suggest virtual waiting at the customer place. This issue is solved in Ichoua et al. (2003). In this model, vehicle which leaves earlier, arrives earlier if the route and the time of arrival are the same.

Li et al. (2009) develop a model for the vehicle rerouting problem. This model is customized for scheduling a disruption due to breakdown of a vehicle. In such a situation, the other vehicles may have to change their tasks. These vehicles can be rerouted to serve the customers which had been originally planned to be served by the broken down vehicle. They show that vehicle rerouting problem is NP-hard problem. They provide a path based formulation of the vehicle rerouting problem with time windows and then, develop a Langrangian based heuristic. In this heuristic, a primal heuristic and finally a dynamic programing is proposed for

solving the Lagrangian relaxation problem quickly. They implement an insertion based heuristic for obtaining the primal solution.

Desrochers et al. (1992) consider a vehicle routing with hard time windows. They propose an exact method for solving VRPTW using column generation technique. In their approach, they convert VRP to a set partitioning model. They claim that set partitioning model can be converted to the set covering problem if the cost matrix dos not violate the triangular inequality. Hence, subset of routes is given to the set covering model and is solved using a general simplex algorithm. The simplex algorithm provides some dual variables associated with each constraint. These dual variables are used in pricing problem. Then, the pricing model is solved to determine which variable(s) should be added to the subset of routes. This pricing model can be modeled as a shortest path problem with resource constraints SPPRC. The regular shortest path problem can be solved using Dijkstra's algorithm. However this algorithm cannot be used when costs are negative. Bellman proposes a dynamic programing algorithm when the costs can be negative.

Desrochers et al. (1992) propose a dynamic algorithm to solve the problem similar to Bellman algorithm for the vehicle routing problem. They propose some algorithm to have one-cycle and two-cycle eliminations. This method can be solved with relatively low computational effort, however, it does not provide very good lower bounds.

Krumke et al.(2005) proved that the vehicle routing problem when each vehicle has to visit at most two requests is an NP-complete problem. They use a reduction function from the problem of 3 dimensional matching, 3DM [1], which is an NP-complete problem.

They consider a dispatching problem when the number of request is twice the number of vehicles. Then, they introduce an instance and show that this instance

---

[1]3DM problem. There are three finite, disjoint sets named $X, Y, and Z$ sets, and there is a subset named M of XYZ and has triples (x, y, z) such that $x \in X, y \in Y, and z \in Z$. Now $M \subseteq M$ is a 3-dimensional matching if the following condition is true: for any two distinct triples $(x1, y1, z1) \in M$ and $(x2, y2, z2) \in M, x1 \neq x2, y1 \neq y2, and z1 \neq z2$.

has an equivalent in 3DM problem. Hence, the dispatching problem when the number of requests is equal to two is an NP-complete problem.

Fillet et al. (2004) propose an algorithm for the elementary shortest path problem with resource constraint ESPPRC. They use some domination rules to avoid cycles.

Li and Lim (2001) propose a hybrid algorithm of tabu search and simulated annealing. They consider up to 50 requests and their computational time takes 30-4000 seconds on a i686. Bent and Van Hentenryck (2003) use a large neighbourhood search (LNS). They consider up to 500 requests and the largest problems was solved in 135 minutes on 1.2 Ghz AMD.

Ropke et al (2006) propose an extended large neighborhood search algorithm for the VRPPD. They consider up to 500 requests, and the average time to solve largest problems was 90 minutes, but some of the problems took up to eight hours to solve (on a Pentium IV, 1.5 Ghz).

Beaudry et al (2010) consider a problem for transporting patients in a hospital campus. Their problem is dynamic and new transportation requests arise dynamically and their objective is to minimize a weighted sum of three criteria: total travel time, total lateness and total earliness. They develop a two-phase heuristic procedure to solve their particular dial a ride problem. They use insertion movements and tabu search respectively.

Similarly, Kergosien et al (2011) consider the problem of transportation of patient between care units. They use a tabu search method similar to the method used in Gendreau et al. (1999).

One of the problems which is a subset of VRP is Dial a Ride Problem (DARP). The main difference between DARP and VRP is that DRP is only for transporting people. The approaches used in both are very similar. Cordeau and Laporte (2003) did an overview on models and algorithms for DARP. They discuss that three important decisions are made by solving DARP.

1) Determining clusters of users for a vehicle to serve.

2) Determining the sequence of users for each vehicle.

3) Scheduling pick-up, driving and drop-off activities along each route.

One of the applications of DARP is in transporting elderly and disabled persons. Madisen et al. (1995) consider this problem when user can either specify pick-up or drop-off time, but not both. In their problem, different types of vehicles are considered, availability of vehicles varies based on vehicles status and requests arrive dynamically. An insertion method named REBUS is used originally developed by Jaw et al. (1986). In this algorithm simply, all variations of insertion a new request to a route are examined and the insertion with the best objective function is selected.

Hunsacker and Savelsbergh (2002) proposed a procedure which evaluates feasibility of an inserting a request to a route which has been already constructed. They have shown that time complexity of insertion is $O(q)$ where q is the number of requests.

Paragh (2011) considers a DARP and develop two mathematical models for this problem. A variable neighborhood search is used to solve the problem.

C   Chapter Discussion

In this chapter, research on ambulance management is reviewed. The papers were categorized into three main areas: coverage problem, dispatching problem and vehicle routing problem.

Most of the research papers consider the static problems and they do not deal with the dynamic aspect of the ambulance dispatch and redeployment problem. In this era, with advanced telecommunication abilities, more accurate approaches are anticipated to facilitate real time decision making for deployment and dispatch of EMS vehicles in real time.

Not many papers dealing with dealing with EMS vehicle management

situations in pandemic outbreak situations are available. There is no unified mathematical approach which can help dispatchers allocate vehicles optimally by properly utilizing vehicle capacity. These two important gaps in the literature trigger the present research. This dissertation first presents an integrated approach for dispatching and relocating patients in a pandemic outbreak which uses real time information of EMS vehicles. After that, we consider the problems of dispatching and relocating separately.

In order to utilize the capacity of EMS vehicles, we have allowed one vehicle to serve maximum two patients. This problem has not been considered before in researches related to EMS dispatching domain. We have proposed new mathematical models and algorithms in Chapters III and V to efficiently dispatch EMS vehicles. Also, in Chapter IV, a new simulation-approach was proposed for the coverage problem which has not been considered in the literature.

CHAPTER III

AN INTEGRATED MODEL FOR DISPATCHING, HOSPITAL SELECTION AND RELOCATION OF EMS VEHICLES

A Problem Statement

We consider the EMS vehicle dispatching and relocating problem for determining, in real-time, the subsequent destination for each vehicle when there is a state change. A state change occurs when a patient makes a request for a vehicle by calling 911, or when a vehicle has just picked up the patient, or when a vehicle has just completed the service.

We make the following assumptions in our model. First, all vehicles are of the same type, and can serve any patient. If a patient needs more than one vehicle, his or her request will be split into two with the same location and priority. Second, all vehicles are equipped with global positioning systems and are tracked continuously.

Third, we assume that the information pertaining to hospital resource is known a priori. In this chapter we develop a mathematical model and corresponding algorithm to be embedded in a real-time decision support system module for the dispatch of EMS vehicles. It is part of a larger effort involving the development of a real time decision support system (RTDSS) (see www.rtdss.org) for use by healthcare and public health officials during and immediately after a medical emergency. Another module (see Sun et al 2013) developed for the RTDSS predicts patient flow to hospitals over a six to ten month period and estimates the resource shortage over that period. Some results from that module are used as input

information to our dispatch module. Upon receiving capacity information from another module, we pre-process the data for our dispatch model as follows. If a hospital does not have enough resources to admit patients, it will be excluded from the set of available hospitals.

Fourth, any time the model is invoked to make a dispatch decision, we only consider available vehicles and exclude those that are busy at hospitals, are at a patient scene and have been re-routed once. We also consider only the available hospitals and exclude those on diversion.

Fifth, the priority of each call is known by the dispatchers. In fact, one of the responsibilities of EMS call takers is to prioritize patient and pass this information to dispatchers. They use some protocols to prioritize patients (see, see e.g. Palumbo et al., (1996) and Clawson et al. (2007)) and our paper does not address this issue. Finally, we assume that vehicles have no fixed home stations.

With the above assumptions, our dispatch model accomplishes three tasks. First, it assigns a vehicle to a patient when a service request is received. This assignment is based on the current location of all vehicles and their availability. Note that a vehicle can be rerouted in this assignment, as long as such rerouting minimizes the total travel time (distance). The second task is to assign a hospital to a patient based on the travel distance to the hospital and the resource availability at that hospital. Finally, the model relocates vehicles that have completed the first two tasks to an EMS station so it can be ready to serve the next patient. The objective of the dispatch model is to minimize the travel time from the vehicles current locations to patient location, from patient location to hospitals, and from hospitals to patient location or EMS stations. It also minimizes the total penalties incurred if the pre-specified response time for patients is not met or if the relocation of EMS vehicles does not provide sufficient coverage for one or more regions.

Finally, unique to our model is its capability to transport a maximum of two patients to increase the service capacity during a medical emergency. Particularly,

we categorize patients into two groups: high priority and low priority patients. The weight of response time for high priority patients is greater than that for the low priority patients. A vehicle transporting a low priority patient is allowed to serve an additional patient. However, a vehicle transporting a high priority patient or two patients is not allowed to serve any other patient. EMS call takers prioritize a patient and this is an input to the model.

B   A Nonlinear Integer Program Formulation

In order to formulate the optimal dispatching model as a mathematical program, we introduce the following notation.

Sets:

$V = V^P \cup V_1^H \cup V_2^H \cup V^Z$: Set of available vehicles; mutually exclusive;

V^P : Set of vehicles en route to serve a patient

V_1^H : Set of vehicles transporting a high priority patient or two patients and enroute to a hospital

V_2^H: Set of vehicles transporting a low priority patient and enroute to a hospital

V^Z : Set of vehicles which are idle or enroute to an EMS base

$W = W^1 \cup W^2$: Set of patients

W^1: Set of high priority patients

W^2 :Set of low priority patients

Indices:

$i = 1, , , I$ Vehicles

$j = 1, , , J$ Patients

$k = 1, , , K$ Hospitals

$s = 1, , , S$ EMS stations

$r = 1, , , R$ Census tracts

$pr(j)$ Priority of patients, j= 1 or 2

Parameters:

x_{ij}^0 =1 if vehicle i has been dispatched to serve patient j; =0 otherwise

y_{ik}= 1 if vehicle i has been assigned to hospital k; =0 otherwise

T_{ij}: The predicted travel time for vehicle i to arrive at patient js location when departing at the current time

T_{ik}: The predicted travel time for vehicle i to arrive at hospital k when departing at the current time

T_{is}: The predicted travel time for vehicle i to arrive at EMS station s when departing at the current time

(Note that T_{ij}, T_{ik} and T_{is} are updated from one time frame to next)

T_{jk}: The predicted travel time for transporting patient j to hospital k

μ: The average service time for an ambulance at the patient location

$PPpr(j)$: The travel cost per time unit to reach patient j with priority of $pr(j)$

$VPpr(i)$: The travel cost per time unit to arrive at hospital for vehicle i which has patient with priority of $pr(i)$

IP: The travel cost per time unit for an idle vehicle

$A_{ij} = PP_{pr(j)} * T_{ij}$, The travel cost for EMS vehicle i to reach a patient

$B_{ik} = VP_{pr(i)} * T_{ik}$, The travel cost for EMS vehicle i to arrive at hospital k

$B_{ij} = VP_{pr(i)} * T_{ij}$, The travel cost for EMS vehicle i to reach another patient j

$B_{w(i)} = VP_{pr(i)} * \mu$,The waiting cost for the patient in an ambulance that is serving another patient at that patients location

$B_{jk} = VP_{pr(i)} * T_{jk}$, The cost for vehicle i to travel from patient js location to hospital k

$C_{is} = IP * T_{is}$, The cost for EMS vehicle i to travel to EMS station s

D: The penalty incurred by an ambulance for serving more than one patient

E: The penalty for inefficient rerouting of vehicle enroute to a hospital

$\boldsymbol{\alpha_p}$: The penalty of not serving a patient with priority $\boldsymbol{p}$ in a required time window

$\boldsymbol{\beta_r}$: The penalty of not covering census tract $\boldsymbol{r}$ by an ambulance

$\boldsymbol{\psi_{sr}}$ =1 if home station s can cover census track $\boldsymbol{r}$; 0 otherwise

$\boldsymbol{\phi_j}$: The required response time for patient $\boldsymbol{j}$ at time $\boldsymbol{t}$

$\boldsymbol{\omega}$: Threshold time for accepting a reroute

$\boldsymbol{\eta}$: Threshold value for allowing rerouting of ambulances to serve another patient

$\boldsymbol{g_i}$: Zip code of the patient served by that ambulance $\boldsymbol{i}$

Decision Variables:

$\boldsymbol{x_{ij}}$ =1 if vehicle $\boldsymbol{i}$ is dispatched to serve patient $\boldsymbol{j}$; 0 otherwise

$\boldsymbol{y_{ik}} = 1$ if vehicle $\boldsymbol{i}$ is dispatched to hospital $\boldsymbol{k}$; 0 otherwise

$\boldsymbol{z_{is}} = 1$ if vehicle $\boldsymbol{i}$ is dispatched to EMS station $\boldsymbol{s}$; 0 otherwise

$\boldsymbol{\pi_i} = 1$ if ambulance $\boldsymbol{i}$ serves more than one patient; 0 otherwise

$\boldsymbol{\theta_i} = 1$ if ambulance $\boldsymbol{i}$ is rerouted to serve more than one patient; 0 otherwise

$\boldsymbol{\tau_{ik}} = \boldsymbol{T_{ik}y_{ik}}$ if $\boldsymbol{\pi_i} = 0$; $\sum_j \boldsymbol{T_{ik}y_{ik}x_{ij}}$ if $\boldsymbol{\pi_i} = 1$

$\boldsymbol{\zeta_{ijk}} = \boldsymbol{x_{ij}y_{ik}}$

$\boldsymbol{u_j} = 1$ if patient j is not served in a required time window; 0 otherwise

$\boldsymbol{\nu_r} = 1$ if census tract $\boldsymbol{r}$ is not covered by at least one ambulance;0 otherwise.

$\boldsymbol{w_i} = 1$ if vehicle $\boldsymbol{i}$ is rerouted; 0 otherwise

Using the above notation, the real-time EMS vehicle dispatching (RTEMSVD) model can be formulated as the following integer nonlinear program.

RTEMSVD-A:

$$\min \sum_{i=1}^{I} \sum_{j=1}^{J} A_{ij}x_{ij} + \sum_{i \in V_1^H} \sum_{k=1}^{K} B_{ik}y_{ik} + \sum_{i \in V_2^H} \sum_{k=1}^{K} B_{ik}y_{ik}(1 - \sum_{i=1}^{I} x_{ij}) + \sum_{i \in V_2^H} \sum_{j=1}^{J} B_{ij}x_{ij} + \sum_{i \in V_2^H} \sum_{k=1}^{K} y_{ik}(\sum_{i=1}^{I} x_{ij})(\sum_{j=1}^{J} B_{jk}x_{ij} + B_{w(i)}) + \sum_{i=1}^{I} \sum_{s=1}^{S} C_{is}z_{is} + \sum_{r=1}^{R} \beta_r \nu_r + \sum_{p=1}^{2} \alpha_p \sum_{j \in W^P} u_j$$

(III.1)

s.t.

$$\sum_{i=1}^{I} x_{ij} = 1 \quad \forall j \tag{III.2}$$

$$\sum_{j=1}^{j} x_{ij} = 1 \quad \forall i \in V^P \tag{III.3}$$

$$\sum_{k=1}^{K} y_{ik} + \sum_{s=1}^{S} z_{is} = 0 \quad \forall i \in V^P \tag{III.4}$$

$$\sum_{i'=1}^{I} \sum_{j \in W^P} x^0_{i'j} A_{ij} - \sum_{i'=1}^{I} x_{i'j} A_{ij} - \omega \geq -M w_i \quad \forall i \in V^P \tag{III.5}$$

$$\sum_{j=1}^{J} x_{ij} x^0_{ij} \leq M(1 - w_i) \tag{III.6}$$

$$\sum_{k=1}^{K} y_{ik} = 1 \quad \forall i \in V_1^H \tag{III.7}$$

$$\sum_{j=1}^{J} x_{ij} + \sum_{s=1}^{S} z_{is} = 0 \quad \forall i \in V_1^H \tag{III.8}$$

$$\sum_{k=1}^{K} y_{ik} = 1 \quad \forall i \in V_2^H \tag{III.9}$$

$$\sum_{j=1}^{J} x_{ij} \leq 1 \quad \forall i \in V_2^H \tag{III.10}$$

$$\sum_{s=1}^{S} z_{is} = 0 \quad \forall i \in V_2^H \tag{III.11}$$

$$\sum_{s=1}^{S} z_{is} + \sum_{j=1}^{J} x_{ij} = 1 \quad \forall i \in V^E \tag{III.12}$$

$$\sum_{k=1}^{K} y_{ik} = 0 \quad \forall i \in V^E \tag{III.13}$$

$$\sum_{i=1}^{I} x_{ij} T_{ij} - \phi_j \leq M u_j \quad \forall j \tag{III.14}$$

$$1 - \sum_{s=1}^{S} \sum_{i=1}^{I} z_{is} \psi_{sr} \leq M \nu_r \quad \forall r \tag{III.15}$$

Equation (III.1) represents the objective consisting of eight terms. The first term

minimizes the vehicle travel cost to reach a patient. The second term minimizes the vehicle travel cost for vehicles transporting high priority patients to arrive at a hospital. Note that these vehicles will not serve any other patients.

