DYNAMIC MODEL FOR REAL-TIME AMBULANCE RELOCATIONS BASED ON COVERAGE VARIATION

by

Saba Sajjadian

BSc, Sharif University of Technology, Tehran, Iran, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Spring 2009

APPROVAL

Name: Saba Sajjadian

Degree: Master of Science

Title of thesis: Dynamic Model for Real-Time Ambulance Relocations based on Coverage Variation

Examining Committee: Dr. Arthur Kirkpatrick
Chair

Dr. Bill Havens, Professor Emeritus
School of Computing Science
Simon Fraser University
Senior Supervisor

Dr. Fred Popowich, Professor
School of Computing Science
Simon Fraser University
Supervisor

Dr. Binay Bhattacharya, Professor
School of Computing Science
Simon Fraser University
SFU Examiner

Date Approved: _____

Abstract

Dynamic real-time relocation of ambulances throughout the day is one of the most important issues that emergency medical services have to consider. Ambulance relocation decisions must be made online under time constraints to guarantee timely response to future accidents as well as optimum regional coverage. We develop a two-stage optimization model to satisfy the objectives included in the problem. To find the optimum solution for our model, we use simulated annealing search along with various problem-specific heuristics. Our case study is based on real data from an emergency services provider in the city of Ottawa, corresponding to making online relocation decisions for emergency responders. We simulate sample days using a historical data set of accidents and apply our optimization models and heuristics during the simulation. We evaluate the performance of our model and solution methods using the simulation results.

Keywords: Ambulance Relocation; Dynamic Optimization; Simulated Annealing.

To my parents, with love.

"Knowledge itself is power."

— Sir Francis Bacon

Acknowledgments

I would like to thank my senior supervisor Dr. Bill Havens for supporting me throughout my masters program. . Without his useful comments and discussions, I could not have finished my project. By giving me the chance to work on a real-world project, he introduced me to the true applications of my research area.

I would also like to thank my supervisor Dr. Fred Popowich and my examiner Dr. Binay Bhattacharya for agreeing to examine my thesis. I am also thankful to the people at Actenum Inc., especially Morten Irgens and Tom Carchrae for valuable discussions on the project and for providing me with useful feedback on my approach. I am also grateful to Precarn Inc. and CAE for letting me be part of their project and providing me with data sets. I would like to thank MITACS for their financial support during the internship program.

I am thankful to all my friends in Vancouver, who never let me feel alone. Their support and energy encouraged me to finish my studies. I would also like to thank my friend Arash for being so helpful and caring at all times.

Last but definitely not least, I want to thank my parents, Sedi and Mohammad, who were always there for me. Nothing could have replaced their love and support through these years. I dedicate this thesis to them.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

In this thesis we investigate the dynamic problem of ambulance relocations in emergency medical services. As these services are directly related to human lives, the attainment of high performance is one of the most important issues in these services. High performance is generally described as good coverage that results in timely response to accidents. Relocation decisions must be made online under severe time constraints in order to keep the performance level as high as possible. Other objectives such as travel time, workload, and human comfort issues are considered in the problem. Each of the previously studied dynamic models considers some of these objectives. They find an optimal set of relocations at each time of decision according to the specified objective. For each set of relocations, the objective function is evaluated assuming that the system will reach the new final configuration. However, this assumption is not always true. Accidents may happen during the relocations, when units are still on their way to new locations. As some of the relocations may not be completed, the system may not reach its predetermined final configuration. So, the problem with these models is that they ignore the value of the objective function and its fluctuations while the relocations are in progress. They only consider the value after all the relocations are completed. However, in this dynamic problem the value of the objective function is important at all times. It helps respondents to react efficiently to unexpected events. To overcome this problem, we focus on the variation of the objective function over time in order to find the best set of relocations. In Section 1.2 we describe the contribution of this thesis and in Section 1.3 outline the content of the remaining chapters.

1.2 Thesis Contribution

In this thesis, we introduce a new method to dynamically relocate ambulances during the day. The method has two parts. The first part predicts the most probable regions of the city for an accident to occur during the day. The second part directs the available ambulances to these regions. For the prediction, we use an *ad hoc* approach and show that it reaches a solution that is very close to the ones found by the previous methods of solving this problem. However, our approach is simple and fast while the previous methods, like mixed integer programming, use sophisticated algorithms that are not fast when the size of the input becomes large. In the second part of our approach - relocation - we consider not only the destination of the ambulances as important, but also the routes they take to get to final destinations. In the previous approaches, the relocation process has focused on the destinations of the ambulances and has sent the ambulances to their new destinations as soon as possible. Instead, we find a route that provides higher coverage during the transition of ambulances from one configuration to the new one. The historical data are aggregated in various ways to increase effectiveness.