The third through fifth terms, which cause the non-linearity of the formulation, minimize the travel costs for those vehicles transporting a low priority patient. For these vehicles, there are two possibilities. First, the vehicle can directly go to a hospital. In this case, because the vehicle will not serve any other patient,$(1 - \sum_{j=1}^{J} x_{ij} = 1)$and the travel cost is calculated as in the third term. Second, the vehicle can be rerouted to serve another patient. In this case,$(\sum_{j=1}^{J} x_{ij} = 1)$and three costs are incurred - the travel cost to the patient location as calculated in the fourth term; the waiting time cost for treating the second patient at the scene and the travel cost from the second patient to a hospital, as calculated in the fifth term.

The sixth term in the objective is the travel cost for vehicles to return to the EMS stations and the seventh term is the penalty for not serving patients in their required time windows. Finally, the last term calculates the penalty of not covering census tracts to fulfill future demands. The penalty of not covering a census tract by any ambulance, β_r, impacts where the vehicles are positioned. This penalty varies per census tract and it depends on the history of the number of calls and also the estimated number of patients in an outbreak in that census tract. The larger the number of calls, the larger this penalty.

Constraint (III.2) states that each patient is served by one ambulance. Constraints (III.3) through (III.6) pertain to those vehicles en route to serve patients. In particular, constraint (III.3) ensures any vehicle that has been dispatched is assigned to a patient. Constraint (III.4) states that vehicles en route to serve patients can neither go to a hospital nor to an EMS station. Constraints (III.5) and (III.6) allow a vehicle to change its route if there is a significant improvement in the overall travel time. Further, constraints (III.7) and (III.8)

pertain to vehicles transporting high priority patients. For these vehicles, (III.7) ensures they are assigned to a hospital, while (III.8) prohibits them from serving another patient or returning to the home station. On the other hand, constraints (III.9) through (III.11) pertain to vehicles transporting low priority patients. In particular, (III.9) assigns these vehicles to a hospital, (III.10) allows them to serve another patient, and (III.11) prohibits them from returning to a home station. Additionally, constraints (III.12) and (III.13) consider vehicles en route to or idle at EMS stations. Constraint (III.12) states that such a vehicle can either serve a patient or remain at an EMS station (the current or a different station), and constraint (III.13) prohibits such a vehicle from being assigned to a hospital. Finally, constraints (III.14) and (III.15) ensure that each patient is served within the required response time and that all census tracts are covered at a pre-specified level, respectively.

C   A Linear Integer Program Formulation

As previously mentioned, RTEMSVD-A is a non-linear integer program because the travel cost for vehicles transporting low priority patients depends on whether or not they will be assigned to a second patient. In the following alternative formulation RTEMSVD-B, we introduce binary variables ζ_{ijk} and π_i to linearize the third and fifth terms in the objective function of RTMESVD-A.

RTEMSVD-B

$$\min \sum_{i=1}^{I} \sum_{j=1}^{J} A_{ij} x_{ij} + \sum_{i \in V_1^H} \sum_{k=1}^{K} B_{ik} y_{ik} + \sum_{i \in V_2^H} \sum_{k=1}^{K} \tau_{ik} + \sum_{i=1}^{I} \sum_{s=1}^{S} C_{is} z_{is} + \sum_{r=1}^{R} \beta_r \nu_r + \sum_{p=1}^{2} \alpha_p \sum_{j \in W^P} u_j$$

(III.16)

s.t.

Constraints (III.1) to (III.14)

$$\sum_{j=1}^{J} x_{ij} \leq \pi_i \quad \forall i \in V_2^H$$

(III.17)

$$\sum_{j=1}^{J} x_{ij} - 0.5 \geq -M(1 - \pi_i) \quad \forall i \in V_2^H$$

(III.18)

$$-B_{ik}y_{ik} + \tau_{ik} \geq -M\pi_i \quad \forall i \in V_2^H and \forall k$$

(III.19)

$$\tau_{ik} - \sum_{j=1}^{J} (B_{jk} - B_{W(i)}\zeta_{ijk} + B_{ij}x_{ij} \geq -M(1 - \pi_i) \quad \forall i \in V_2^H and \forall k$$

(III.20)

$$x_{ij} + y_{ik} - \zeta_{ijk} \leq 1 \forall i \in V_2^H, \forall j and \forall k$$

(III.21)

$$-x_{ij} - y_{ik} + 2\zeta_{ijk} \leq 0 \forall i \in V_2^H, \forall j and \forall k$$

(III.22)

In RTEMSVD-B, the travel cost for the (low priority) patient currently in a vehicle to arrive at a hospital,τ_{ik} , is expressed in the third term, whether or not the vehicle is rerouted. Indeed, it is calculated in constraints (III.17) through (III.22). In particular, τ_{ik} equals to the time for the (low priority) patient in ambulance i to arrive at hospital k. If vehicle i is rerouted to serve another patient, as defined in constraint (III.20), τ_{ik} is the sum of the travel time cost from the current vehicle location to the second patient's location, the waiting time cost at the second patient's location, and the travel time cost from the second patient's location to the hospital. If the vehicle does not serve any other patient, τ_{ik} is the travel time from its current location to the hospital, as in constraint (III.19). Other constraints (III.17), (III.18), (III.21) and (III.22) are auxiliary constraints.

Although RTEMSVD-A and RTEMSVD-B are mathematically equivalent,

our numerical study suggests that the linear integer program RTEMSVD-B is computationally more efficient. In the next section, we present a heuristic method for solving RTEMSVD-B that further reduces the solution time for large-size dispatching instances.

D   An Efficient Approximation Method

If a vehicle transporting a low priority patient is rerouted to serve a second patient, the total travel costs in RTEMSVD-B include the travel cost to arrive at the second patient, the time spent at the second patient's scene, and the travel cost from the second patient to the hospital. The exact calculation of these costs requires additional variables (e.g., π_i and ζ_{ijk}) and constraints in the integer linear program. We propose to use an estimated parameter to represent the fixed travel cost incurred by vehicles that are rerouted. Doing so significantly reduces the number of binary variables and constraints, thus increasing the computational efficiency. The approximate model is formulated as follows.

RTEMSVD-C:

$$\min \sum_{i=1}^{I} \sum_{j=1}^{J} A_{ij} x_{ij} + \sum_{i \in V_1^H \ V_2^H} \sum_{k=1}^{K} B_{ik} y_{ik} + \sum_{i=1}^{I} \sum_{s=1}^{S} C_{is} z_{is} + \sum_{r=1}^{R} \beta_r \nu_r + \sum_{p=1}^{2} \alpha_p \sum_{j \in W^P} u_j + D \sum_{i \in V_2^H} \pi_i + E \sum_{i \in V_2^H} \theta_i$$

(III.23)

s.t.

$$Constraints (III.2) to (III.15)$$
$$Constraint (III.17)$$

$$\sum_{j \in W^P} t_{jk} x_{ij} - \sum_{k'=1}^{K} t_{ik'} y_{ik'}^0 - \eta \leq M(1-\pi_i) + M\theta_i + M(1-y_{ik}) \quad \forall i \in V_2^H \, and \, \forall k$$

(III.24)

39

As in RTEMSVD-A and RTEMSVD-B, the objective function in RTEMSVD-C minimizes the total travel costs, penalties of not transporting patients within the required time window and not covering census tracts. However, objective (III.23) simplifies the calculation of costs incurred due to re-routing all vehicles by introducing a fixed rerouting cost D in the sixth term.

Note that with this simplification, the travel cost between the vehicle locations and the hospitals in the second term are generalized to all vehicles, i.e., $V \in V_1^H \, or \, V_2^H$, whether or not they are rerouted. Furthermore, the seventh term in (III.23) penalizes any inefficient re-routing with a cost of E. An inefficient rerouting occurs when the travel time from the second patient's location to the suggested hospital is larger than the travel time from the vehicles current location to its previously assigned hospital plus the threshold of η. The value of η is set to be 25% of the time for a high priority patient who has just called, i.e $\eta = \phi_j/4$. Figure III.1 illustrates "efficient" and "inefficient" re-routings. The solid arrows in these figures represent the current destinations of vehicles and the dashed arrows represent the destinations after re-routing. Finally, constraint (III.24) determines whether or not the rerouting is efficient using binary variable θ_i.

E   Numerical Results

We evaluated the proposed models using numerical examples. LINGO is used to solve the associated linear/nonlinear programs. The CPU times reported here are from a Dell Intel(R) W3503 Processor with 2.4 GHz and a 4GBRAM: on a 64-bit operating system.

As illustrated in Figure III.2, Example 1 has five vehicles, four patients, two hospitals, six census tracts and three EMS stations. Table III.1 shows the current locations and status of the five vehicles. Similarly, Table III.2 provides the information on the location, time of call, priority class and required service time window of the four patients, while Table III.3 displays the locations of hospitals and
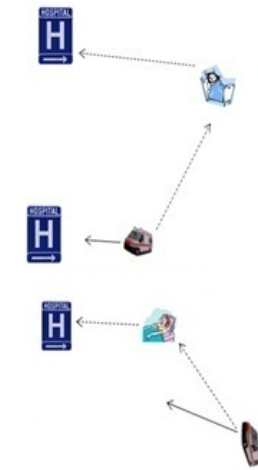
Figure III.1. Examples of inefficient rerouting(upper) and "efficient" rerouting(lower)

P: Patient
V: Vehicle
S: EMS Station
H: Hospital
→ : Current Destination
-→ : Recommended Destination
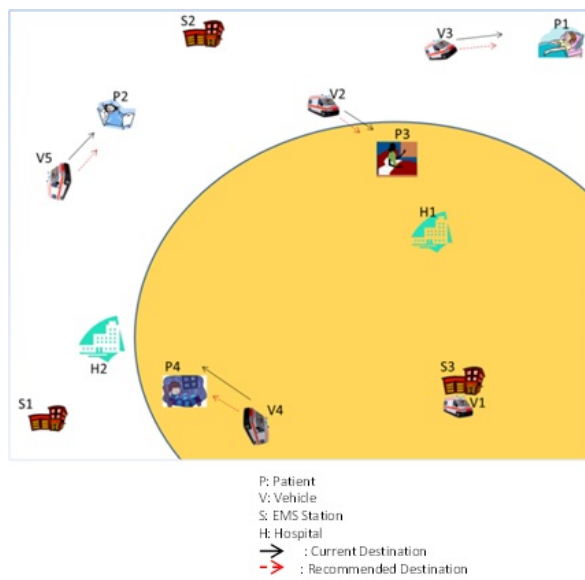
Figure III.2. Illustration of Example 1

TABLE III.1

Vehicle Information

Vehicles	X	Y	Status
1	14	3	idle at EMS station#3
2	12	14	en route to serve patient#3
3	16	16	en route to serve patient#1
4	9	3	transporting a low priority patient to hospital#2
5	2	10	on the way to serve patient#2

TABLE III.2

Patient Information

Patients	Priority	X	Y	Remaining Time Window(minutes)	Time of Call
1	Low	17	17	7	8:22
2	High	4	12	3	8:25
3	Low	14	12	12	8:27
4	High	8	4	8	8:30

TABLE III.3

Hospitals and EMS Stations Information in Example 1

Hospital	X	Y
Hospital 1	14	14
Hospital 2	7	5
EMS Station 1	1	1
EMS Station 2	7	16
EMS Station 3	14	3

TABLE III.4

The value of coefficients in the objective function in Example 1.

α_1	α_2	PP(1)	PP(2)	VP(1)	VP(2)	IP	D	E
10000	2000	100	20	30	20	5	200	400

EMS stations. In this and subsequent examples, we assume the distance unit to be a kilometer. For example, if the position of hospital 1 is (7, 5), it is located 7 km to the east and 5 km to the north of the origin. The speed of ambulance is assumed to be 60 km/hr. The desired response time is 8 minutes for high priority and 15 minutes for low priority patients and the average waiting time at the patient scene is 10 min. Table III.4 lists the value of coefficients in the objective functions. The value of β_r ranges from 8 to 52.

We assign the values of penalties for the multiple factors based on priorities associated with these factors. The most important consideration in dispatching EMS vehicles is to reach patients in the required time window. The required time window for the high priority patients is less than that for the low priority patients. This penalty should be larger than $J * PP_1 * Max(T_{ij})$ to ensure that the penalty of not reaching a patient in the specified time window is larger than the maximum penalty of travel time to reach all high priority patients in their required time windows. For the same reason α_2 should be larger than $J * PP_1 * Max(T_{ij})$ for low priority patients.

The next important coefficient is the travel time penalty for reaching a high priority patient (PP_1). In other words, this penalty is the third priority after time window penalties (1 and 2). On the other hand, the travel time penalty for reaching a low priority patient (PP_2), does not need to be very large, as long as a low priority patient can be reached within the required time window. These penalties (PP_1 and PP_2) factors can be adjusted by estimating the cost of waiting per minute by EMS administers. The travel time penalties for transporting a patient to a hospital are VP_1 and VP_2. VP_1 should be larger than VP_2 for obvious reasons. VP_2 should be close to the value of PP_2 to almost have the same cost per time unit for a low priority patient to be waiting in ambulance or in the patient scene.

The penalty of not covering a census tract by any ambulance, β_r, affects where the vehicles are positioned. This penalty varies per census tract and it

depends on the history of the number of calls for that census tract. The larger the number of calls, the larger this penalty. To prevent frequent idle movements of vehicles and long travel times, the penalty parameter $\boldsymbol{IP}$ is set such that the reconfiguration of vehicles occurs when there is a substantial improvement to the coverage.

In the fixed cost model, coefficients $\boldsymbol{D}$ and $\boldsymbol{E}$ play important roles in the optimal solution. Intuitively, we should set $\boldsymbol{D} \geq \boldsymbol{VP_2\mu}$ to guarantee that the fixed cost for rerouting a vehicle from a hospital is at least larger than the penalty of waiting time at the patient scene. On the other hand, the coefficient $\boldsymbol{E}$ should discourage large deviation of vehicles transporting low priority patients from their ways to the hospital. Therefore, we should have $\boldsymbol{E} > \boldsymbol{VP_2\phi_j}$, where $\boldsymbol{\phi_j}$ is the time window for a low priority patient who just called. In this case, if there is an idle ambulance which can reach patient $\boldsymbol{j}$ in the desired time window, an inefficient rerouting will be discouraged.

Assuming the maximum travel is 17(2)0.5 which is the diameter of the virtual square city in Figure III.2, $\boldsymbol{\alpha_1}$ needs to be 4* 17(2)0.5 * 100 which is almost 10,000 and $\boldsymbol{\alpha_4 * 17(2)0.5 * 20}$ which is almost 2,000. The factor $\boldsymbol{D}$ is to be larger than 20*10=200 and $\boldsymbol{E}$ is to be larger than 20*15=300.

We have also tested 100 random instances similar to Example 1 with four patients, five vehicles, three EMS stations and two hospitals. In these instances, the values of $\boldsymbol{\alpha_1}$ ranged from 4,000 to 15,000 and no change in the optimal solution is found compared to the values listed in Table III.4. Similarly for $\boldsymbol{\alpha_2}$ 100 instances are run with the values of $\boldsymbol{\alpha_2}$ changes from 1,000 to 3,000 and in 90 of the instances, the optimal solution is the same as the optimal solution yielded by the value of 2,000. Also, in these instances it is observed that within the range of (1800, 2400) all of the solutions are the same.

This example demonstrates the value of allowing an EMS vehicle to transport two patients. From Figure III.2, patient 4 is on the way of vehicle 4 to hospital 2.

Thus, assigning vehicle 4 to this patient is the most efficient. But when vehicles are not allowed to transport two patients, vehicle 1 would be assigned to patient 4. The latter causes some area (the area shaded in in Figure III.2) in the eastern part of the city to lose its coverage. Finally, we note that for this example, the same optimal solution is returned by the heuristic method based on the approximation model RTEMSVD-C.

In order to examine the benefit of allowing two patients in an ambulance more extensively, more cases should be considered. Table III.5 shows the average waiting time of patients for the cases when only one patient in an ambulance is permitted versus the case when two patients in an ambulance are allowed. In this table, 12 different sizes of vehicles and patients are listed in the rows and each row is a product of five random instances. In these instances, it is assumed that 60% of vehicles are idle and no vehicle has been assigned to patients yet. As shown in Table III.5, in all of the examples, the average waiting for the case when two patients are allowed is less than the case when only one patient is allowed in an ambulance. As seen in the last column, the average waiting time for models are close to each other when the number of patients is low, but, when the number approaches the number of idle vehicles, average waiting time when more than one patient is considered is significantly better.

We also examine the computational time of the three models. As shown in Tables III.6 and III.7, RTEMSVD problems with various numbers of vehicles and patients are tested. Table III.6 reports the average computational times (in CPU seconds) for five randomly generated instances for each combination of vehicle and patient sizes using three models RTEMSVD-A, RTEMSVD-B, RTEMSVD-C (listed as A, B and C, in the table, respectively). In these examples, if the number of patients exceeds the number of vehicles, we assume patients are put in the queue and are not considered in the model. Each time a patient's request is being considered, there is a state change and the dispatch model is invoked. The columns

TABLE III.5

Average Waiting Time for Patients

No. of Vehicles	No. of Patients	Not Allowing two patients	Allowing two patients	Percentage Improvement
100	10	5.254699234	5.125296403	2.5%
100	20	5.693098413	5.46074175	4.5%
100	30	6.532742911	5.861419792	11.3%
100	40	8.551500106	7.272180501	17.6%
100	50	8.16786918	6.931952421	17.8%
100	60	9.567018703	6.313189941	51.5%
150	15	4.950919443	4.636213204	6.8%
150	30	5.293697101	4.860380155	8.9%
150	45	6.50916452	5.673660528	14.7%
150	60	6.189192738	5.360870132	15.5%
150	75	6.588699506	5.400426864	22%
150	90	8.409576115	5.613277238	49.8%

in Tables III.6 and III.6 represent the number of patients in the system who have called thus far and it is assumed two news calls are just received. For example, in column 1 of Table III.6, there are 10 patients waiting for EMS vehicles and two patients call and request two vehicles. Thus, the model assigns two vehicles to the newly called patients and also decides whether any of the previously assigned vehicles require rerouting.