1.3 Thesis Outline

In Chapter 2 we define the problem formally and introduce the key parameters we address in this thesis. In Chapter 3 we review the literature on related topics and survey the results and approaches that have been considered in the past to tackle this problem. Chapter 4 considers the optimization model we use to solve the problem; in this chapter we give more detail on our model with regard to variations in coverage. In Chapter 5 we describe the methods used to solve our 2-phase model and introduce some of the search methods for solving the problem. In Chapter 6 we present the specifications of the case study we use to evaluate our approaches, and in Chapter 7 analyze the performance of different methods and discuss the experimental results. In Chapter 8, we present conclusions from the study and suggest possible future work.

Chapter 2

Problem Definition

We can describe the problem as dynamically finding an assignment of available ambulances to the specified waiting sites throughout the city. The goal is to achieve coverage of the most important and highly demanded areas. In this way, ambulances can be ready to respond to upcoming accidents. Suggested repositioning should be done considering various soft constraints under time restrictions. The most important entities of the problem are described as follows.

2.1 Demand Point

Demand points are representative points scattered in the city. Their value, d_i, represents the aggregated demand in their area for medical services. These values are based on either the population size of the area or the historical data of accidents in the area. In previous studies, demand points have always been generated independently from time, meaning they have the same value throughout the day. However, according to the nature of the problem in hand, these points may have different values at different times of the day. Connecting roads between business areas and residential areas, as an example, have a higher record of accidents during the rush hours. However, at other times there is a lower chance of accidents on these roads. This example shows how the demand values may change according to different times of the day. So we introduce another class of demand points as *time-tagged* demand points.

Instead of one value for each point i, we have a set of values. Each value $d_{i,t}$ shows the demand values at time t for position i. We partition the life time, namely a day, into a number of time intervals. All the time intervals are the same for each demand point i. $d_{i,t}$

is generated based on the population of the area in that interval or the historical data of the accidents that occurred during that interval.

Based on our experimental results in Chapter 7, it turns out that in the cases where there is access to the raw information for generating demand points, using time-tagged demand points can be helpful in getting more efficient online decisions.

2.2 Post

Posts are the specified waiting sites throughout the city, where ambulances wait to be dispatched to accidents. These sites can be either actual facility locations or just a corner of two streets, for example. Determining their best locations is known as a facility location problem. In the relocation problem, however, we assume that their locations are determined beforehand and are fixed.

Each post p_i has three different states in the problem. It can be either *empty*, *full* as an ambulance is already residing there, or *to-be-full* as an ambulance is on the way to the post.

2.3 Unit

Units are actual emergency vehicles or ambulances. Each unit u_j starts a shift from its base according to the schedule and goes back to its base by the end of the shift. Units have two different general states. They are considered *available* if one of the following happens:

- They are waiting at the post.
- They are ready to relocate at the hospital after completing their previous service.
- They are ready to start their shift at their bases.
- They are on their way to an assigned post.

Or, they are *unavailable* as they are responding to an accident or when they are off shift.

2.4 Coverage

We say demand point i is covered if there is an available unit in distance r of i. r is the maximum distance that a unit can reach in a reasonable time defined by the emergency

medical services. Given the distance r as an input, the total coverage is the total demand values of the covered points. This definition has been used in previous studies on coverage models and can be formalized as follows:

$$TotalCoverage = \sum_{\text{point i is covered}} d_i \tag{2.1}$$

Also, the coverage ratio is defined in the following equation:

$$CoverageRatio = \frac{TotalCoverage}{\sum_i d_i} \times 100 \tag{2.2}$$

Remember that, in the models studied, demand values are time-independent. Hence, coverage value is independent from time as well and only depends on the location of available units. However, if we use time-tagged demand points, the coverage definition should be changed as demand values are dependent on time. At first glance, a simple modification is required to make the Equations 2.1 and 2.2 valid. To calculate the coverage at time interval t_c , current time interval, we use these two equations:

$$TotalCoverage2 = \sum_{\text{point i is covered}} d_{(i,t_c)} \tag{2.3}$$

$$CoverageRatio2 = \frac{TotalCoverage2}{\sum_i d_{(i,t_c)}} \times 100 \tag{2.4}$$

Although previous modifications in Equations 2.3 and 2.4 seem valid, the question is whether they are as useful as the two first definitions in Equations 2.1 and 2.2. The main idea behind the coverage definition in coverage models is to know approximately what percentage of upcoming accidents can be serviced in the acceptable time. This idea comes from the following facts:

1. Covering a demand point is equivalent to being able to respond to the accidents in the area.

2. The demand value in each area is expected to be proportional to the number of accidents in the area.

In definitions 2.3 and 2.4, we only consider the information for time interval t_c, which only shows us whether the current configuration is effective for responding to probable accidents during time interval t_c. So, the duration of the interval is an important parameter in our coverage evaluation. Assume our time interval is the 24-hour interval; in this case we will only have one interval, and the equations are the same as Equations 2.1 and 2.2. But, as we shorten the time interval, we reduce our look-ahead boundaries and start focusing more on current time interval. We may prefer a configuration for our next half-hour which is not as effective for our next hour in total. So, during the next hour we need to make another set of relocations half-way through to get better configuration for the second half. However, if we consider total information for the next hour in the beginning, we may not need to make redundant relocations after half an hour. This is just an example to show that we need to consider demand values not only for the current time interval but also for the intervals after that. But the point is that there must still be a difference between these intervals, as this is the basic idea of introducing time intervals in the first place. More concisely, we can say that although we are looking at future intervals, still the ones closer to current time must have greater weight. We give weight to time intervals based on their closeness to the current time. According to this definition, the closer interval t is to t_c the higher is its weight. The weights are then used to calculate the coverage in the next two definitions:

$$w(t) = \frac{t_{last} - t_c}{t - t_c + \frac{IntervalDuration}{2}} \qquad for \quad t > t_c \tag{2.5}$$

t_{last} is the last time interval of the day. Interval duration is the size of each time interval. In the calculation of coverage, each demand value is multiplied by the weight of its time interval so that the demand values of the near future have more influence in the coverage.

This modification considers the timely services to accidents in the near future in the first place and possible timely responses to further-away accidents afterwards. The following two equations use $w(t)$ in their calculations:

$$TotalCoverage3 = \sum_{\substack{\text{point i is covered} \\ \& \\ t \geq t_c}} d_{i,t} \times w(t) \tag{2.6}$$

$$CoverageRatio3 = \frac{TotalCoverage3}{\sum_{\substack{i \\ \& \\ t \geq t_c}} d_{i,t} \times w(t)} \times 100 \tag{2.7}$$

Other changes can also be made to the equation, depending on the problem specifications. For example, we can set an upper bound on the time intervals greater than t_c. So instead of considering all the intervals, for example, the next 24 hours, we choose an upper bound and only consider the next 5 hours as an example in our calculations. In this way we are not concerned about what might happen in the far future, since the state of the problem will change greatly until reaching that point of time.

2.5 Relocation

Relocation can be defined as a redistribution of paramedic resources or units to ensure adequate coverage throughout the area of responsibility. Relocation is triggered by any change in the number of available units. Whenever a unit changes its state from *available* to *unavailable*, the area of coverage is adversely affected unless some other unit is in the area to restore that coverage. On the other hand, whenever a unit becomes *available*, there is a possibility to increase area coverage and provide more efficient service if the unit is assigned to a suitable post. So, relocating the available units among the posts is needed to improve the coverage and satisfy maximum constraints.

At each step, we have two options:

- Single relocation
- Multiple relocations

In single relocation, only one unit can change place. A unit that has just become available is assigned to an empty post. Or, a post that has just become empty becomes full through the relocation of an available unit. The advantage of single relocation is in its processing time. Finding the best possible relocation according to the objectives is fast in this case. On the other hand, sometimes long travel time is required to make the relocation.

Multiple relocations can solve this problem, as they imply a sequence of relocations in order to cover a post. For example, we may need to move unit a from its position to the position of unit b and move unit b to a new position.

In general, in each step all the available units can relocate at most once among all the posts in multiple relocations. One of the most important issues in this problem is the definition of objectives. The relocation set found dynamically in each round is the best set found according to the objectives. These relocations affect the performance of the system in responding to future events. Most of the previous studies were focused on defining the suitable objective that could navigate the problem towards a better situation in the real world. In our optimization model, we define a new objective and evaluate its efficiency against the previous objectives.

2.6 Other Side-Constraints

There are multiple hard and soft constraints corresponding to the problem. Hard constraints should always be satisfied. A hard constraint is that there is at most one unit residing in each post at each time. Soft constraints may not be satisfied, but the goal is to satisfy them as much as possible. Soft constraints can be considered as follows:

- High coverage is required.
- Units do not want to work overtime.
- Long travel time for relocation is not preferred.
- Lower response time is required.

Having introduced the entities of the problem, we can now restate our problem definition more precisely. The problem is to dynamically find the set of optimal relocations whenever required. The optimal set is the set that provides the maximum satisfaction level of hard and soft constraints. With regard to the soft constraints, it is obvious that we have no control directly over future response times; however, maximizing the coverage can indirectly decrease the response time. This makes coverage one of the important factors in finding the best solution.

Chapter 3

Literature Review

The emergency medical services (EMS) location and relocation problem has been extensively studied over the past 40 years. The first models introduced in this area were simple and did not capture the aspects of the problem in its reality. As further research was done in this area, models became more sophisticated, capturing more characteristics and insight about the nature of real systems [10]. Starting from static models where no uncertainty was involved in the problem, gradually over time some uncertainty factors were added. Later, the idea of solving the problem offline based on these uncertainty factors changed to a proposal to solve the problem dynamically. Dynamic models respond to any changes in the system in a real-time manner.

There are a number of reviews of the research in this area. One of the most recent is [10]. In this review, the models are divided into three groups based on their ability to consider and deal with ambulance availability in the city. *Static* models, *probabilistic* models, and *dynamic* models are mentioned in [10]. In another review [18], two sources of uncertainty are used as partitioning criteria. Apart from uncertainty in ambulance availability, this review also considers models based on the uncertainty in travel time between two locations. Among all these models, it is the dynamic models that are most appropriate in solving our problem according to the dynamic nature of our problem. We have therefore carried out a detailed study of them in order to get a better understanding of the problem and to find a new approach to attack it.