Some observations can be made from Table III.6. First, the solution time for all three models increases as the number of vehicles increase. Second, the heuristic method based on the approximation model RTEMSVD-C requires the least CPU time, followed by the linear integer model. Because the proposed EMS dispatch model is intended for a real-time decision support system, we consider the acceptable solution time to be no more than one minute. Our computational results in Table III.6 suggest models B and C are suitable for practical implementation when the number of patients and vehicles are in these ranges. Hence, we drop the nonlinear integer model A from consideration, and only compare the performance of

TABLE III.6

Computational Time (in CPU seconds)

No.of Vehicles	No. of Patients														
	10			20			30			40			50		
	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
15	6	3	1												
25	45	5	1	21	1	3									
35	112	13	2	104	6	2	30	2	2						
45	64	7	2	155	12	4	1161	13	3	32	2	3			
55	113	9	2	175	15	5	1464	8	3	86	6	5	46	3	3

TABLE III.7

Computational Time (in CPU seconds)

No.of Vehicles	No. of Patients											
	60		70		80		90		100		120	
	B	C	B	C	B	C	B	C	B	C	B	C
65	5	4										
75	7	5	6	7								
85	48	8	6	6	6	5						
95	114	8	64	10	7	6	8	6				
110	191	11	85	9	67	8	6	4	5	4		
130	180	13	127	14	224	13	107	10	80	8	8	6

the heuristic model C and the integer linear model B in the further evaluations.

To evaluate the performance of the model when the number of patients is large, more experiments are considered. In this case, we have assumed that the number of patients (at the one time instance) in the system in a medium size metropolitan (such as Louisville, KY) would not exceed 120, which appears to be an acceptable estimate given that the Level 1 Trauma Center Emergency Department at the University of Louisville had 38,000 ER patients in 2011, about 104 per day. Table III.7 reports the computational time of Models B and C for sizes of vehicles between 65 and 130, and the range of patients is between 65 and 120. Similar to

Table III.6, each CPU time is a product of five instances. As shown in Table III.7, Model C in almost all the cases is faster than Model B. Also, when the number of patients approached the number of vehicles, the computational time decreases. This could be explained by the fact that the set of feasible solutions is small because most of the vehicles need be assigned. Hence, there is not too much room for optimization.

While Tables III.6 and III.7 reveal the impact of problem size on the solution time for all three models, Figures III.3 and III.4 show how the computational time is affected by the number of vehicles requested at the moment, and by the number of vehicles transporting low priority patients, i.e., the number of vehicles that may be rerouted, respectively. Particularly, the numerical examples for Figure III.3, consider a scenario with 8 hospitals, 25 EMS stations, and 62 vehicles. Among the 62 vehicles, 30 are transporting low priority patients (i.e., $|V_2^H| = 30$). Further, the number of vehicles requested range from one to ten, as seen in the horizontal axis in the figure. Figure III.4 shows that the computational time for the heuristic model C is less sensitive to the number of vehicles requested than model B. For example, when the number of requested vehicles increases from 2 to 10, the computational time for model C stays around 3 seconds, while that for model B increases from 3 to 44 seconds. On the other hand, Figure III.4 uses the same scenario as in Figure III.3 by fixing the number of vehicles currently requested as 15. Particularly, it compares the computational times for models B and C when the number of vehicles subject to rerouting (i.e., the number of vehicles transporting low priority patients) changes. Again, the heuristic model C is more robust in terms of CPU time than model B. For instance, as the number of vehicles transporting low priority patients changes from 5 to 25 to 30, the computational time for model C stays around 3 seconds, while the one for model C increases from 12 to 300 seconds. In summary, both Figures III.3 and III.4 suggest to adopt the heuristic model C when the number of requests is large as in a medical emergency.
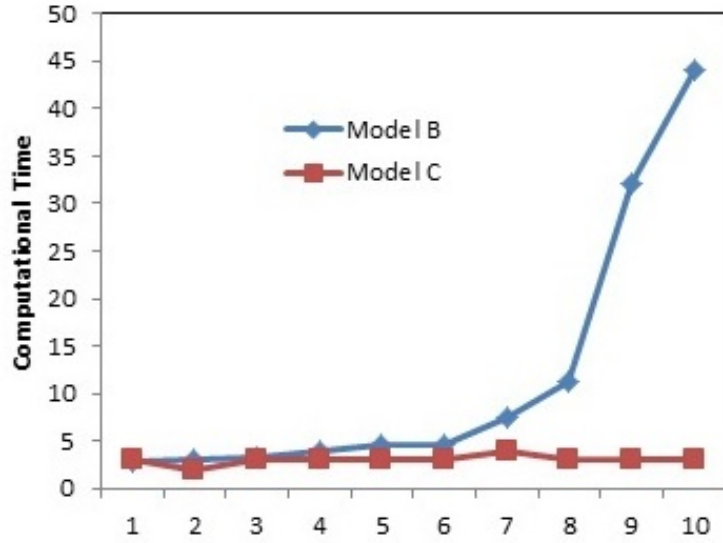
Figure III.3. Computational Time vs. the Number of Vehicles Requested at the Moment
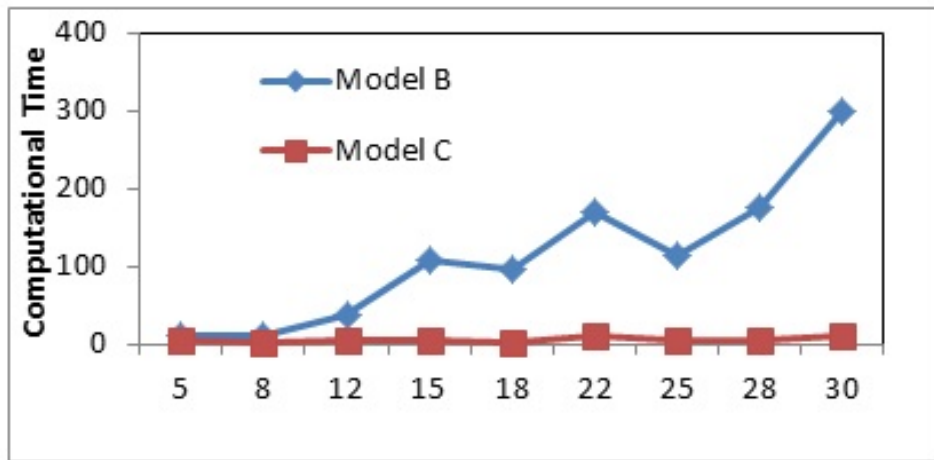
Figure III.4. Computational Time vs. The Number of Vehicles Belonging to V_2^H

The above analyses suggest that the heuristic method based on model RTEMSVD-C is more advantageous in solution time and is more robust compared to RTEMSVD-A and RTEMSVD-B. We now study the solution quality from the approximation model RTEMSVD-C. Particularly, we compare solutions from RTEMSVD-C with those from RTEMSVD-B using two sets of tests. Test instances in the first set allow only one patient to call and request EMS vehicles at a time, while in the second set of test instances multiple patients may call at one time.

We randomly created 30 instances similar to Example 1 in the first set. The number of patients in these instances ranges from 10 to 27. In the first 10 instances, the number of vehicles is 100, among which 20 are transporting low priority patients (thus are subject to rerouting), and the remaining 90 vehicles are either idle or en route to serve a patient. In the second 10 instances, the number of vehicles is 60, among which 30 are transporting low priority patients (thus are subject to rerouting), and the remaining 32 vehicles are either idle or en route to serve a patient. In the last 10 instances, the total number of vehicles is 35, among which 30 are transporting low priority patients and are subject to rerouting. All 30 instances in the first test have fixed numbers of EMS stations (26), hospitals (8) and census tracts (196). These values are chosen to represent the real-world situation in cities like Louisville, KY. The results for the 30 instances in this set are quite favorable to the heuristic model RTEMSVD-C. Particularly, RTEMSVD-C yields the same optimal dispatching solutions as does RTEMSVD-B for all of the first and second 10 instances, where the percentage of vehicles that are subject to rerouting is 10% (10/100) and 48.3% (30/62). In the third set of 10 instances, where the percentage of vehicles that are subject to rerouting is 85.7% (30/35), RTEMSVD-C provides the same optimal dispatching solutions as RTEMSVD-B for nine instances in this group. In the only instance where the heuristic model (RTEMSVD-C) provides a different solution from the exact model (RTEMSVD-B), the discrepancy occurs only to one patient's vehicle assignment.

In the second set of tests, we created 30 instances in which the numbers of vehicles, hospitals, EMS stations and census tracts are the same as those in the first set. However, the number of patients who call to request EMS vehicles is set to be 10. These attributes of the 20 instances are listed in columns of No. of Vehicles, No. of Vehicles in Set V_2^H , and No. of Patients Calls, respectively, in Table III.8. The computational comparisons between RTEMSVD-B and RTEMSVD-C are summarized in Column No. of Vehicles Dispatching the Same Patients. The criterion in this comparison is to examine among the total of 10 vehicles requested (the same as the total number of patients calls), how many of them are dispatched to the same patients by the two models. For example, for instance 18, there are 62 vehicles (30 of them are subject to rerouting) and 10 patient calls from different locations at the moment. Models RTEMSVD-B and RTEMSVD-C assign the same vehicles to 9 of the 10 patients. Overall, 91.6% of the patients are assigned to the same vehicles by the two models. When examining the first group of 10 instances where the percentage of vehicles that are subject to rerouting is 10% (10/100), the two models assign the same vehicles to all 10 patients for all instances. For the instances 11 to 20, where the percentage is 48.3% (30/62), the two models assign the same vehicles to all 10 patients for three instances, to 9 patients for six instances, to 8 patients in one instance. This averages about 92% of patients who are assigned to the same vehicles. For, instances 21 through 30 where the percentage of vehicles that are subject to rerouting is 85.7% (30/35), the overall percentage of patients who are assigned to the same vehicles drops to 83%. Particularly, there is one instance that both models agree on the vehicle assignments for 9 of the 10 patients, two instances for 8 of the 10 patients, and four instances for 7 of the 10 patients.

Collectively, the results on the 60 instances from the two sets of tests suggest that the heuristic model RTEMSVD-C provides quality solutions when the percentage of vehicles that are subject to rerouting is low, medium or high, regardless the number of patients currently requesting EMS vehicles. Furthermore,

TABLE III.8

Computational Time (in CPU seconds)

Instances	No. of Vehicles	No. of Vehicles in Set V_2^H	No. of Patients Calls	No. of Patients Assigned to the Same Vehicle	Percentage of Patients Assigned to the Same Vehicle
1	100	10	10	10	100%
2	100	10	10	10	100%
3	100	10	10	10	100%
4	100	10	10	10	100%
5	100	10	10	10	100%
6	100	10	10	10	100%
7	100	10	10	10	100%
8	100	10	10	10	100%
9	100	10	10	10	100%
10	100	10	10	10	100%
11	62	30	10	9	90%
12	62	30	10	10	100%
13	62	30	10	9	90%
14	62	30	10	9	90%
15	62	30	10	9	90%
16	62	30	10	8	80%
17	62	30	10	10	100%
18	62	30	10	9	90%
19	62	30	10	9	90%
20	62	30	10	10	100%
21	35	30	10	10	100%
22	35	30	10	8	80%
23	35	30	10	7	70%
24	35	30	10	7	70%
25	35	30	10	9	90%
26	35	30	10	7	70%
27	35	30	10	7	70%
28	35	30	10	10	100%
29	35	30	10	10	100%
30	35	30	10	8	80%

the heuristic model also provides high quality solutions (90% solutions are optimal) when the percentage of vehicles that are subject to rerouting is (about 85%) and there is only one patient requesting EMS vehicle at a time. Finally, the solution quality from the heuristic model RTEMSVD-C is still good (83% solutions are optimal) when a high percentage of vehicles are subject to rerouting, and there are multiple patients requesting vehicles at a time. We envision that in the real world, it is rather improbable that such a situation would occur, especially when the decision support system is implemented in real time. Therefore, it is concluded that solutions from the heuristic model is robust and high quality.

F   Conclusions

In this chapter, an integrated approach for dispatching EMS vehicles during a pandemic outbreak is presented. In such situations the demand for EMS vehicles increases. To increase the operational capacity, the idea of serving more than one patient by an ambulance is introduced. Mixed integer programming is used for modeling the system. Three optimization models are proposed for dispatching EMS vehicles. The objective is to minimize the total travel distance, the penalty of not reaching a patient in a required time window and the penalty of not covering a census tract. The first model is an integer nonlinear programming model (INLP). The other two models are linear. The first linear model called ILP model is a linear version of INLP model. The second linear model called the fixed cost model is not as accurate as the two previous models in terms of calculating the predicted travelling distance of each ambulance transporting a low priority patient.

In addition, we propose a heuristic model to approximate the mixed integer linear program. Numerical results on randomly generated instances show that the linear integer model requires less CPU time than does the nonlinear integer model. Furthermore, the heuristic approximation model uses substantially less CPU time than the linear integer model and provides robust and high quality solutions for

most cases we have tested.

CHAPTER IV

AN OPTIMIZATION AND SIMULATION MODEL FOR THE EMS VEHICLE COVERAGE PROBLEM

A   Problem Definition and Mathematical Model for the Real Time Coverage Problem

In this chapter, we consider the coverage problem which deals with reassigning units to stations in real time. First, a mathematical model is presented. Then, a simulation model is presented to tune the parameters and analyze the benefit of applying the model.

The problem is to relocate EMS vehicle in a real time environment. A vehicle is considered relocated when it has changed its station. The ultimate goal of solving this problem is to minimize number of calls which cannot be serviced in a desired time window. Here, we consider a call as a missed call when it cannot be serviced in a specified desired time window. Ideally, if vehicles can relocate instantaneously, the percentage of missed calls will be minimized. However, in reality this is not possible and relocation has a cost. Therefore, as a means to the goal, which is minimization of the percentage of missed calls, we wish to maximize the coverage while minimize the relocation costs. In order to find the tradeoff between these two factors, a simulation is modeled and applied.

The mathematical model for the coverage problem is as follows.

Indices:

$i = 1, , , I$ Vehicles

$s = 1, , , S$ EMS stations

$r = 1, , , R$ Census tracts

Parameters:

C: The penalty assigned to an idle vehicle traveling to an EMS station

β_{rq}: The encouragement factor to cover tract r with q level, $q = 1, 2$,

$\beta_{rl} \geq \beta_{r2}$

T_{is}: The predicted travel time for vehicle i to arrive at EMS station s when departing at the current time

$\psi_{sr}=$ if driving time from station s to tract r is less than the desired time window

$\alpha=$ The ratio of total number of tracts which need to be covered.

Decision Variables:

$z_{is} = 1$ if vehicle i is assigned to EMS station s; 0 otherwise

$\nu_{rq} = 1$ if census tract r is covered by at least q ambulances; 0 otherwise. $q = 1, 2$

η : Penalty variable for the case when percentage of covered tracts is less than α; $0 \leq \alpha \leq 1$

$$\max \sum_{r=1}^{R} \sum_{q=1}^{2} \beta_{rq} \nu_{rq} - C \sum_{i=1}^{I} \sum_{s=1}^{S} z_{is} T_{is} - M\eta \tag{IV.1}$$

$$\sum_{s=1}^{S} \sum_{i=1}^{I} z_{is} \psi_{sr} \geq \sum_{q=1}^{2} \nu_{rq} \quad \forall r \in R \tag{IV.2}$$

$$\nu_{r1} \leq \nu_{r2} \forall r \in R \tag{IV.3}$$

$$\sum_{r=1}^{R} \nu_{r1} \geq \alpha R - M\eta \tag{IV.4}$$

The objective function in equation (IV.1) consists of four parts. In the first part, the model maximizes the census tracts covered. The weight of each tract is set to the expected number of calls in that tract. In the second part, the model maximizes the tracts which have double coverage. In the third part, the model minimizes the travel time for relocating between stations. In the last part, the

56

objective function is penalized with a relatively large number, $\boldsymbol{M}$, if the percentage of covered tracts is less than $\boldsymbol{\alpha}$

Constraint (IV.2) ensures that the number of ambulances covering tract $\boldsymbol{r}$ is at least one if $\boldsymbol{\nu_{r1}} = \boldsymbol{1}$ and two if $\boldsymbol{\nu_{r1}} = \boldsymbol{\nu_{r2}} = \boldsymbol{1}$. Constraint (IV.3) ensures that $\boldsymbol{\nu_{r1}}$ is greater than or equal to $\boldsymbol{\nu_{r2}}$. Constraint (IV.4) ensures that $\boldsymbol{\alpha}$ percent of tracts are covered; otherwise, the objective function is penalized with $\boldsymbol{M}$.

B   The Discrete Event Simulation Model

In this section, we describe a simulation model which is developed to analyze the optimization model. In addition to this, the simulation model can be used to compare the performance of alternate optimization models and also to tune some parameters in the optimization model. This model simulates the coverage of a city in different areas and also the availability and location of vehicles. The simulation model has an interaction with the optimization model and uses it for relocating vehicles; hence, the output of the optimization model will determine the subsequent deployment of vehicles.

The following assumptions are made in this simulation model:
- The vehicle service times follow an exponential distribution
- The closest vehicle is dispatched to serve a call
- One vehicle is needed per each call
- Calls are put in the queue when there is no vehicle available

The details of the simulation model are provided below.