Online stochastic combinatorial optimization is another research area that seems to be related to our problem. Some of the problems discussed in this area are introduced in [22], and algorithms combining stochastic programming and online algorithms are suggested to

solve them. The algorithms are tested on different applications, and the results show an improvement over previous online algorithms. However, there are no results on the outcome of these models being applied in the dynamic relocation of emergency services.

In the following sections of this chapter, we go through some of models in the ambulance location and relocation research area, from the simple to the complex, concluding with models used in dynamic relocation problems. We will focus on these dynamic models and investigate their advantages and shortcomings. We describe the objectives, constraints, and solution methods for each of these models. Following that, we take a look at the class of online stochastic optimization problems and the online anticipatory algorithms introduced to solve them. Despite our first assumption, which considers our problem as a member of this class, we show the reasons why it cannot be in this class.

3.1 Static and Probabilistic Models

Static models ignore ambulance availability over time and assume that there are a fixed number of ambulances available at all times. These models served as the basic models in EMS location problems. As the first model in this series, we cite the Maximal Covering Location Problem $MCLP$ [13]. Its objective is to maximize the number of demand points to be covered. Each demand point is either in the coverage area of a specific ambulance (coverage =1) or it is not in the coverage area (coverage=0). Demand points can only be covered once in this model. The Double Standard Model DSM [19] is another static model; it maximizes the demand points that are covered twice according to specific coverage radius. The model makes sure that all the demands are covered within radius r_2 and also α portion of demands are covered within radius r_1 where $r_1 < r_2$. DSM [19] uses tabu search [28] to improve the initial solution resulting from the linear relaxation of the problem.

In previous models, the travel time between two locations is always assumed to be the same. However, in reality the travel time between two specific locations changes during the day. As an example, an ambulance from location l_1 can reach the demand point in location l_2 in 8 minutes in quiet hours of the day, whereas it may take up to 12 minutes to get from l_1 to l_2 in rush hours. Assuming the critical time as 10 minutes, obviously the demand point is not covered by this ambulance in busy hours. The Maximum Covering Location Problem with Probabilistic Response time $MCLP + PR$ is introduced in [16] to model the uncertainty in the response time. Considering the probabilistic response times, the goal is

to maximize the total demand points covered.

Another extension of static models considers coverage probability instead of 0-1 coverage. Differentiating between points that are closer to the ambulance in the covering radius and points near to the boundary of the coverage area is discussed in [3]. Clearly, closer points have higher probability of being covered by an ambulance in their area. This idea is extended in Maximum Survival Location Problem $MSLP$ [17], where the coverage is replaced by the concept of survivability. The idea of survivability had been addressed before in the medical literature. Different formulas were introduced to approximate and measure the chance of survival for a patient. Many factors are considered such as priority of the call, response time, paramedic equipment, age of the patient, *et cetera*. This paper [17] considers one of the survivability formulas based on the response time. For each pair of demand point d and ambulance location l there is a survival value. The value shows the survival chance of the patient at d serviced with an ambulance from l. It is assumed that the patient is in a critical condition such as a heart attack. The model finds a set of locations that can maximize the total survival values for all the demands. It is assumed that each demand point can be served with at most one ambulance. The formulation of this new problem, maximum survival location, can be extended by adding probabilistic response time as well. The new linear model, Maximum Survival Location Problem with Probabilistic Response time $MSLP+PR$ is examined against the previous coverage based model, $MCLP+PR$. The solutions found by the coverage model are evaluated using the survival objective. The results show that the solutions found by coverage models are not the best solutions according to the survival objective. Survival measure is also more understandable for people in medicine or EMS and gives them a better idea for using the output of these models.

The fact that these models do not consider ambulance availability makes them not a good choice for the dynamic relocation problem, where we want to cope with the changes happening to the system. However, we can assume that whenever a change happens at time t to the system, a new problem is generated. For this new static problem at time t, we can find a best set of locations by the use of static models. Definitely, it is not enough just to use these models. We need to combine their solutions with some other algorithms to find the set of desired relocations. This is the main idea of the first part of our work: to make use of these static models and their solutions to find the set of final locations for ambulances. How ambulances should move to these final locations is the second part of our work. We introduce a new model and a new objective for this second part.

Adding probabilistic response times as well as replacing 0-1 coverage with probability brings the models closer to reality; however, all the mentioned models are based on the continuous availability of units. This assumption is not always true, as there are times when emergency vehicles are not available at the sites. *Probabilistic models* try to overcome this deficiency by considering the probability that a vehicle is busy. Considering the busy probability as p, the probability that a demand point is covered by one of the k ambulances in its coverage radius is equal to $1-p^k$. Models in this series maximize the expected number of covered demand points. The Maximum Expected Covering Location Problem $MEXCLP$ [15] is one of the first models of this series. All ambulances in all locations have the same value of p. The Maximum Availability Location Problem $MALP$ [27] is another model in this category that introduces a specified reliability level. The idea is to set a lower bound on the number of ambulances that should cover a point according to the busy probability of each site. The objective in this model is to maximize the demands which are covered under specified reliability level. Again in this model all the sites have the same busy fraction. In reality, the busy fraction is completely dependent on the location of the sites. Sites in more crowded areas with more demand points in their area of service have a higher busy fraction. $MALP2$ [27] as an extension to $MALP$ considers this fact.

A more complicated model is also introduced, based on the combination of the busy fraction and probabilistic response time in the Maximum Expected Coverage Location Problem with Probabilistic Response time $MEXCLP + PR$ [11].

3.2 Dynamic Models

One of the first dynamic models to consider the relocation problem was an extension of DSM introduced in the Dynamic Double Standard Model DDSM [20]. The solution to the model is the set of relocations that consists of the new locations for each ambulance. The model solves the relocation problem at time t and maximizes the double coverage demand minus the relocation cost. The relocation cost at time t is defined by the penalty coefficients, ${M_{jl}}^t$ associated with the relocation of ambulance l to new location j at time t. The penalty coefficient is used to penalize the movement of an ambulance that has already been moved a lot. The coefficients are updated at each time of relocation to capture the new state of ambulances and their locations. The definition of this penalty value seems very useful in making the location decisions in order to satisfy human comfort issues. However, the way

to measure this value is not mentioned in the paper. The model should be solved repeatedly during the day. In order to be able to respond quickly, one must pre-compute solutions for the next possible event. More precisely, for each available ambulance at time t, a set of relocation decisions is pre-computed in case of the possible assignment of this ambulance to the next accident. In this way, whenever an ambulance is dispatched to an accident, the corresponding pre-computed solution is available. In order to efficiently calculate these solutions for all possible next assignments, parallel processing is used. The solution for each possible next state is solved on a separate machine. Similar to their previous work in static models [19], these researchers used a tabu search heuristic [28] to solve their new model.

Use of parallel processing is one of the main limitations in using this model in many situations, as there might not always be access to parallel machines. Another model that tries to avoid the cost of parallel processing while still keeping the model feasible for time constraint situations is studied in [21]. The Maximal Expected Coverage Relocation Problem $MECRP$ introduces a dynamic relocation strategy. The problem is divided into two phases. The first phase finds the best set of final locations, as final configuration, while the second phase considers the real movements of available ambulances to reach this new configuration. In order to reduce the cost of computation at the time of decision, the first phase is pre-computed. Having the total of N ambulances, the first phase pre-computes the series of best locations which maximize the expected coverage for all possible number of available ambulances $n = \{1, 2, ..., N\}$. The problem is solved once at the beginning with the restriction on the number of locations that can be changed at each event. First phase is solved by the use of integer programming [29] and commercial integer programming solver CPLEX [2]. The second phase is the only part that should be solved repeatedly during the day. Knowing the set of current ambulance locations at time t and the set of future best locations obtained from the first phase suffices to solve a transportation problem to find the best set of relocations. As mentioned by the authors themselves [21], this model is only feasible for problems of small size, because of the pre-computation in the first phase. We have used the same idea of dividing the problem into two phases. However, there is no pre-computation involved. Pre-computed solutions are not always the best, since they do not consider the situation of the ambulances at the time of decision. We decided to change the first phase from pre-computation to real-time computation to consider those situations at the time of decision. More details are given in Chapter 4. Also we introduce another objective for the second phase, based on the coverage variation during the relocations, to

find the best set of movements. We believe that our new objective can result in a better set of relocations towards the new configuration. It helps to control the coverage value not only at the start and end of relocations but also in between, while the real movements are happening in the city.

A new quantitative measure is defined in [4] to evaluate a configuration based on the preparedness level for responding to accidents. Geographical area is partitioned into sets of zones, where each zone is assigned a weight value based on the population of the zone or the historical accident data. The preparedness for zone j is calculated as

$$p'_j = \frac{1}{c_j} \sum_{l=\{1...L_j\}} \frac{\gamma^l}{t^l{}_j} \tag{3.1}$$

where c_j is the weight of zone j, L_j is the number of closest ambulances considered in the preparedness of zone j and $t^l{}_j$ is the time it takes ambulance l to get to zone j. Contributing ambulances have different contribution factors, γ, dependent on the travel time. Closer ambulances have higher contribution factors. Following is the mentioned property:

$$t^1{}_j \leq t^2{}_j \leq ... \leq t^{L_j}{}_j \tag{3.2}$$

$$\gamma^1 \geq \gamma^2 \geq ... \geq \gamma^{L_j} \tag{3.3}$$

The minimum acceptable preparedness value, P_{min}, is evaluated based on the coverage constraints considered by health care services. In order to satisfy these constraints, preparedness values for all zones during the day should be higher than P_{min}. Whenever the preparedness level drops below P_{min} for one or more zones, a relocation is required. The relocation model considering the preparedness of different zones is described in [5]. The objective is to minimize the maximum travel time for the relocated ambulances. There are constraints on the number of relocations as well. But the most important constraint is to satisfy a preparedness level higher than P_{min} for all the zones after the relocations. Tree search heuristics are used to find this optimal set. Of course, in some cases there is no feasible solution to the problem.