Step 0. Read the Initial Data which reflects the current state of the system

While $\boldsymbol{time} \leq \boldsymbol{FinishTime}$

$\boldsymbol{NextTimeOfCall} = \boldsymbol{Ln(1 - Rnd())/(\sum \lambda) + CurrentTime}$

Set $\boldsymbol{EarliestTimeOfGettingFree}$ as the earliest time a vehicle-
becomes available

If $\boldsymbol{NextTimeOfCall} \leq \boldsymbol{EarliestTimeOfGettingFree}$ Then

57

$$CurrentTime = NextTimeOfCall$$

Get the tract of call i.e. $TractNo = \{k|\Lambda_k \leq (Rnd * \sum \lambda) \leq \Lambda_{k+1}\}$

If there is an available vehicle Then

Make the closest vehicle busy and set its busy time to $-Ln(Rnd)/\mu$;

Else

Put the patient in queue

EndIf

Else

Make the vehicle available

If there is any patient waiting in the queue Then

Assign the vehicle to the first in queue

Set its busy time to $-Ln(Rnd)/\mu$

EndIf

EndIf

If(ThereIsNeedtoRunOptimizer)

Run Optimization Model

EndIf

EndWhile

C   Simulation Model Results

In this section, we present the results of the simulation model. The simulation model can be used for tuning the parameters of the mathematical model. we used C# to implement the simulation algorithm. In the simulation model, we directly solve the optimization model using Lingo 14.0 which is called from the simulation model in C#.

The time simulation length is set to one week. As an input for our simulation model, we have considered a virtual square city with 625 square miles and 169 census tracts. In this virtual city, it is assumed that calls arrive according to a

Poisson distribution and β value for each tract is inversely proportional to its distance from the centroid of the city.[1].

In the first experiment, we tried to find the best C (travel time penalty for an idle vehicle). We first set the Threshold (the coverage threshold for triggering the optimization model) to be 0.8. We incremented the value C by two starting from one. We ran the simulation model 10 times using different random number seeds. We compared the percentage of calls which could not be reached within the desired time window for different scenarios. This desired time window is the same as the one used for calculating ψ_{sr} in the optimization model. Table IV.1 shows the percentage of missed calls for different Cs for ten different random numbers. By default, it is assumed while vehicles are traveling between stations for repositioning the are considered as unavailable and do not provide coverage. The last row in Table IV.1 considers the case when vehicles provide coverage at the destination station while repositioning.

In order to find the best C for our model, we follow a simple procedure. In this procedure, we eliminate any C value which has caused an increase in percentage of missed calls. Among the remaining C values, we select the one which has the lowest value of missed calls. Using this procedure, we obtained $C = 3$ as the best value for our optimization model when the threshold, α is set to 0.8.

As shown in Table IV.1 considering vehicles providing coverage at the destination station, on average, the compliance rate improves approximately by 4%.

We have also let the simulation model run using ten different random number seeds for $C = 3$. In all the instances, the percentage of missed calls decreased using our optimization model. Table IV.2 shows the results for the best C value found.

In another experiment, we found the ratios between coefficients of single coverage β_{r1} and double coverage β_{r2}. Simulation was used for determining the best combination of C and ratio of $ratio = \beta_{r1}/\beta_{r2}$. In this experiment the value

---

[1]We have tried the data of the virtual city to be similar to a medium sized city like Louisville, KY. The number of stations and tracts we have considered are close to those of Louisville, KY.

TABLE IV.1

Comparison of results of optimization model for different Cs given threshold=0.8

RN Seed	1	2	3	4	5	6	7	8	9	10
Wo Optimzation	0.1530	0.1586	0.1722	0.1533	0.1646	0.1601	0.1660	0.1496	0.1621	0.1565
C=1	0.1463	0.1415	0.1500	0.1479	0.1472	0.1648	0.1556	0.1471	0.1452	0.1313
C=3	0.1362	0.1359	0.1484	0.1366	0.1412	0.1437	0.1460	0.1336	0.1407	0.1470
C=5	0.1500	0.1367	0.1566	0.1391	0.1565	0.1605	0.1581	0.1268	0.1315	0.1492
C=7	0.1291	0.1356	0.1512	0.1479	0.1490	0.1580	0.1496	0.1341	0.1341	0.1455
C=9	0.1480	0.1395	0.1546	0.1528	0.1532	0.1457	0.1651	0.1380	0.1402	0.1575
C=11	0.1536	0.1437	0.1650	0.1504	0.1594	0.1691	0.1652	0.1393	0.1489	0.1520
C=13	0.1489	0.1410	0.1625	0.1526	0.1560	0.1573	0.1548	0.1637	0.1465	0.1617
C=15	0.1680	0.1588	0.1713	0.1565	0.1571	0.1622	0.1696	0.1432	0.1541	0.1533
C=17	0.1600	0.1560	0.1643	0.1539	0.1707	0.1643	0.1575	0.1578	0.1520	0.1536
C=19	0.1690	0.1699	0.1632	0.1595	0.1706	0.1674	0.1710	0.1690	0.1369	0.1758
C=21	0.1640	0.1483	0.1746	0.1605	0.1847	0.1711	0.1704	0.1652	0.1660	0.1665
C=23	0.1662	0.1572	0.1843	0.1731	0.1608	0.1751	0.1776	0.1668	0.1549	0.1804
C=25	0.1743	0.1549	0.1929	0.1764	0.1753	0.1790	0.1835	0.1541	0.1732	0.1626
C=27	0.1737	0.1807	0.1944	0.1777	0.1896	0.1796	0.1857	0.1916	0.1776	0.1884
C=29	0.1680	0.1637	0.2073	0.1692	0.1929	0.2020	0.1994	0.1787	0.1689	0.1737
C=31	0.1770	0.1821	0.2015	0.1752	0.1845	0.1975	0.1981	0.2054	0.1755	0.1885
C=33	0.1875	0.1787	0.1869	0.2032	0.1914	0.1975	0.2080	0.1728	0.1847	0.1882
Coverage in repositioning	0.1211	0.1136	0.1257	0.1159	0.1138	0.1198	0.1212	0.1346	0.1121	0.1192

TABLE IV.2

Results of best travel time penalty, C, found

RN Seed	11	12	13	14	15	16	17	18	9	20
WO Optimization	0.165	0.148	0.142	0.163	0.153	0.160	0.171	0.159	0.149	0.165
Optimization Model Results for $C = 3$	0.140	0.129	0.135	0.150	0.133	0.154	0.147	0.146	0.136	0.155

TABLE IV.3

Comparison of results of the optimization model and Gendreau et al.(2001)

RN Seed	1	2	3	4	5	6	7	8	9	10
Genderea:2001 Model	0.137	0.155	0.138	0.139	0.152	0.158	0.157	0.140	0.140	0.141
Proposed Model	0.112	0.133	0.119	0.116	0.120	0.126	0.126	0.113	0.117	0.132

of *ratio* changed from one to eight and C from one to 15. We ran five replications, and the combination of $ratio = 7$ and $C = 7$ resulted the best performance.

In a separate experiment, we compared our optimization to the one presented in Gendreau et al. (2001). In this experiment to avoid the effect of travel time penalty, we have set the travel time penalty to 0 (which is C in our optimization model and M_{jl}^t in Gendreau et al. (2001)). We have set α to 0.9 in both models (as suggested in Gendreau et al. (2001)). Table(IV.3) shows the results of both models. As shown in this table for all of the cases, the proposed model was able to provide more favorable solutions.

D Conclusions

In this chapter, we have considered the problem of real time relocation of EMS vehicles. We have introduced a new linear programming model for this problem. After that, we developed an event based simulation model to first tune the parameters of the mathematical model and second evaluate the benefit of applying the model. The simulation was able to set the mathematical model parameters. It was shown how setting different parameters can affect the performance of the system. The simulation results show that the model can significantly reduce the number of missed calls even if we assume that the vehicles cannot respond to any calls while repositioning. Additionally, when we compared the model with one of the well-known models in the literature (Gendreau et al. (2001)), it was shown that the proposed model can provide high quality solutions in the studied virtual city.

CHAPTER V

DISPATCHING EMS VEHICLES TO PATIENTS AND THEIR CHOICE OF HOSPITALS

A   Introduction

In this chapter, the problem of dispatching vehicles is considered in a more general way. Despite Chapter III in this chapter, the dispatcher cannot determine the hospital for an ambulance; hence, the EMS vehicles transport patients to the hospitals which they have requested.

The problem is to assign vehicles to patients. For the high priority patient, always the closest vehicle is dispatched to serve the patient. An ambulance serving a high priority patient cannot serve any other patient. On the other hand, an ambulance serving a low priority patient can serve more than one patient.

The following assumptions are made:

1. An idle ambulance can serve any patient.

2. An ambulance can transport the patient to a hospital which the patient has requested.

3. An ambulance which has been rerouted cannot be rerouted any more until it is finished with the current task.

4. An ambulance en route to a high priority patient or transporting a high priority patient cannot be rerouted.

5. An ambulance serving a low priority patient can serve more than one patient.

6. Routing software packages can provide us the route between each pair of

nodes.

7. Each patient has a desired time window to be reached by an ambulance.

8. Each patient has a desired time window for spending in ambulance to be transported to hospital.

Unlike Chapter III, the high priority and low priority patients are not considered in one mathematical model, and each of these patients is considered in a separate mathematical model.

In other words

1. For the high priority patient: Always the closest vehicle is dispatched

2. For the low priority patient:

2.1. When the number of busy vehicles is less than a threshold: a vehicle is dispatched which can reach the patient in his/her desired time window and its dispatch has the least effect on coverage.

2.2. When the number of busy vehicles is more than a threshold: a vehicle is dispatched which is idle or transporting a low priority patient while ensuring that it can satisfy the time window constraints. Otherwise, it will be treated similar to 2.1

Considering the above scenarios, 1 and 2.1 are relatively easy problems to solve, however the problem in 2.2 is harder to solve. In the remainder of this chapter we will describe the methods to solve this case.

B   Problem Description and Mathematical Formulation

The EMS dispatching problem (EMSDP) assigns vehicles to patients so that the total travel distance is minimized while the times that the vehicle arrives at the patient scene and delivers the patient to the hospital are within specified time windows. We categorize patients as low priority and high priority patients, and assume that an ambulance serving a low priority patient can serve one more patient. Thus, vehicles that are idle, or transporting a low priority patient, or en route to serve a low priority patient are considered in our model. In other words, we exclude

those vehicles that are en route to high priority patients or are transporting them.

Specific assumptions in the model are as follows:

- An idle ambulance can serve any patient.

- An ambulance transports the patient to a hospital that is pre-determined either by patients or EMS staff.

- An ambulance serving a high priority patient cannot serve another patient.

- An ambulance can be rerouted after it finishes its current task.

- An ambulance serving a low priority patient can serve one more patient.

- Routing software packages can provide us the route between each pair of nodes.

- Each patient has a desired time window to be reached by an ambulance.

- Each patient has a desired time window to finally arrive at the hospital.

In the literature, Krumke et al. [49] study the vehicle dispatching problem with at most two requests (VDP2), which simply dispatches vehicles to serve requests with each vehicle serving at most two requests, so that the total transportation cost is minimized. Note that unlike a k-customer VRP (e.g., Haimovich et al. [36]), all vehicles in VDP-2 are geographically dispersed and thus are not necessarily based at one depot. Furthermore, Krumke et al. [49] how that the VDP2 is NP-complete. In analyzing the computational complexity for the EMSVPTP, we demonstrate that essentially VDP2 is reducible to the EMSVPTP in polynomial time.

Below, we first formally introduce the VDP2 in [49], and then show that the vehicle dispatching problem with two requests and a common destination depot

(VDP2CDD), a slight variation of VDP2, is NP-complete. Finally we establish the proof that the EMSVPTP is NP-complete through the reduction from VDP2CDD.

Vehicle Dispatching Problem with two Requests (VDP2) ([49])
Given a set of requests R, a set of vehicles U ($|U| \leq 2|R|$), cost function $c : (U \cup R) \times (U \cup R) \mapsto R^+$, the vehicle dispatching problem with two requests (VPD2) finds a tour $t_u = (u, r_{u,1}, r_{u,l(u)})$ for each vehicle $u \in U$ that serves $l(u) \leq 2$ requests, such that each request is served in exactly one tour and the total cost of all tours is minimized.

Vehicle Dispatching Problem with Two Requests and a Common Destination Depot (VDP2CDD) Given a set of requests R, a set of vehicles U ($|U| \leq 2|R|$), a common destination depot d, cost function $c : (U \cup R) \times (U \cup R \cup \{d\}) \mapsto R^+$, the vehicle dispatching problem with two requests and a common destination depot (VDP2CDD) finds a tour $t_u = (u, r_{u,1}, .., r_{u,l(u)}, d)$ for each vehicle $u \in U$ that serves $l(u) \leq 2$ requests, such that each request is served in exactly one tour, all tours end at d and the total cost of all tours is minimized.

The VDP2CDD is NP-hard.

Proof. Consider any VDP2 instance (R, U, c). Construct a VDP2CDD instance (R, U, c, d), where $c(r, d) = a \neq 0$ for all $r \in R$. As described, the construction of VDP2CDD can be done is polynomial time. In addition, any optimal solution to VDP2 solves the above constructed VDP2CDD and vice versa. In Krumke et al. [49], it is shown that the decision version of the VDP2 is NP-complete, thus VDP2 is NP-hard. Therefore, VDP2CDD is NP-hard. ∎

To establish the complexity proof for the EMSVPTP, next we restate the EMSVPTP as follows.

EMS Vehicle Patient Transportation Problem (EMSVPTP)

Given a set of patients P and a set of their corresponding requested hospitals $H = \{h(p) | \forall p \in P\}$ ($|H| = |P|$), a set of vehicles V ($|V| \leq 2|P|$) each with a capacity of k patients, cost function $s : (V \cup P \cup H) \times (V \cup P \cup H) \mapsto R^+$, the EMS vehicle patient transportation problem (EMSVPTP) finds a tour $t_v = (v, p_{v,i_1}, p_{v,i_2}, \cdots, p_{v,i_{l(v)}}, h_{v,i_{l(v)+1}}, h_{v,i_{l(v)+2}}, \cdots, h_{v,i_{2l(v)}})$ for each vehicle $v \in V$ that serves at most $l(v) \leq k$ patient requests, i.e., picks up each patient and drops each patient off at the requested hospital, such that each patient is served in exactly one tour and the total cost of all tours is minimized.

Note that in the above definition for EMSVPTP, $(i_1, i_2, \cdots, i_{2l(v)})$ is a permutation for $\{1, 2, \cdots, 2l(v)\}$. Further, if patient p_0 is the i_mth stop for vehicle v in its tour and its associated hospital $h_0 = h(p_0)$ is the i_nth stop for the vehicle, i.e., $p_0 = p_{v,i_m}$, $h_0 = h_{v,i_n}$ and $h_0 = h(p_0)$, then $i_n > i_m$.

Below, Theorem B establishes the NP-hardness result for the EMSVPTP with $k = 2$ from the reduction of the VDP2CDD problem in Lemma B in polynomial time.

The EMS patient transportation problem with $k = 2$ is NP-hard.

Proof. Consider any instance of VDP2CDD (R, U, c, d). To construct an equivalent EMSVPTP with $k = 2$, let $V = U$, $P = R$, $s = c$, and $H = \{d, d, ..., d\}$, i.e., all of destination hospitals are at the same location. Clearly, the EMSVPTP(V, P, H, s) problem can be constructed in polynomial time.

Let $t^* = \{t_u^*\}_{u \in U}$ be an optimal solution to VDP2CDD. Then, for each vehicle u, the optimal tour serves either one or two requests.

(i) If $u \in U$ serves one request , i.e., $t_u^* = (u, r_{u,1}^*, d)$, then the corresponding solution to the EMSVPTP with $k = 2$ is $w_v^* = (v, p_{v,1}^* = r_{u,1}^*, h = d)$.

(ii) If $u \in U$ serves two requests , i.e. $t_u^* = (u, r_{u,1}^*, r_{u,2}^*, d)$, the corresponding solution to the EMSVPTP with $k = 2$ is $w_v^* = (v, p_{v,1}^*, p_{v,2}^*, h = d)$.

Note that the (ii) is true because both requests have the common destination. We show that $\{w_v^*\}_{v \in V}$ is optimal to the above-constructed EMSVPTP with $k = 2$ by contradiction. If there exists a solution $\bar{w} = \{\bar{w}_{v \in V}\}$ with total cost $\sum_v \bar{s}_v \leq \sum_v s_v^*$, then solution $\bar{w}$ will induce an alternative solution $\bar{t}$ using (i)-(ii) with lower cost for VDP2CDD. This contradicts that t^* is an optimal solution to VDP2CDD. Thus, any optimal solution to VDP2CDD solves the above constructed EMSVPTP. Similarly, it can be shown that any optimal solution w^* to the EMSVPTP with $k = 2$ solves the VDP2CDD. Thus, the VDP2CDD is equivalent to the so constructed EMSVPTP. From Lemma B, EMSVPTP with $k = 2$ is NP-hard. ∎

Due to the above complexity results for EMSVPTP, we focus on developing efficient heuristic methods for solving EMSVPTP in a real-time decision support system. Because the current paper deals with the EMSVPTP with $k = 2$ as a prototype problem, in subsequent sections we refer to the EMSVPTP with $k = 2$ simply as EMSVPTP without explicitly specifying $k = 2$.

C   Mathematical Formulation

Given the above problem description and assumptions, we model the EMSDP as a VRPPD. In this problem, patients' requests are considered as pickup nodes and the hospital associated with each patient request is defined as a delivery node. Similar to VRPPD, patients (pick up nodes) are required to be visited prior to associated hospitals (delivery nodes). Several characteristics unique to EMSDP are as follows. First, in EMSDP each ambulance can service at most two patients. Second, if different patients (pick up nodes) choose the same hospital, we treat these hospitals as different delivery nodes. In the other words, each request is associated with a unique combination of pick-up and delivery nodes. Third, if an ambulance is

already transporting a patient, we consider the location of that patient (a request node) to be the same as the location of the ambulance. Finally, similar to VRP, each ambulance starts from a depot and ends at a depot. In EMSDP, we assume the "virtual origin depots" to be at the current locations of ambulances, and the "virtual destination depots" to be the last hospital visited by that the ambulances would go. In such a setting, the travel time from hospitals to the destination depots is zero.