Trying to keep the preparedness level above P_{min} is definitely an ideal goal. But the most challenging time for dispatchers in the calling centres is when the number of available ambulances drops, and there are not enough ambulances available in the city to provide P_{min} for all zones. In these cases, the model mentioned cannot come up with any feasible relocation set, and so the previous state is kept. However, in the real world, dispatchers

may still make relocations in order to increase the number of prepared zones. This case is considered in our work. Assuming a zone is prepared whenever it is covered by an available ambulance, we maximize the covered zones during and after the relocations. In most cases, however, we are not able to cover all the zones.

Another main issue in the previous model is the definition of preparedness. According to Equation 3.1, by increasing the weight of a zone the corresponding preparedness decreases. Let us consider there are two zones, z_1 and z_2 with $c_{z_1} = 2 \times c_{z_2}$, $L_j = 1$ and $\gamma^1 = 1$. In order to have the same preparedness level for both zones, the closest ambulance to zone z_1 should be two times closer than the closest ambulance to z_2. The question here is whether it is important to have a closer available ambulance in the area for z_1 or whether it is more important to have more ambulances available in the area of z_1. In our opinion to have the same preparedness, there must be more ambulances supporting zone z_1 than the number supporting z_2.

In [6], an assisting tool for dispatchers is introduced which will give them relocation recommendations along with information about the ideal coverage and the comparison of current state and the ideal solution. To find the next set of relocations, pre-computation is required to find the best set of locations for any number of available ambulances, which results in maximum expected coverage. Knowing the ideal set of locations, whenever a relocation is required, a transportation problem [26] is generated based on the current location of vehicles, the ideal locations and the travel time matrix. The objective is chosen to minimize the maximum travel time. The optimal solution to the transportation problem is revealed as relocation recommendations. Using an advance optimal solver to set ideal locations is one of the challenging parts in this work in [6]. As mentioned in this paper, the system is not efficient in solving problems with a large number of ambulances available. Distributed computing networks or even multi-core processors are suggested to decrease the processing time. However, these options are not always available and that is why we are using a local search algorithm instead. These algorithms have shown their efficiency in the case of combinatorial problems. Choosing an efficient local search method along with good heuristics can give us near optimum solutions in an acceptable time. In our solution, we have used simulated annealing [28] as one of the successful local search methods to find the best set of locations.

On the other hand, not all the available vehicles are resting at a location at the time of relocation. The dynamic nature of the problem will bring up situations where some

number of vehicles are already relocating between two sites at the time of relocation. Not considering these vehicles, the mentioned approach may force them to reroute. This could be frustrating for the drivers and paramedics. In order to avoid it, we can find the best set of locations at the time of relocation request. In this way, we consider the ambulances which are already on the route and try to avoid relocating them. Their destination can be ignored in our coverage optimization and it can be assumed that it is already manned with an ambulance and should be kept manned with the same ambulance after the relocation. The corresponding vehicles may not be considered in the set of available vehicles for relocation.

3.3 Online Stochastic Combinatorial Optimization OSCO

This is a class of dynamic problems where online decision making is required in order to change the existing plan or situation. The existing plan needs to be adjusted according to the new changes occurring in the system. And it is obvious there is an uncertainty factor in the changes that may occur in the system. According to this definition, many applications can be considered in the class; however, an important characteristic defined by the authors for this class is not satisfied in all the cases. It is emphasized that the uncertainty in these problems should not depend on the decision-making process. Also, for these uncertainty factors, a distribution or an approximation of it should be available for sampling. These characteristics build the main idea of the online anticipatory algorithms introduced. Some of the applications mentioned and tested in [22] are online packet scheduling, online stochastic reservation, and online vehicle routing. All these problems have the cited characteristics. As an example, consider the online vehicle routing problem, where requests are not known in advance. During the day, new requests come in, and the goal becomes to find out dynamically which request is the best to serve next. The distribution of the new requests appearing is obviously independent from the decision made by the system.

The most important challenge in dynamic problems is ignorance of the future in making decisions. Considering just the current state and past decisions is not a good way to tackle what might come up in future. Having a distribution of inputs gives us the opportunity to make decisions not just based on the current situation but also based on the sample of future inputs or events. Sampling will help us get wiser and more informative decisions. Consider a maximization problem, an online algorithm A , and an input distribution. The goal of OSCO is to maximize the expected profit of A on future input sequences [22].