The objective is to minimize the travel time for EMS vehicles. Also, we ensure that each patient is served in a desired time window by putting on visiting pickup and delivery nodes. The time window for a pick up node is regarded as a time window which a patient has to be served in; likewise the time window for a delivery node is a time window which within that a patient has to be dropped off at the hospital. Hence, we do not consider the patient waiting time in our objective function and consider their requirements as our hard constraints.

In order to formulate the EMSCP as linear integer program, we define the following sets and parameters:

Sets:

$V = V_P \cup V_H$	Set of vehicles
V_P	Set of vehicles which are idle or en route to serve patients
V_H	Set of vehicles en route to hospitals
P	Set of patient locations
H	Set of hospitals requested by patients
$N = P \cup H$	All nodes including patient locations and hospitals
O	Set of origin points of vehicles which are tied to vehicles
D	Set of final destination of vehicles.

Parameters:

t_{ij}	The driving time between points i and j
ψ_j	The desired time by which point j must be visited by an ambulance
$Ord(k)$	Order of vehicle k in set of V_H
n	The total number of patients including those in ambulances
q	The number of patients who have not yet been picked up
$o(k)$	Location of vehicle k
$d(k)$	Final destination of vehicle
$i(k)$	Patient that vehicle k is transporting

Let binary variable x_{ijk} be 1 if vehicle k goes from location i to j and 0 otherwise. In addition, we use T_{ik} to represent the time that vehicle k arrives at i.

Using the above notation, we formulate the EMSDP as the following mixed integer program.

$$\text{(EMSDP)} \quad \min \quad \sum_{ij \in N} \sum_{k \in V} t_{ij} x_{ijk} \tag{V.1}$$

$$\text{s.t.} \sum_{i \in N} \sum_{k \in V} x_{ijk} = 1, \quad \forall j \in P \tag{V.2}$$

$$\sum_{j \in N} x_{ijk} - \sum_{j \in N} x_{j(n+i)k} = 0, \quad \forall i \in P, \; \forall k \in V \tag{V.3}$$

$$\sum_{j \in P} x_{o(k)jk} = 1, \quad \forall k \in V^P \tag{V.4}$$

$$x_{o(k)i(k)k} = 1, \quad \forall k \in V^H \tag{V.5}$$

$$\sum_{i \in N \cup o(k)} \sum_{j \in P} x_{ijk} \leq 2, \quad \forall k \in V \tag{V.6}$$

$$\sum_{i} x_{ijk} - \sum_{i} x_{jik} = 0, \quad \forall j \in N, \; \forall k \in V \tag{V.7}$$

$$T_{ik} + t_{i,n+i,k} \leq T_{n+i,k}, \quad \forall i \in P, \; \forall k \in V \tag{V.8}$$

$$T_{i,k} \leq \psi_i, \quad \forall i \in N, \forall k \in V \tag{V.9}$$

$$T_{jk} \geq T_{ik} + t_{ijk} - M(1 - x_{ijk}), \quad \forall i,j \in N, \forall k \in V \tag{V.10}$$

Equation (V.1) minimizes the total travel time for vehicles transporting low priority patients to arrive at hospitals. Note that these vehicles will not serve any other patients. Constraint (V.2) states that each patient is to be served by a vehicle. Constraint (V.3) states that each vehicle must transport the patient to the designated hospital for the patient. Constraint (V.4) states each en-route vehicle is to be assigned to a patient. Constraint (V.5) ensures that a vehicle en-route to hospital is assigned to its patient. One important key point in this model is related to those patients which are already on board and being transported to a hospital. In order for our model to ensure that these patients are served by the same vehicle, we have defined $i(k)$ which in constrain V.5 it is ensure that these patients are served by the same vehicle. Constraint (V.6) states that a vehicle can serve maximum two patients. Constraint (V.7) states any vehicles going to a patient or a

hospital must depart from that location. Constraint (V.8) states that the time at which a vehicle arrives at a patient location must be less than by which the time that the vehicle must arrive at the designated hospital. Constraint (V.9) states that each node has to be visited in the desired time window. Finally, constraint (V.10) calculates the time that a vehicle visits either a patient or a hospital.

D   Obtaining Lower Bound

In this section we present a column generation approach for obtaining good lower bounds for the EMSVPTP. We first present the basics of applying the column generation technique to the general VRP similar to many works in the literature (e.g., Desrochers et al. (1992) , Xu et al.(2003), Fillet (2004) and Ropke and Cordeau (2009). Then we discuss a special pricing algorithm for solving the sub problem in the column generation method for the EMSVPTP.

1   Preliminaries

In the literature, many have used the column generation technique to solve the VRP through a reformulation of the set covering problem (e.g., Xu et al.(2003). When considered as a set covering problem, the vehicle routing problem is to determine the set of routes with the minimum total travel cost given the entire set of all possible routes.

More formally, let $a_{iku} = 1$ if route u visits node i by vehicle k and 0 otherwise, and c_{ku} be the cost of route u for vehicle k. In addition, let Ω be the set of all feasible routes and $\Omega_1 \subseteq \Omega$ be an arbitrary subset of Ω. Suppose decision variable $\theta_{ku} = 1$ if route u is served by vehicle k and 0 otherwise. Then, our problem (which we have shown that it is a VRP) can be written compactly as a set

covering problem as follows:

$$\text{(VRP-SC)} \quad \min \quad \sum_{k \in V} \sum_{u \in \Omega} c_{ku} \theta_{ku} \quad\quad (\text{V.11})$$

$$\text{s.t.} \quad \sum_{k \in V} \sum_{u \in \Omega} a_{iku} \theta_{ku} \geq 1, \quad \forall i \in N \quad\quad (\text{V.12})$$

$$\sum_{u \in \Omega} \theta_{ku} \leq 1, \quad \forall k \in V \quad\quad (\text{V.13})$$

$$\theta_{ku} \in \{0, 1\}. \quad\quad (\text{V.14})$$

Note that in (VPR-SC) the time window constants have already been addressed in defining feasible routes. The column generation approach starts with a subset of routes Ω_1 to solve the following (restricted) master problem of (VRP-SC).

$$\text{(MP)} \quad \min \quad \sum_{k \in V} \sum_{u \in \Omega_1} c_{ku} \theta_{ku} \quad\quad (\text{V.15})$$

$$\text{s.t.} \quad \sum_{k \in V} \sum_{u \in \Omega_1} a_{iku} \theta_{ku} \geq 1, \quad \forall i \in N \cdots \lambda_i \quad\quad (\text{V.16})$$

$$\sum_{u \in \Omega_1} \theta_{ku} \leq 1, \quad \forall k \in V \cdots \omega_k \quad\quad (\text{V.17})$$

$$\theta_{ku} \in \{0, 1\}, \quad \forall k \in V. \quad\quad (\text{V.18})$$

When solving the linear relaxation of the master problem (MP), we can obtain a lower for our problem. (It is a lower bound not the optimal solution because linear relaxation of the problem is solved). Let λ_i and ω_k be the dual variables associated with constraints (V.17) and (V.18), respectively. Then, the subproblem of minimizing the reduced cost for all routes $u \in \Omega$ is solved. If the minimum reduced cost is zero, then the optimal route is found; otherwise, we update Ω_1 by including the reduced-cost minimizing route and resolve the master problem (MP). Therefore, many refer to the subproblem as the "pricing" problem. Rather compactly, the subproblem can be written as:

$$\text{(PP)}: \quad \min_{u \in \Omega} \{- \sum_{i \in N} a_{iku} \lambda_i + \omega_k + c_{ku}\}$$

Let $b_{ijku} = 1$ if route u uses vehicle k to visit arc (i, j) and 0 otherwise. Then, $a_{iku} = \sum_{j \in N} b_{ijku}$ and

$c_{ku} = \sum_{(i,j) \in (N \cup O) \times N} b_{ijku} t_{ij} = \sum_{i,j \in N} b_{ijku} t_{ij} + \sum_{j \in N} b_{o(k)jku} t_{o(k)j}$. Thus, the objective function in (PP) can be simplified as

$$c_{ku} - \sum_{i \in N} a_{iku} \lambda_i + \omega_k = \sum_{i,j \in N} b_{ijku}(t_{ij} - \lambda_i) + \sum_{j \in N} b_{o(k)jku} t_{o(k)j} + \omega_k. \quad \text{(V.19)}$$

In other words, the travel cost between nodes is $(t_{i,j} - \lambda_i)$ for $i, j \in N$ and $(t_{o(k),j} + \omega_k)$ for $(o(k), j) \in O \times N$. Therefore, the subproblem (PP) can considered as an elementary shortest path problem with resource constraints (ESPPRC) (see e.g., Desrochers et al(1992) and Ropke and Cordeau (2009), where the resource constraints correspond to the time window constraints in the underlying route selection.

2 A Labeling Method for Solving the Pricing Problem

In order to solve the pricing problem (PP) as an elementary shortest path problem with resource constraints, we propose a method similar the Dijkstra algorithm (e.g., Fillet (2004). Specific to our EMS vehicle patient transportation problem are the labels associated with each node:

Label	Description
l.VisitedPatients:	patients who have been picked up in route u
(either delivered or not delivered)	
l.NotDeliveredPatients:	patients who have been picked up in route u but not delivered yet
l.Cost:	reduced cost of the route calculated by (V.19)
l.Time:	time of the route

In addition, several domination rules are used to facilitate efficient computing of the shortest paths. Particularly, route l_1 dominates l_2 if: 1) l_1.Cost $\leq l_2$.Cost;

and 2) l_1.Time $\leq l_2$.Time; and 3) l_1:VisitedPatients $\subseteq l_2$:VisitedPatients; and 4) l_1.NotDeliveredPatients $\subseteq l_2$.NotDeliveredPatients.

Furthermore, the shortest paths are dynamically extended from the origin, i.e., EMS vehicle's current location, to the destination, i.e., hospitals for the last patient. We develop an "Extend" function to perform this task. Particularly, the "Extend" function of route u to node i returns four labels (Cost, Time, VisitedPatients, and NotDeliverdPatients) for node i if the extension to node i is feasible with respect to time window constraints, or nothing is returned otherwise. Let j b the last node visited by route u. When extending route u to node i, Cost label is incremented by $(t_{i,j} - \lambda_i)$, and Time label by $(t_{i,j})$. If node i is a patient node, then it is added to VisitedPatinet and NotDeliveredPatient sets.If node i is a hospital node, its associated patient is deleted from NotDeliveredPatient label. After that, the feasibility of route is examined. $l.Time \leq TimeWindow(i)$ and not more than two patients nodes cannot be in VisitedPatient label. If these criteria is met then route u is extended to node i; otherwise null is returned.

Let L and NL be the set of all labels and the set of all labels with negative reduced costs, respectively. Then, using the above labels, domination rules and "Extend" function, the labeling method for the pricing problem (PP) can described in Subroutine 4 below.

Subroutine 4. The Labeling Method for (PP)

$u_{candidate} = \mathcal{F}(\lambda_i, \omega_k)_{i \in N, k \in V}$

Step 0. (Initialize)

    Create label associated with each node $i \in N$

    $L = O$

    For all $l \in O$ { $l.NotDeliveredPatients = 0, l.Time = 0, l.Cost = 0, l.VisitedPatients = 0$ }

Step 1. Find a label with the smallest Cost value.

    Let the label associated with that be l and the node associated with l, be j

Step 2. Scanning label l

    Step 2.1) Delivering Patients

    If ($l.NotDeliveredPatients$ is Empty) **Then**

get the associated Cost with that.

    If ($l.Cost < 0$) **Then** $NL = NL \cup l$

  Else

   For all $i \in l.NotDeliveredPatients$

     Extend label l by going to node h_i (delivering i) .

     Compare the new label l_{h_i} with all labels of node h_i, L_{h_i}.

     If l_{h_i} is dominated by a label in L_{h_i} , then l_{h_i} is deleted.

     Delete all of the labels in L_{h_i} dominated by l_{h_i}

   EndFor

  EndIf

  Step 2.2) Picking up Patients

  If ($|l.VisitedPatients| \leq 2$ **Then**

    For all ($i \in P$)

     If ($i \notin l.VisitedPatients$) **Then**

      Extend label l by going to node i .

      Compare the new label l_i with all labels of node i, L_i.

      If l_i is dominated by a label in L_i , then l_i is deleted.

     EndIf

    EndFor

  EndIf

Step 3. Let $L = L \setminus l$.

If ($L = \emptyset$) **Then**

  STOP.

 Else

  go to Step 1.

 EndIf

Step 4. Return the route associated with the label with most negative cost in NL.

E   Conclusions

In this chapter, we consider EMS dispatching problem during an emergency that minimizes the total travel distance by all vehicles while ensuring all patient requests are served with pre-specified time constraints. In order to better utilize the

EMS vehicles' capacity, we propose that each EMS vehicle can serve at most two patients. We present a mixed integer program formulation for the EMSDP problem, similar to the vehicle routing problem with pickup and delivery. We have designed an algorithm for calculating lower bound using column generation method.

CHAPTER VI

HEURISTIC ALGORITHMS FOR DISPATCHING AND RELOCATING EMS VEHICLES

A Introduction

In this chapter, we present two heuristic algorithms for solving the mathematical problems in chapters IV and V. First, we present a heuristic for solving a coverage problem (presented in Chapter IV). After that a heuristic is presented for solving the dispatching problem. We present the results for each of these heuristics at the end of each section.

B Heuristics for The EMS Vehicle Coverage Problem

In order to solve the above model, the following algorithm has been designed. We have tried a variety of heuristic search algorithms for solving this coverage problem. In here, we only present an algorithm which performed the best. This algorithm has two modules. The first one generates an initial solution. And , the last one is a simulated annealing.

1 Obtaining an Initial Solution

In order to obtain an initial solution, at the first, ambulances are assigned to the closest home station.

In this module, idle vehicles which have not been assigned to any patient will keep their last destinations and if they dont have any destination, the closest EMS station is assigned to them. This algorithm is described as follows.

Step0. Set all of z_{is} s equal to 0.

Step 1. For idle vehicles, keep their destination i.e. $\forall i \in V_E$ and $\sum_{s=1}^{S} z_{is} = 1$ set $z_{is} = z_{is}^0$.

Step 2. Assign vehicles which are idle and dont have any destination, to the closest EMS station i.e. If $i \in V^E$, and $\sum_{s=1}^{S} z_{is}^0 = 0$, then : set $z_{is} = 1$ such that $T_{is} = MinT_{is}|s = 1, .., S$

2 A Simulated Annealing Algorithm

In this module, an ambulance is randomly assigned to an EMS station. This is the last search module which tries to escape from a bad local optimum which the previous greedy algorithm might have found. This module is described as follows.

Randomly, one home station and one ambulance are selected. Then if there is an improvement from the last solution it is saved otherwise with the probability of $\exp^{(-differenceintheobjfn)/T}$ the new solution is accepted. Total number of iteration is almost one twentieth of all of the possible solutions for coverage problem. The SA module is shown in below. In the algorithm, the total number of iterations is dependent upon the number of vehicles. A variable named **MaxIteration** is defined set to $int((I+1)/5) * 20,000$. The total number of iteration in each temperature is set to one tenth of **MaxIteration**

Step 0: Set the z_1=(Current Solution) ; set incumbent=z1 ; T=T(0) and Counter=0.

Step1: Set N(T)=0.

Step2. If N(T)¿ **MaxIteration** go to Step7.

Step3: Randomly select an ambulance and a base and allocate that ambulance to that base and let this solution be z_2.

Step4. Calculate the difference in the objective between z_1 and z_2 and set it equal to dif. $(d = f(z_2) - f(z_1))$.

Step5. Generate a Random Number = RND.

Step6: If (dif¡0) or (RND¡$e^{-dif/T}$) then $z_1 = z_2$.

Step7: If (dif¡0) Set the incumbent= z_1.

Step8: N(T)=N(T)+1 and Counter= Counter+1 and go to Step 2.

Step9. If Counter $\geq$ $MaxIteration$/10, go Step 11 otherwise go to Step 10.

Step 10: T=0.9*T and go to Step1.

Step 11. Go to Step 8 in Module 3.1.

3 Numerical Results

Visual Basic is used to implement the simulated annealing heuristic, and Lingo called from C# are used to model and solve the mixed integer. In order to evaluate the proposed simulated annealing heuristic for the real time coverage problem, we have considered a virtual city. In our numerical experiment, all test instances are generated randomly in this virtual square city with the dimension of 50 miles by 50 miles. The location of patients and vehicles are chosen based on uniform distribution within the city. We have considered 23 stations which their locations remain the same in all instances.

As shown in Table(VI.1) out of 60 instances, the SA algorithm was able to find the optimal solution in 58 instances. The results show that the SA can be successfully implemented in medium size departments when the number of vehicles and stations are around the values in Table(VI.1)

C   Heuristics for Dispatching EMS Vehicles to Patients and Their Choice of Hospitals

1   A Simulated Annealing Algorithm

This section develops a meta-heuristic method for the EMSVPTP using the simulated annealing approach. We define a pairwise exchange type of neighborhood

TABLE VI.1

Comparison of results of optimization model with simulated annealing for Coverage
Problem

Instance No	No of Vehicles	SA	IP Model	Gap	Instance No	No of Vehicles	SA Model	IP	Gap
1	30	15066.78	15066.78	0	31	40	15004.35	15004.35	0
2	30	15013.11	15038.29	0	32	40	15002.28	15002.28	0
3	30	15069.53	15069.53	0	33	40	14996.29	14996.29	0
4	30	15045.89	15045.89	0	34	40	14960	14997.02	0
5	30	15064.74	15064.74	0	35	40	15013.21	15013.21	0
6	30	15074.25	15074.25	0	36	40	14828.55	15015.28	0.01
7	30	15043.75	15043.75	0	37	40	14995.85	14995.85	0
8	30	15045.05	15045.05	0	38	40	14973.84	14973.84	0
9	30	15073.52	15073.52	0	39	40	15020.13	15020.13	0
10	30	15064.93	15064.93	0	40	40	14964.17	14964.17	0
11	30	15060.78	15060.78	0	41	50	14956.41	14956.41	0
12	30	15063.23	15063.23	0	42	50	14976.55	14976.55	0
13	30	15080.60	15080.60	0	43	50	14888.18	14948.90	0
14	30	15066.78	15066.78	0	44	50	14946.38	14946.38	0
15	30	15060.65	15060.65	0	45	50	14938.96	14938.96	0
16	30	15034.45	15084.20	0	46	50	14950.39	14950.39	0
17	30	14787.94	15037.85	0.02	47	50	14993.34	14993.34	0
18	30	15058.40	15058.40	0	48	50	14935.47	14935.47	0
19	30	15000.36	15054.32	0	49	50	14968.86	14968.86	0
20	30	15022.07	15022.07	0	50	50	14954.40	14954.40	0
21	40	15014.72	15014.72	0	51	50	14952.29	14952.29	0
22	40	15006.52	15006.52	0	52	50	14943.69	14943.69	0
23	40	14977.64	14977.64	0	53	50	14958.73	14958.73	0
24	40	15010.89	15010.89	0	54	50	14966.25	14966.25	0
25	40	14999.54	14999.54	0	55	50	14938.34	14938.34	0
26	40	14960.69	14960.69	0	56	50	14917.31	14917.31	0
27	40	15015.40	15015.40	0	57	50	14962.55	14962.55	0
28	40	14987.87	14987.87	0	58	50	14964.22	14964.22	0
29	40	15015.97	15015.97	0	59	50	14938	14938	0
30	40	14965.67	14965.67	0	60	50	14941.92	14941.92	0

function/search in Subroutine 2, followed by Subroutine 3 which describes how we optimally sequence the nodes visited by a vehicle once the selections of nodes are determined for all vehicles. Finally, Subroutine 4 conducts a specialized local search aiming to introduce more diverse solutions at later stage (with lower temperatures) of the SA procedure. We refer to the combination of this local search with the pairwise exchange neighborhood search as the "hybrid" method in this and subsequent sections.

In particular, Subroutine 1 uses $\{(n_i, n_j, p_i^1, p_i^2, p_j^1, p_j^2)\}_{i,j \in V}$ to denote a pair of patient assignments pertaining to vehicles i and j that are en route either to patient scenes or to hospitals, where n_i and n_j are the numbers of patients assigned to vehicles i and j, respectively. Note that if $i \in V^P$, then $1 \leq n_i \leq 2$, and if $i \in V^H$, then $0 \leq n_i \leq 1$. In addition, p_k^i and p_k^j for $k = 1, 2$ are the two patients served by vehicles i and j, respectively. Note that it is possible that $p_k^l = \emptyset$ for $k = i, j$ and $l = 1, 2$. Using this notation, the neighborhood function in Subroutine 1 performs pairwise swaps on patients in all currently busy vehicles. First, two vehicles are randomly chosen. Second, we randomly select one patient from each vehicle and swap them. Note that because a patient in a vehicle can be an empty set as described previously, such a swap has three possible outcomes: 1) swapping two patients between the two chosen vehicles when both selected patients are non-empty set; 2) moving one patient from one vehicle to the other when one patient is indeed an empty set and the other patient is not; 3) doing nothing when both patients are empty set. Subroutine 1 excludes all swaps with outcome 3). Finally, sequence optimization Subroutine 3 is performed after a random feasible swap.

Subroutine 1: The Pairwise Exchange Neighborhood Function

$$\bar{X} = (\bar{n}_i, \bar{n}_j, \bar{p}_i^1, \bar{p}_i^2, \bar{p}_j^1, \bar{p}_j^2)\}_{i,j \in V} = \mathcal{N}(X = \{(n_i, n_j, p_i^1, p_i^2, p_j^1, p_j^2)\}_{i,j \in V})$$

Step 0: (Initialize the neighbor solution)

$(\bar{n}_i, \bar{n}_j, \bar{p}_i^1, \bar{p}_i^2, \bar{p}_j^1, \bar{p}_j^2) = (n_i, n_j, p_i^1, p_i^2, p_j^1, p_j^2)$ for all $(i,j) \in V$.

Step 1: (Select the patients to swap from vehicles i and j)

  Loop $l = i, j$

   if $l \in V^P$, then choose randomly one patient index k_l =rand(1,2);

   if $l \in V^H$, then choose patient index $k_l = 1$;

  EndLoop

Step 2: (Swap selected patients in vehicles i and j)

  If $p_i^{k_i} \neq \emptyset$ and $p_j^{k_j} \neq \emptyset$, **Then** $\bar{p}_i^{k_i} = p_j^{k_j}$, $\bar{p}_j^{k_j} = p_i^{k_i}$);-

   -(if both patients are non-empty then swap them)

   Else (perform only two type of feasible swaps)

    Loop $a = i, j$

    $b = \{i, j\} \backslash \{a\}$;

     If $p_a^{k_a} = \emptyset$ and $p_b^{k_b} \neq \emptyset$, **Then** $\bar{p}_a^{k_a} = p_b^{k_b}$, $\bar{p}_b^{k_b} = p_a^{k_a}, \bar{n}_a = \bar{n}_a + 1, \bar{n}_b = \bar{n}_b - 1$);

    EndLoop

Step 3: (Optimize the routes in vehicles i and j)

  Loop $l = i, j$

   if l is idle or enroute to a patient find the best possible sequence from all possible

    six sequences(see Subroutine 2);

   if l is transporting a patient, find the best possible sequence from all possible three

    sequences(see Subroutine 2);

  EndLoop

Subroutine 2 OptimizeSequence determines the best sequence of all nodes to be visited vehicle k when it serves two patients. In this case, at most six possible sequences exist because there are, at most, two patients and two hospitals to be visited by a vehicle, and each patient is visited before the associated hospital. In particular, let $1 \leq \rho \leq 6$ be the sequence index, p_1 and p_2 be the two patients and h_1 and h_2 be their respective hospitals. Then sequences 1 through 6 are defined as follows. For example, in sequence 3 vehicle k visits patient p_1 first, followed by his/her hospital h_1, then patient p_2 and finally the his/her hospital h_2.

sequence index ρ	the actual sequence
1	(k, p_1, p_2, h_1, h_2)
2	(k, p_1, p_2, h_2, h_1)
3	(k, p_1, h_1, p_2, h_2)
4	(k, p_2, p_1, h_1, h_2)
5	(k, p_2, p_1, h_2, h_1)
6	(k, p_2, h_2, p_1, h_1)

Using this notation, Subroutine 2 examine all applicable sequences l through u ($l \leq u$) for vehicle k in serving patient j with X being the associated solution. If improvement is found, the subroutine will update X with the optimal sequencing. Note that for vehicle $k \in V^P$, all six sequences are feasible, hence $l = 1$ and $u = 6$. However, for vehicle $k \in V^H$, because only sequences 1 through 3 are feasible, $l = 1$ and $u = 3$.

Subroutine 2: OptimizeSequence($\bar{X}$, k, j, l, u)

  Loop $\rho = l, \cdots, u$ // l through u are the applicable sequence indices

    Calculate times to reach the first and second patient $g^1(\rho)$ and $g^2(\rho)$;

    Calculate times to transport the first and second patient to their respective hospitals-

      -$g^3(\rho)$ and $g^4(\rho)$;

    If $g^l(\rho), l = 1, 2, 3, 4$ are all feasible, **Then**

      Evaluate the new objective value associated with sequence ρ, i.e., Obj(ρ);

      If Obj(ρ) < BestObj **Then**

        Assign patient j to vehicle k using sequence ρ and update solution $\bar{X}$;

        BestX=$\bar{X}$;

        BestObj=Obj(ρ);

  EndLoop

 Return BestX

In addition to the pairwise exchange neighborhood search in Subroutine 1, we propose a nested vehicle search (NVS) algorithm to be used in later stages of the SA process when the temperature in the SA is relatively low. In essence, the NVS in

Subroutine 3 attempts to diversify the solution space when the SA process starts to concentrate its search. In Particular, starting with the current incumbent solution thus far in the SA process, the NVS first randomly select a patient (j) and cancels the assignment of patient p_1 to its current hospital h_1. Second, a local greedy search of vehicle is performed for this patient j. This search checks all vehicles with three possible scenarios. In the first scenario where vehicle i is assigned to only one patient, step 1.2.1 checks if patient j can be reached by vehicle i with required time windows. In addition, the sequence is optimized from six possible sequences as described in Subroutine 2. In the second scenario where vehicle i is transporting a patient to a hospital, step 1.2.2 checks all three possible sequence when performing sequence optimization. This is because the current patient has to be the first patient, thus making $\rho = 4 \rightarrow 6$ infeasible in Subroutine 2. In the third scenario where a vehicle i is assigned to serve two patients. Let j_1 be the immediate patient and j_2 the second patient, then step 1.2.3 checks if swapping current patient j with the second patient j_2 can lead to a better solution. In addition, step 1.2.3 examines if j_2 can be served by either a vehicle which is assigned to a patient (step 1.2.1) or a vehicle en-route to a hospital (step 1.2.2). If so, the swap will be accepted. After considering all three scenarios for patient j, the best assignment is obtained. This process repeats for all patients other than j before a best local solution is identified.

Subroutine 3: The Nested Vehicle Search Neighborhood Function
$$BestX = \mathcal{N}(\bar{X})$$

Step 0. (Initialize) Set the initial solution to the best solution obtained by SA

Step 1:// Assign vehicles to patients selected by SA based on a greedy selection

Loop $j = p_1, p_2$ going through patients selected by SA

    If $(j \neq 0)$ Then

        Step 1.1: Do not assign any vehicles to patient j

        Step 1.2: Select a vehicle for patient j using the following procedures

            Loop $k = 1, ..|V|$ //searching through all vehicles

                Step 1.2.1: //first, examine vehicles en route to a patient and not yet -

-being assigned to a second patient

      If Vehicle $k \in V^P$ and it is assigned to only one patient **Then**

          Obtain $\bar{X}$ resulted from assigning patient j to vehicle k using-

            -$\boldsymbol{OptimizeSequence}(\bar{\boldsymbol{X}}, \boldsymbol{k}, \boldsymbol{j}, \boldsymbol{1}, \boldsymbol{6})$

      EndIf

    Step 1.2.2: // second, examine vehicles en route to a hospital

      If Vehicle $k \in V^H$ and no patient is assigned to it **Then**

          Obtain $\bar{X}$ resulted from assigning patient j to vehicle k using-

            -$\boldsymbol{OptimizeSequence}(\bar{\boldsymbol{X}}, \boldsymbol{k}, \boldsymbol{j}, \boldsymbol{1}, \boldsymbol{3})$

      EndIf

    Step 1.2.3:// third, examine vehicles that are assigned to two patients-

      -with different requested hospitals,

      If $k \in V^P$ and it is assigned to two patients with two different-

        - hospitals **Then**

        Examine if the second patient of vehicle k can be served by a different -

          -vehicle k' using similar processes as in Steps 1.2.1 and 1.2.2;

        If a k' is found **Then**

           Obtain $\bar{X}$ resulted from assigning patient j to vehicle k-

             -using $\boldsymbol{OptimizeSequence}(\bar{\boldsymbol{X}}, \boldsymbol{k}, \boldsymbol{j}, \boldsymbol{4}, \boldsymbol{6})$

        EndIf

      EndIf

   EndLoop // going through vehicle k

    Step 1.3:

      If $\boldsymbol{obj} \leq$BestObj **Then**

        BestX=$\bar{\boldsymbol{X}}$, BestObj=$\boldsymbol{Obj}$;

      EndIf

   EndIf

EndLoop // going through patient j

Step 2. STOP

2 Numerical Results

In order to evaluate the proposed simulated annealing heuristic, we used C#
to implement the SA heuristic, as well as the mixed integer model for EMSVPTP

via the ILOG/Concert technology by CPLEX 12.4 [16]. The CPU times reported here are from a Dell Intel(R) W3503 Processor with 2.4 GHz and a 4GBRAM on a 64-bit operating system.

Preliminary pilot studies are conducted to fine tune the parameters in the proposed SA heuristic. From the pilot runs, the best parameter setting recommends: 1) the initial temperature $\boldsymbol{T_{init} = 20}$ which equates to an approximate 80% probability of accepting an uphill transition initially; 2) the choice of the exponential cooling function $\boldsymbol{T_k = T_0 \alpha^k}$ with $\boldsymbol{\alpha = 0.9}$; 3) the total epoch length $\boldsymbol{L = 10000}$ and total length $\boldsymbol{L_{total} = 100000}$.

In our numerical experiment, all test instances are generated randomly in a virtual square city with the dimension of 50 miles by 50 miles. The location of patients and vehicles are chosen based on uniform distribution within the city. We have considered the locations of hospitals the same in all instances.

We first compare the performance of the SA heuristic for EMSVPTP against that of the MIP model in Section V.C by CPLEX on 43 test instances. These test instances are of relatively small size, with the number of patients ranging between 5 and 35 and the number of EMS vehicles ranging between 5 and 15. Multiple instances are created for each combination of patient and vehicle sizes. The comparison results are summarized in Tables VI.3, which displays the number of patients and EMS vehicles for each group of instances in column 2, the objective function value and the computational time for SA in columns 3 and 4, and for CPLEX in columns 5 and 6. We have set the time limit for CPLEX to be 30 minutes. Also reported in Table VI.3, are the average objective function value and CPU time for SA and CPLEX. Note that because CPLEX experienced "out of memory" issue before termination for instance 34, we have excluded this instance when calculating the average CPU time and objective function value.

Overall, Table VI.3 suggests that from small size EMSVPTP instances, the average CPU time used by SA is 7.39 seconds, compared to 38.63 seconds for

CPLEX. On the other hand, the improvement on the objective function by the SA is about 0.016% (966.5-966.35/966.5). More specifically, in three of the 43 instances, the CPLEX solution of is better than those of SA solution. On the other hand, for 38 out of 43 examples both solvers found the same optimal solution. Among these instances, 21 experience less CPU time with the SA than with CPLEX. Thus, we conclude that the proposed SA algorithm is efficient and robust, because for most instances it was able to provide the same solution as CPLEX in much less CPU time.

In another experiment, we compared the result of the proposed heuristic with the results of the lower bound method described in Chapter V. Table VI.4 lists similar information to those in Table VI.3, with the exception that column 4 is a lower bound instead of an optimal objective value from CPLEX. Furthermore, column 7 provides the optimality gap for the SA solution using the lower bound from the column generation method. we generated 50 medium to large size EMSVPTP instances whose number of patients and vehicles range between 70 and 80, and between 55 and 95, respectively. All associated master and pricing problems for the column generation method were again solved by CPLEX via Concert technology using C#.As previously, the maximum CPU limit for CPLEX is set as 30 minutes. As shown in Table VI.4 the solution provided by SA is very close to the lower obtained by column generation method. The average gap of the solutions is 2.5% indicating that the SA algorithm was able to provide a solution very close to the lower bound. Thus, Table VI.4 shows that the proposed SA algorithm provides quality solution for large scale EMSVPTP instances.

Finally, we compare the results for the pure and hybrid SA methods. The pure SA method refers to the SA method using purely pairwise exchange neighborhood function in Subroutine 1, and the hybrid SA method refers to the one using the pairwise exchange neighborhood function in Subroutine 1 and the nested vehicle search neighborhood function in Subroutine 4. As mentioned in SA

algorithm, preliminary tests on the hybrid method suggests that the benefit of including the nested vehicle search neighborhood search is only significant after the temperature in SA is relatively low. Thus the hybrid SA method only includes the nested vehicle search when the algorithm is in final 5 % iterations.

Table VI.5 shows the results of this hybrid method. Overall, this table suggests that the hybrid method is able to provide better solutions for 11 out of 50 instances with an average improvement on the objective value of 0.4%. For instances 5 and 28, the improvement is larger than 1%.

D   Conclusions

In this chapter, we have designed two heuristic methods for solving coverage problem and dispatching problem independently. In the first heuristic, simulated annealing was used to solve the coverage problem which gives recommendation for relocating EMS vehicles in real time. Our results show that the algorithm can efficiently provide favorable solutions for medium-sized cities. In the second algorithm, a hybrid simulated annealing algorithm is developed to solve the dispatching problem. Our numerical results show that the proposed method is efficient in solving large-scale instances for EMSDP, when compared to the lower bound obtained in Chapter V.