A traditional approach that uses sampling in its decision-making process is mentioned in [12]. For making a decision at time t, the *expectation* algorithm generates a set of samples. Samples include probable events for the time horizon of $[t, t + \Delta t]$, where Δt is defined by the programmer. For each sample set, the problem is solved as an offline optimization problem, once per each possible decision at time t. Having a set of C choices to choose from at time t, each sample set is solved $|C|$ times considering different choices made at time t. The best choice overall is selected which has generated overall best solutions for sample sets. Obviously, the algorithm is not efficient under the time constraints, since it can only consider few samples. The solution to this problem is given in the *consensus* algorithm introduced in [8]. In this algorithm, each sample set is solved only once without fixing the choice at time t. For each solution, the choice made at t is given points. At the end, the choice with the highest point will win. In this algorithm, we are still able to examine an acceptable number of samples in the time-limited situations. Consensus outperforms expectation in these situations, but as the decision time increases, the difference between performances decreases to the point where expectation has the better performance. But there is still a limitation involved in consesus algorithm. *Elitism* is the problem arising in this algorithm since only the best choices are given credit for each sample set. There might be choices that are not the best, but which can generate results close to the best and perform more robustly on different samples [23]. The *regret* algorithm [7] overcomes this problem while keeping it possible to be executed under time constraints. The main idea is to let all the choices get credit from each sample set, as in the expectation algorithm. To have a reasonable execution time, the algorithm is not going to solve each sample set for $|C|$ times. Each sample set is solved once by the offline optimizer similar to consensus. The approximated solution is generated for each member of C based on this solution and problem specifications. These solutions are then used to give credit to all the choices for each sample set. Overall, the regret algorithm combines the good features from previous methods, and the result shows its better solutions in comparison to them.

Regret seems a good option for solving online stochastic optimization problems. The applicability of this algorithm is even broader as there is no need to have a distribution for input data to create the sample sets [9]. The distribution can be learned online using machine learning techniques. Even simpler, a sample set can be generated using simple sampling algorithms and historical data. The results on packet scheduling and vehicle routing demonstrate that the historical sampling produces the same solution quality as the

actual distribution.

The fact that the above algorithms are so promising on this class of problem encouraged us to consider them for the ambulance relocation problem. The first step in this process was to find out whether the problem had all the specifications mentioned for this class. And this is where the process failed. In our idea, the dynamic relocation problem was not a member of this class since the uncertainty factor in the problem was not independent from the decision-making process. At first glance, the uncertainty factor of the problem is that accidents happened throughout the day, independently of whatever decisions dispatchers were making on relocations. However, the point is that the uncertainty factor is not the accident. Whenever an accident happens, an ambulance is going to serve the accident and one post will become empty. This is when the relocation is required to cover the area of the dispatched ambulance. So, the problem can be defined as full posts becoming empty stochastically during the day, which may cause the coverage and the efficiency of the whole system to drop. Relocation decisions should be made to improve the situation. It seems that now we have a different uncertainty factor in the problem: "posts becoming empty." In the relocation problem, we are not concerned with the dispatching process. It is assumed that in the case of high-risk accidents, the nearest available ambulance is dispatched to the accident. So the new uncertainty factor, posts becoming empty, is dependent on the accidents as well as on the configuration of ambulances at the time of the accidents. And the configuration of the ambulances is the result of previous relocations made in the city. This shows the dependency of our uncertainty on the decision-making process, which prevents us from considering our problem as an online stochastic combinatorial optimization.

Chapter 4

Optimization Model

Our optimization model consists of two phases. For each phase a separate submodel is introduced. Whenever a relocation is needed, the problem is solved based on these models and according to the current state of each unit and post. We are generally looking at multiple relocations.

Our first-phase objective considers the final coverage that we want to achieve after relocations. How we may reach this new configuration is not included in our model at this level. After finding the solution to this part, we may focus on how units should relocate in order to arrive at this new configuration. Our second-phase objective is focused on relocation of units to the known final set of posts that need to be covered.

The idea of dividing the problem into two phases has been addressed before in [21]. But more important is the definition of objectives. Objectives are our guides to moving in the search space, and the final solution for the problem is chosen in respect to them. Again, we should confirm the importance of choosing an objective that can lead us towards a better configuration satisfying the maximum soft constraints in the problem. There have been studies on considering coverage as a first-phase objective [21, 6]. Our second-phase objective considers the coverage variations during relocations, which have not been studied before. We have applied different solution methods to these models as will be discussed later. In the following sections, we introduce both objectives and corresponding models in detail.

4.1 Coverage

Let us assume that when a relocation request is triggered, we have n available units to relocate among posts. We can omit available units that are already on their way relocating between two locations from this set. This way, we can stop rerouting them in the city. This may help in satisfying one of the human comfort issues, as asking the units to change their route is frustrating for the drivers. Also assume that the total number of *full* and *empty* posts is m. These m posts are the ones that are going to be involved in our future relocation decisions. Again, in our relocations, we can omit those posts which have a unit assigned to them on the way. We call the set of the posts considered in the relocation as P.