TABLE VI.3

Comparison between the SA and the MIP model by CPLEX

Instance	(Patients,Vehicles)	SA Method		CPLEX	
		CPU time (seconds)	Objective function	CPU time (seconds)	Objective function
1		2	262.20	1	262.20
2		2	236.25	0	236.25
3	(5,5)	2	284.18	0	284.18
4		3	285.63	0	285.63
5		3	274.73	0	274.73
6		2	530.81	0	530.81
7		3	538.01	1	538.01
8	(15,5)	3	578.88	1	578.88
9		2	446.21	1	446.21
10		3	566.01	1	566.01
11		6	609.68	2	609.68
12		6	587.08	2	587.08
13	(15,10)	5	601.61	2	601.61
14		5	627.41	2	627.41
15		5	553.03	3	553.03
16		11	808.54	11	808.54
17		10	785.50	11	785.50
18	(15,15)	11	777.32	10	777.32
19		11	733.41	6	733.41
20		11	713.87	6	713.87
21		2	831.93	3	831.93
22	(25,5)	2	701.01	3	701.01
23		14	767.80	3	767.80
24		5	913.04	5	913.04
25	(25,10)	4	847.58	4	847.58
26		5	944.23	6	944.23
27		9	898.14	14	898.14
28	(25,15)	8	1030.70	19	1030.70
29		8	1000.25	18	1000.25
30		10	1114.78	50	1113.73
31	(25,20)	11	1059.30	404	1055.47
32		8	1059.88	27	1059.88
33		14	1183.19	371	1178.5
34	(25,25)	12	1109.18	out of memory	
35		12	1115.84	1800	1121.22
36		3	1150.65	8	1150.65
37	(35, 5)	3	1156.25	9	1156.25
38		3	1164.39	9	1164.39
39		9	1146.83	15	1146.55
40	(35,10)	4	1213.31	11	1213.31
41		6	1211.02	14	1211.00
42		7	1269.86	17	1269.82
43	(35,15)	8	1249.56	24	1249.56
Average (excluding instance 34,35)		7.39	966.50	38.63	966.35

TABLE VI.4

Comparison between the SA and the lower bound from column generation

Instance	(Patients,Vehicles)	Column Generation		SA		Gap
		CPU time (seconds)	Lower bound	CPU time (seconds)	Objectiven function	
1		58	2212.98	22	2257.01	0.02
2		66	2188.19	21	2249.69	0.03
3	(70,55)	40	2528.95	21	2560.59	0.01
4		36	2544.59	22	2594.91	0.02
5		75	2139.33	23	2275.72	0.06
6		77	2826.89	23	2875.12	0.02
7		117	2592.88	24	2674.12	0.03
8	(70,65)	71	2750.32	24	2808.20	0.02
9		126	2550.82	23	2623.34	0.03
10		81	2665.34	24	2707.24	0.02
11		139	3089.35	24	3152.08	0.02
12		143	2948.28	27	3006.09	0.02
13	(70,75)	253	2531.67	25	2686.45	0.06
14		144	3028.31	27	3118.06	0.03
15		230	2586.04	24	2672.26	0.03
16		230	3129.15	29	3208.77	0.03
17		381	2845.20	29	2929.64	0.03
18	(70,85)	219	3349.72	29	3387.80	0.01
19		426	2858.99	28	2991.63	0.05
20		378	2919.77	26	3008.02	0.03
21		697	3071.83	31	3138.05	0.02
22		666	2994.84	27	3094.19	0.03
23	(70,95)	413	3589.32	28	3684.37	0.03
24		684	3314.72	27	3404.28	0.03
25		681	3210.95	30	3280.08	0.02
26		84	2385.12	24	2422.17	0.02
27		51	2595.33	27	2669.51	0.03
28	(80,55)	67	2334.90	23	2400.11	0.03
29		58	2515.78	23	2551.03	0.01
30		31	2716.51	24	2732.64	0.01
31		106	2569.75	23	2605.09	0.01
32		71	2753.18	27	2768.36	0.01
33	(80,65)	111	2641.25	28	2690.82	0.02
34		74	2856.14	29	2904.54	0.02
35		100	2788.42	24	2828.41	0.01
36		173	2980.87	26	3085.75	0.04
37		225	2788.54	25	2849.62	0.02
38	(80,75)	174	2876.06	26	2889.58	0.00
39		179	3175.96	30	3234.47	0.02
40		171	3037.90	29	3119.75	0.03

Instance	(Patients,Vehicles)	Column Generation		SA		Gap
		CPU time (seconds)	Lower bound	CPU time (seconds)	Objectiven function	
41		408	3012.30	33	3088.57	0.03
42		435	2974.11	29	3067.08	0.03
43	(80,85)	452	2950.10	27	3042.16	0.03
44		263	3342.04	32	3409.64	0.02
45		303	3239.11	33	3324.23	0.03
46		483	3496.24	34	3532.93	0.01
47		375	3538.92	29	3623.35	0.02
48	(80,95)	476	3418.44	29	3512.85	0.03
49		691	3500.45	33	3572.43	0.02
50		490	3477.38	29	3573.89	0.03
	Average	249.64	2888.66	26.68	2957.74	0.024

TABLE VI.5

Comparison between the Pure SA and Hybrid SA

Instance	$(|V|,|P|)$	Pure SA	Hybrid SA	Instance	$(|V|,|P|)$	Pure SA	Hybrid SA
1		2535.99	2535.99	26		2603.40	2603.40
2		1594.66	1594.66	27		2202.94	2202.94
3	(70,55)	1854.27	1854.28	28	(80,55)	1820.79	1802.47
4		2138.83	2138.83	29		2378.62	2378.62
5		1996.92	1975.63	30		2038.20	2038.20
6		2393.18	2393.18	31		1609.04	1605.60
7		2555.89	2546.30	32		3333.48	3333.48
8	(70,65)	3151.53	3151.53	33	(80,65)	3080.37	3080.37
9		3134.20	3134.04	34		3310.50	3310.50
10		2372.09	2372.09	35		1836.93	1832.58
11		4151.72	4151.72	36		2753.34	2753.34
12		3563.92	3563.92	37		2135.84	2131.59
13	(70,75)	2133.42	2133.42	38	(80,75)	4107.14	4107.14
14		2141.11	2141.11	39		1989.81	1989.81
15		2707.39	2707.39	40		3294.01	3294.01
16		2297.93	2297.93	41		2945.17	2945.17
17		2114.90	2114.90	42		1808.49	1804.25
18	(70,85)	3382.58	3382.58	43	(80,85)	2403.39	2387.27
19		3646.00	3636.19	44		2513.34	2513.34
20		2651.87	2651.87	45		2853.63	2853.63
21		2570.54	2567.53	46		2856.95	2856.95
22		2946.79	2946.79	47		4387.25	4387.25
23	(70,95)	2693.41	2693.41	48	(80,95)	4658.17	4658.17
24		2571.92	2571.92	49		3942.90	3942.90
25		2438.35	2438.35	50		2635.10	2635.10

CHAPTER VII

THE EMS DISPATCHER APPLICATION IN A REAL TIME DECISION SUPPORT SYSTEM

A   Introduction

In this chapter, we present a real time decision support web application. The EMS Dispatcher module is a web based application developed in ASP.Net 4.0 technology, C# programming language and MS SQL Server 2008 R2. [MacDonald (2010), MacDonald, and Freeman(2010), and Rankins et al (2010)].

B   Application

This application is capable of performing multiple, important functions such as determining the three best EMS vehicles to serve a patient, committing one of them, maintaining an active case log, handling un-served patient requests and EMS vehicle status on a real-time basis. Figure VII.1 shows how this module can be used in a real time decision setting for the EMS dispatcher

When the EMS dispatcher clicks on any of the records (unserved patients) in Figure VII.2, the optimization algorithm discussed in the next section is triggered, an EMS routing problem is solved in real-time and the related information is displayed on the screen displaying the three alternative optimal solutions. Based on the information displayed on the screen, the dispatcher can select an ambulance to assign to a patient. These assigned patients are shown in the Active Case Log Table.(see Figure VII.3)

In order to help the dispatchers check the availability of the ambulances,
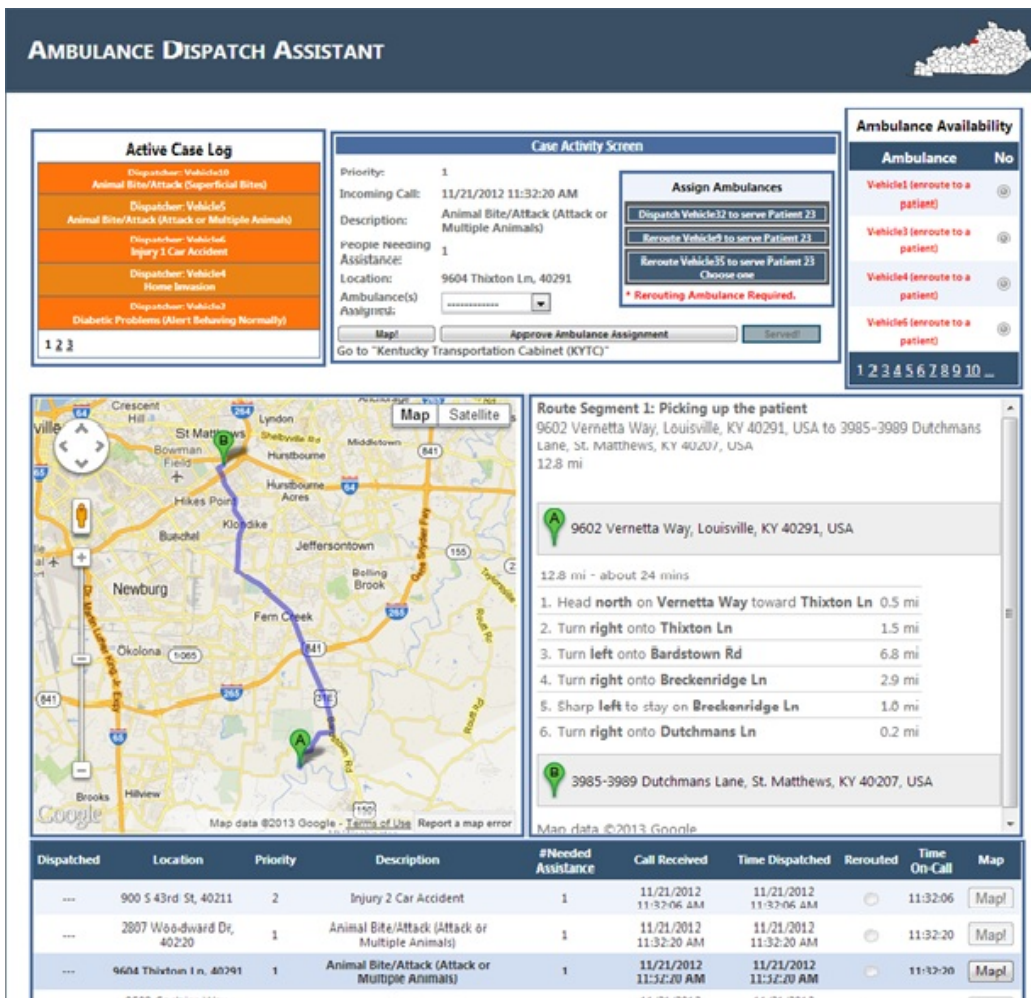
Figure VII.1. User interface for ambulance dispatch assistant

Figure VII.2. Current unserved patients

93

there is a list of ambulances and with related status. (see Figure VII.3)

By clicking on any records in Active Case Log Table, all the information on that patient will be displayed on Case Activity Screen and by clicking on Map button, the directions from the assigned ambulance to the assigned patient and for that patient to the assigned hospital will be shown on a Google Map.

When a patient is taken to the hospital, clicking on the Served button causes to assign the patient as served patient and that patient will be erased from unserved patient list on Active Log Case table. (see Figures VII.3 and VII.4)
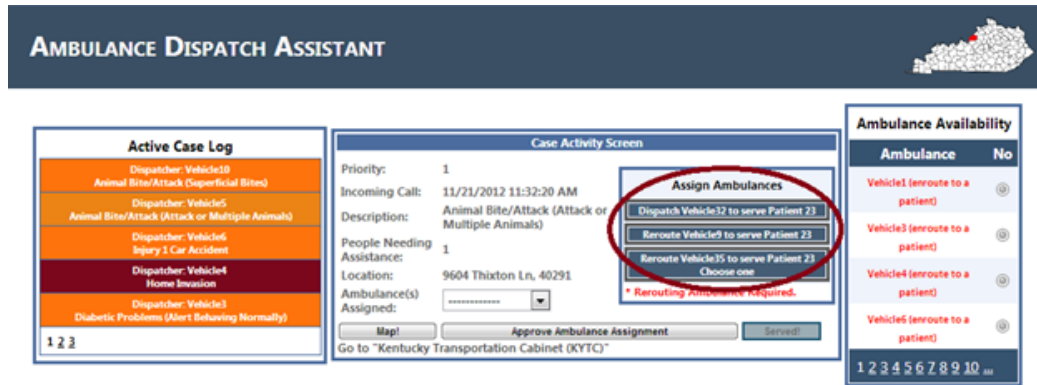

Figure VII.3. Assignment of EMS vehicles in real-time

C   Algorithm

This section describes an algorithm when serving more than one patient by an ambulance is permitted. This algorithm is a greedy algorithm that: 1) minimizes the travel time; 2) ensures that all patients are served within the required time window; and 3) minimizes the number of required routes.

The algorithm first sorts the patients who have not been served in a list named "Unservedpatients". Then, the first patient from this list is selected. After that, the first vehicle which is either transporting a low priority patient, or, enroute to service a patient, or assigned to serve two patients is considered. For each of these vehicle statuses, the algorithm considers three cases.
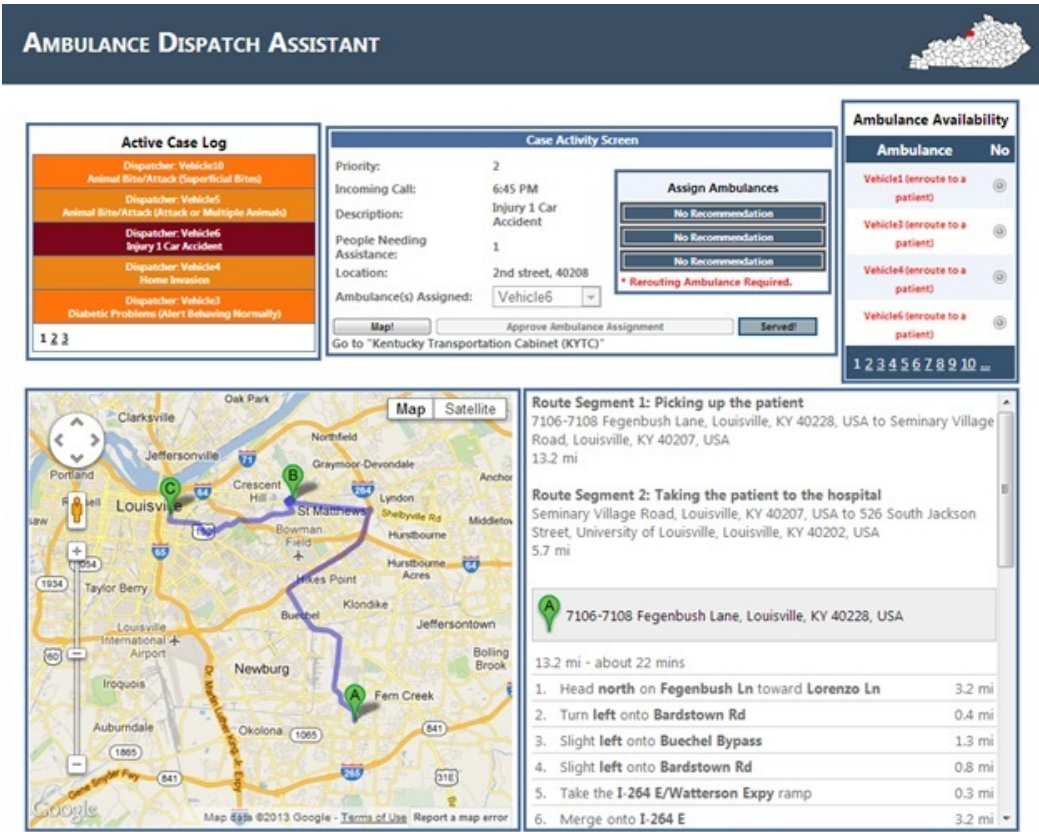
94

Figure VII.4. After Assignment of EMS vehicles

The first case is related to a vehicle transporting a low priority patient to a hospital. In this case it is examined if this vehicle can serve the patient while satisfying time window constraints. If so, the total travel time is calculated.

In the second case, vehicles that are en route to serve one patient are considered. The algorithm checks if the patient can be inserted to this vehicle's route. If time window constraints are satisfied, this vehicle can be another candidate and we calculate the total travel time.

In the third case, the algorithm considers a vehicle already assigned to two patients, but, has not yet served any of them. In this case, it is examined if the second patient of this vehicle can be served by another vehicle. If so, then the objective function for serving patient using this vehicle is calculated.

If a feasible solution is found using the above steps, a vehicle with the least objective function is assigned to the patient. Otherwise, an idle vehicle is dispatched which can serve this patient in the required time window and its dispatch has the least effect on the coverage. This procedure repeats until all patients in the UnservedPatient list have been added to routes.

CHAPTER VIII

CONCLUSIONS AND FUTURE RESEARCH

In this dissertation, we have considered the problem of dispatching and relocating vehicles from different aspects. We have considered the idea of serving two patients by an EMS vehicle for the first time for two cases: 1- when hospital selection is a decision variable 2- when hospital selection is an input. Additionally, we used a new approach for relocating EMS vehicles in real time.

At first, we proposed an integrated approach for assigning vehicles to patients, selecting hospitals and determining returning EMS station for the vehicles. We introduced the idea of serving two patients by an EMS vehicle which has not been considered before for the EMS dispatching problem. Three mathematical models were proposed which two of them calculate the exact travel cost of vehicles, one is nonlinear and the other is its linearized version. In the third model, the travel cost is approximately calculated for vehicles. Several observations are made from numerical experiments. First, numerical results show serving two patients by a vehicle results in a significantly better performance in terms of waiting time for patients, especially when the demand for vehicles is high. Second, it was shown that the approximated model can provide high quality solutions in a real time environment, thus, this model can be implemented when there is a surge and EMS managers have the authority to determine hospitals for patients.

In the second problem, we considered the coverage problem independently. The optimization-simulation approach used for this problem has not been considered in other researches related to the ambulance coverage problem. We have used a simulation approach to first tune the parameters of the optimization model, and

second, evaluate the performance of the optimization model. The simulation results show that implementing the optimization model can result in decreasing the number of patients not served in the required time window. Additionally, our simulation results showed that our proposed model had better performance than the model in Gendreau at al. (2001) for the considered virtual city. Also, a simulated annealing algorithm was developed for this problem and the results show that for medium sized cities in 97% of the cases, the algorithm can provide optimal solutions.

In the third problem, we considered the problem of dispatching independently. In the third problem, contrasting to the assumptions made in the first problem where dispatchers choose hospitals for patients, the choice of hospitals can be made by either a patient or by a separate patient-hospital assignment module (e.g., Sun et al. (2013)) available in an EMS management system. For the first time, we proved that the EMS dispatching problem, when the number of pickups for an ambulance is limited to two, is NP hard. The problem is modeled as a vehicle routing problem with pickup and delivery, and, a simulated annealing algorithm was proposed for solving it. Further, a column generation method was proposed for obtaining the lower bound. The results showed that the algorithm was able to provide solutions within 2.5 % of the lower bound.

We have also presented a real time decision support tool for dispatching EMS vehicle and the algorithm within that is presented. A customized heuristic algorithm was also developed for this decision support tool.

The current research can be continued in different directions. Below, we have listed the main ideas for the future research:

1) One extension of the research can be developing an exact algorithm for the models proposed in Chapter III using decomposition techniques.

2) Another extension of the current research can be related to developing a customized exact algorithm for EMSDP. In Chapter V, a column generation method was proposed. We can extend this method by implementing a branch and price

algorithm to find the optimal feasible solution.

3) Another important case of ambulance can be found in the areas of hospital evacuation. The proposed model can be extended to solve this problem.

4) Analyzing the proposed algorithm with the real data can be another area for the future research for the models proposed in Chapters III, IV and V. Testing the models with real data is desirable and it will show how optimization models behave under real life situations, it is also useful in fine tuning parameters of the models.

5) In Chapter IV, we have designed a simulation-optimization approach for relocating EMS vehicles. Similarly, we can design an optimization-simulation approach for the dispatching problem as well. By applying simulation and optimization, an approach similar to approximate dynamic programming can be used to determine the optimal policy for dispatching at each time of a day.

REFERENCES

[1] Armstrong, E., Corning, D., Gainor, D., Kalinowski, E., King, M., Manz, D., and Margolis, G., (2006). The national EMS scope of practice model, National Highway Traffic Safety Administration, Washington, DC.

[2] Andersson, T., and Varbrand P., (2007). Decision support tools for ambulance dispatch and relocation. Journal of the Operational Research Society, 58, 195-201.

[3] Aytug, H., and Saydam, C., (2002). Solving large-scale maximum expected covering location problems by genetic algorithms: A comparative study. European Journal of Operational Research, 141, 480-494.

[4] Badri, M., Mortagy, A., and Alsayed, C., (1998). A multi-objective model for locating fire stations, European Journal of Operational Research, 110(2), 243-260.

[5] Batta, R., Dolan, J, and Krishnamurty, N., (1989). The maximal expected covering location problem: Revisited. Transportation Science, 23 (4), 277-287.

[6] Bent, R., and Hentenryck, P. V. (2006). A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. Computers and Operations Research, 33(4), 875-893.

[7] Beaudry A., Laporte G., Melo, T., and Nickel, S., (2010), Dynamic transportation of patients in hospitals, OR Spectrum 32;77-107.

[8] Brotcorne, L., Laporte, G., and Semet, F., (2003). Ambulance location and relocation models. European Journal of Operational Research ,147(3) , 451-463.

[9] Charnes, A. and Storbeck, J., (1980). A goal programming model for the siting of multilevel EMS systems. Socio-Economic Planning Sciences, 14(4), 155-161.

[10] Chelst, K., and Jarvis, J., (1979). Estimating the probability distribution of travel times for urban emergency service systems. Operations Research, 27(1), 199-204.

[11] Church, R., (2002). Geographical information systems and location science. Computers and Operations Research, 29, 541-562.

[12] Church, R. and ReVelle, C., (1974). The maximal covering location problem, Papers of Regional Science Association, 32 ,101118.

[13] Church, R. and Weaver, J., (1985). A median location model with non-closest facility service. Transportation Science,, 19(1), 58-74.

[14] Cordeau, J. F., and Laporte, G. (2003). The dial-a-ride problem (DARP): Variants, modeling issues and algorithms. Quarterly Journal of the Belgian, French and Italian Operations Research Societies, 1(2), 89-101.

[15] Emergency Medical Services and Non-Emergent (Medical) Transport Organizations Pandemic Influenza Planning Checklist, (2006), Department of Health and Human Services, USA.

[16] IBM Corp. IBM ILOG CPLEX Optimization Studio V12.2, 2010.

[17] Cuninghame-Greene, R., and Harries, G., (1988). Nearest-neighbour rules for emergency services. Operations Research, 32(5), 299-306.

[18] Current, J., Tatick, S., and ReVelle, C., (1997). Dynamic facility location when the total number of facilities is uncertain: a decision analysis approach. European Journal of Operational Research, 110, 597-609.

[19] Current, J., and Schilling, D., (1987). Elimination of source of A and B errors in p-median location problems. Geographical Analysis, 19(2), 95-110.

[20] Daskin, M., (1983). A maximal expected covering location model: Formulation, properties, and heuristic solution. Transportation Science, 17, 48-69.

[21] Desrochers, M., Desrosiers, J., and Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. Operations research, 40(2), 342-354.

[22] Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. Networks, 44(3), 216-229.

[23] Fisher, M., Jaikumar, R., and Wassenhove, L., (1986). A multiplier adjustment method for the generalized assignment problems. Management Science, 32(9), 1095-1103.

[24] Fujiwara, M., Makjamroen, T., and Gupta, K., (1987). Ambulance deployment analysis: a case study of Bangkok. European Journal of Operational Research, 31(1), 9-18.

[25] Galvao, .D., Chiyoshi, F, and Morabito, R., (2005). Towards unified formulations and extensions of two classical probabilistic location models, Computers and Operations Research, 32 (1), 15-33.

[26] Gendreau, M., Guertin, F., Potvin, J. Y., and Seguin, R. (2006).Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. Transportation Research Part C: Emerging Technologies, 14(3), 157-174.

[27] Gendreau, M., Guertin F., Potvin, J., and Taillard, E., (1999). Parallel tabu

search for real-time vehicle routing and dispatching. Transportation Science, 33(4), 381-390.

[28] Gendreau, M., Laporte, G., and Semet, F., (1997). Solving an ambulance location model by Tabu search. Location Science, 5(2), 75-88.

[29] Gendreau, M., Laporte, G., and Semet, F., (2001). A dynamic model and parallel Tabu search heuristic for real-time ambulance relocation. Parallel Computing, 27(2), 1641-1653.

[30] Gendreau, M., Guertin, F., Potvin, J. Y., and Seguin, R. (2006). Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. Transportation Research Part C: Emerging Technologies, 14(3), 157 - 174.

[31] Goldberg, J., (2004). Operations research models for the deployment of emergency services vehicles. EMS Management Journal, 1(1), 20-39.

[32] Gong, B. and Lin, C., (2009). Dispatching optimization and routing guidance for emergency vehicles in disaster. Proceedings of the IEEE International Conference on Automation and Logistics Shenyang, China August 2009.

[33] Gong, Q. and Batta, R., (2004). Methodology to manage priority queues of casualties in a dynamic disaster environment. Proceedings of the 13th Industrial Engineering Research Conference, Institute of Industrial Engineers, Norcross, GA.

[34] Gong, Q., and Batta, R., (2007). Allocation and reallocation of ambulances to casualty clusters in a disaster relief operation. IIE Transactions, 39(1):27-39.

[35] Haghani, A. and Jung, S., (2005). A dynamic vehicle routing problem with time-dependent travel times. Computer & Operation Research, 32(11), 2959-2986.

[36] Haimovich M. and Rinnooy Kan A., (1985)"Bounds and Heuristics for Capacitated Routing Problems," *Mathematics of Operations Research*, 10(4), pp. 527-542

[37] Hakimi, S., (1964). Optimum locations of switching centers and the absolute centers and medians of a graph. Operations Research, 12(3), 450-459.

[38] Hogan, K., and ReVelle, C., (1986). Concepts and applications of backup coverage. Management Science, 32(11), 1434-1444.

[39] Hodgson, M., and Neuman, S., (1993). A GIS approach to eliminating source C aggregation error in p-median models. Location Science, 1, 55-70.

[40] Hillsman, E., and Rhoda, R., (1978). Errors in measuring distances from populations to service centers. Annals of Regional Science, 12(3),74-88.

[41] Hunsaker B, Savelsbergh MWP (2002) Efficient feasibility testing for dial-a-ride problems. Operations Research Letters 30: 169-173.

[42] Ibri, S., Nourelfath, M., and Drias, H., (2010). A parallel hybrid ant-tabu algorithm for integrated emergency vehicle dispatching and covering problem. International Journal of Innovative Computing and Applications, 2(4), 226-236.

[43] Ichoua, S., M. Gendreau, and Potvin, J., (2003). Vehicle dispatching with time-dependent travel times. European Journal of Operations Research, 144, 379-396.

[44] Islam, K., (1998) Spatial dynamic queueing models for the daily deployment of air tankers for forest fire control. Ph.D. dissertation, University of Toronto, Toronto, Canada.

[45] Jaw, J. , Odoni, A., Psaraftis, H., and Wilson, N., (1986). A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transportation Research Part B: Methodological, 20(3), 243-257.

[46] Jotshi, A., Gong, Q., and Batta, R.,(2009). Dispatching and routing of emergency vehicles in disaster mitigation using data fusion. Socio-Economic Planning Sciences, 43(1), 1-24.

[47] Jung, S., and Haghani, A., (2001). A genetic algorithm for the time dependent vehicle routing problem, Transportation Research Record. Journal of Transportation Research Board . 1771, 161-71.

[48] Kergosien, Y., Lente, Ch., Piton, D., Billaut, J.C., (2011), A tabu search heuristic for the dynamic transportation of patients between care units, European Journal of Operational Research 214 ,442-452.

[49] Krumke S., Saliba S., Vredeveld T. and Westphal S., (2008), "Approximation Algorithms for a Vehicle Routing Problem," *Mathematical Methods of Operations Research*, 68(2), pp. 333-359

[50] Larson, R, (1974). A hypercube queueing model for facility location and redistricting in urban emergency services. Computers and Operations Research, 1(1), 67-75.

[51] Larson, R, (1975). Approximating the performance of urban emergency service systems. Operations Research, 23(5), 845-868.

[52] Li, J., Mirchandani, P., and Borenstein, D., (2007). Real-time vehicle rerouting problems with time windows. European Journal of Operational Research, 194, 711-727

[53] Malandraki, C., Daskin, M., (1992). Time dependent vehicle routing problems: formulations, properties and heuristic algorithms. Transportation Science, 26 (3), 185-200.

[54] Marianov, V., and ReVelle, C., (1992). The capacitated standard response fire

protection siting problem: deterministic and probabilistic models. Annals of Operations Research, 40(1-4), 303-322.

[55] Marianov, V. and ReVelle, C., (1996). The queueing maximal availability location problem: A model for the siting of emergency vehicles. European Journal of Operational Research, 93(1), 110-120.

[56] Marianov, V. and Serra, D., (1996). The coherent covering location problem. Papers in Regional Science, The Journal of the RSAI, 75(1), 79-101.

[57] Marianov, V., and Serra, D., (2001). Hierarchical location-allocation models for congested systems. European Journal of Operational Research, 135(1), 195-208.

[58] Marianov, V., and Serra, D., (2002). Location-allocation of multiple-server service centers with constrained queues or waiting times. Annals of Operations Research, 111, 35-50.

[59] Martello, S., and Toth, P., (1987). Linear assignment problems. Annals of Discrete Mathematics, 132, 259-282.

[60] Maxwell, M., Restrepo, M., Henderson S., and Topaloglu, H., (2010). Approximate dynamic programming for ambulance redeployment. Informs Journal on Computing, 22(2), 266-281.

[61] Miller R., (1981). Computation of Steady-State Probabilities for M/M/1 Priority Queues. Operations Research. 29(5), 945-958.

[62] Owen, S., and Daskin, M., (1998). Strategic Facility Location: A Review. European Journal of Operational Research, 111, 423-447.

[63] MacDonald, M., (2010). Beginning ASP. NET 4 in C # 2010. Apress.

[64] MacDonald, M., & Freeman, A. (2010). Pro ASP. NET 4 in C # 2010 (4th ed.). Apress.

[65] Parragh, S., (2011), Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. Transportation Research Part C: Emerging Technologies 19.5, 912-930.

[66] Rajagopalan, K., Vergara, F., Saydam, C., and Xiao, J., (2007). Developing effective meta-heuristics for a probabilistic location model via experimental design. European Journal of Operational Research, 177, 83-101.

[67] Rankins, R., Bertucci, P., Gallelli, C., & Silverstein, A. T. (2010). Microsoft SQL server 2008 R2 unleashed. Sams Publishing.

[68] Recommended Actions for EMS Providers To Prepare For and Respond to Pandemic, (2009), Los Angeles County Emergency Medical Services Agency.

[69] Repede, J., and Bernardo, J., (1994). Developing and validating a decision support system for locating emergency medical vehicles in Louisville, Kentucky. European Journal of Operational Research, 75(3), 567-581.

[70] ReVelle, C., (1993). Facility siting and integer-friendly programming. European Journal of Operational Research, 65(2), 147-158.

[71] ReVelle, C., an Hogan, K., (1989). The maximum availability location problem. Transportation Science, 23(3) , 192 200.

[72] Ropke, S., and Pisinger, D.(2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science 40.4, 455-472.

[73] Ross, G., and Soland, M., (1975), a branch and bound algorithm for the generalized assignment problem, Mathematical Programming, 8(1), 91-103.

[74] Russo,T., (2006), Pandemic planning, Emergency Medical Service, 35 (10), 51-56.

[75] Sathe, A., and Miller, E., (2005). Optimizing location and relocation of response units in guarding critical facilities. Transportation Research Record: Journal of the Transportation Research Board, 1923, 127-136.

[76] Saydam, C, Repede J, and Burwell, T., (1994). Accurate Estimation of Expected Coverage: A Comparative Study. Socio-Economic Planning Sciences, 28(2), 113=120.

[77] Sayre, M., White, L., Brown, L., Armacost, M., Dean, J., Larmon, B., MacLean, S., Mann, N., Margolis, G., Melese, I., Neely, K., OKeefe, M., Spaite, D., McHenry, S., Davis, T., Walker, E., Krohmer, J., Panjada, D., and Kimzey, J., (2001). National EMS Research Agenda. National Highway Traffic Safety Administration, Washington, DC.

[78] Schilling, D., Elzinga, D., Cohon, J., Church, R., ReVelle, C., (1979). The TEAM/FLEET models for simultaneous facility and equipment sitting. Transportation Science, 13(2), 163-175.

[79] Sun, L., DePuy, G., and Evans, G., (2013). Multi-objective optimization models for patient allocation during a pandemic influenza outbreak. Computers & Operations Research

[80] Taillard, E. D., Badeau P., Gendreau M., Guertin, F., and Potvin, J., (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. Transportation science. 31(2), 170-186.

[81] Toregas, C., Swain, R., ReVelle, C., and Bergman, L., (1971). The location of emergency service facilities. Operations Research, 19(6) , 1363-1373.

[82] Toth, P., and Vigo, D., (2002). The Vehicle Routing Problem. Society for Industrial & Applied Mathematics (SIAM), Philadelphia.

[83] Weintraub, A., Aboud, J., Fernandez, C., Laporte, Gilbert, and Ramirez, E., (1999). An emergency vehicle dispatching system for an electric utility in Chile. Journal of the Operational Research Society, 50(7), 690-696.

[84] Xu, H., Chen, Z. L., Rajagopal, S., and Arunapuram, S. (2003). Solving a practical pickup and delivery problem. Transportation science, 37(3), 347-364.

[85] Yang, S., Hamedi, M., and Haghani, A., (2004). Integrated Approach for Emergency Service Location and Assignment Problem. Journal of the Transportation Research Board, 1882, 184-192.

CURRICULUM VITAE

NAME: Farshad Majzoubi

ADDRESS: Industrial Engineering Deprtment

                    University of Louisville, KY

DOB: Hamedan, Iran-Sep 21, 1983

EDUCATION &

TRAINING   B.S., Mechanical Engineering

                    Tehran Polytechnique, 2001-2006

                    M.S., Social-Economics Systems Engineering

                    Sharif University of Technology, 2006-2009

                    Ph.D., Industrial Engineering

                    University of Louisville, 2009-2014

HONOURS

AWARDS: Awarded featured Article of the journal of IIE Transactions on-

                    -Healthcare Systems Engineering, 2(3), Sept 2012,

Outstanding Student Logistics Distribution Institute (LoDI)-

-Louisville KY, Oct 2012

Informs President, University of Louisville Student Chapter, 2012

Manufacturing & Logistics Network Academic Award,-

-Greater Louisville Inc., October 2011

University of Louisville, Grosscurth Scholarship, August 2009.

PROFESSIONAL

SOCIETIES: INFORMS, IIE

PUBLICATIONS:

Majzoubi F., Bai L. and Heragu S. An optimization approach for

dispatching and relocating EMS vehicles,IIE Transactions

on Healthcare Systems Engineering, Oct 2012, V2-3

Majzoubi F., Bai, L. and Heragu S., Model for dispatching ambulances

during a pandemic, Proceeding of Industrial Engineering Research

Conference, Reno,NV, May 2011

Majzoubi F., Eshragh A., Reliability optimization of series-parallel

systems with a choice of redundancy strategies using a variable

neighborhood descent, Proceeding of IIEC February 2009, Tehran, Iran.