So, recalling our goal, we are looking for a set of relocations of our n units among m posts. According to the nature of the problem, we will have more posts than units in the city. This leaves at most $m - n$ posts empty after the relocations. Finding the posts that are more important to be staffed and identifying the ones of less importance are matters of immediate interest. More precisely, in this phase we are looking for a subset of n best posts. To find this set we go back to our problem definition. According to our definition, coverage ratio is one of the most important factors in the problem. High coverage can lead to better response time and fewer missed accidents. Coverage after relocation can be determined by observing the posts that are going to be full after that relocation. So, the most important posts to choose in our set are those that generate the best coverage ratio. Finding this set is known as the Maximum Coverage Location Problem $MCLP$ [13]. The mathematical formulation of the problem is as follows:

Let variable x_j be a binary variable equal to 1 if and only if post p_j is chosen to be full in the new configuration. The acceptable distance for coverage is r, as discussed in the problem definition.

$$\max \quad CoverageRatio \qquad Eq.[2.7, 2.4, 2.2] \tag{4.1}$$

$$\exists(x_j = 1 \;\; \&\& \;\; dis(i, p_j) <= r) \Rightarrow \quad \text{demand point i is covered} \tag{4.2}$$

$$\sum_{p_j \in P} x_j \leq n \tag{4.3}$$

$$\forall p_j \in P; \quad x_j \in \{0, 1\} \tag{4.4}$$

$$\forall p_j \notin P; \quad x_j \in \{1\} \tag{4.5}$$

In this model, constraint 4.2 defines when a demand point is covered. The number of full posts after the relocation is limited to n in constraint 4.3. In constraint 4.5, posts not in P are assumed as full posts in the next configuration. The $CoverageRatio$ can be calculated considering time-tag demand points or the time-independent ones. It could be determined according to the problem specifications and the data in hand.

This model suffices for our problem and we use it in our experimental chapter. However, for problems that may have more units than posts, $n > m$, we redefine our optimization model. In this new definition, x_j is a variable equal to the number of units residing in p_j in the new configuration. The objective function is also changed to a multi-level objective

$$< 1^{st}CoverageRatio, 2^{nd}CoverageRatio, ..., n^{th}CoverageRatio >$$

where $s^{th}CoverageRatio$ is the ratio of demand points that are covered s times (with s different units). Also, objectives are ordered in terms of importance with the $1^{st}CoverageRatio$ having the highest importance. The goal will be to maximize these objectives based on their importance, maximizing the s^{th} objective before the $(s+1)^{th}$ objective.

4.2 Relocation

After finding the final set of posts that are to be covered after the relocations, we need to find the best repositioning of available units to cover these posts. As mentioned in Section 2.6, at most one unit can be assigned to a post. So, a complete set of assignments can be considered as a one-to-one mapping, where every available unit is assigned to exactly one post from our chosen set. The size of the space of all possible one-to-one mappings is $n!$, recalling n as the total number of available units for relocation. To search in this space we should define an objective based on the problem specifications.

Referring to Section 2.6 on problem definition, the satisfaction levels of all the mentioned constraints are affected by different repositionings. Maximizing the coverage, as one of the basic important constraints, is also affected by different repositionings. Ideally, better coverage should result in better response time. This shows the importance of this objective. Although we considered this objective in the previous phase, further consideration is now required. In the previous phase, we found the final coverage that was our goal after completion of the relocations, but what is important is that the coverage ratio will keep changing during the relocations. And if we are lucky enough and no change happens to the system, it

reaches its predetermined final value by the end of relocations. To get a better idea of these changes in the coverage value, we look at the different scenarios happened in our problem. Let's consider the following two cases:

1. Relocations when unit becomes available.

   Assume the coverage ratio is 0. At time 0, six units start their shifts. After the relocations, each of the units is residing at a post. Figure 4.1 shows the changes in the value of coverage ratio during the relocations. The graph starts at time 0 and ends when all relocations are completed. The coverage ratio is evaluated whenever a unit reaches its destination, creating six data points, which we connect to create the graph. We can see that the coverage ratio changes as units are moving. At the time of unit arrivals, we can observe an increase in the value of the coverage. It reaches its final value 74.73% after 1422 seconds, which is the total time, required for the relocations.

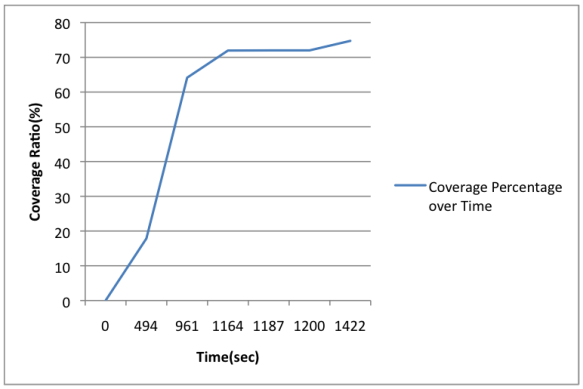

Figure 4.1: Coverage Ratio Variation as Unit Becomes Available

2. Relocations when unit becomes unavailable.

   Assume we have five available units providing 73.88% coverage at time 0. A unit becomes unavailable after 200 seconds, as it is dispatched to respond to an accident. This causes the coverage to drop to 25.83%. Three units from those remaining start relocating in order to increase the coverage as soon as possible. Figure 4.2 demonstrates the changes in the coverage as each of these units reaches its destination. All the units start relocation at the same time, moving towards an acceptable final configuration. After 870 seconds and by the arrival time of the first unit, the coverage increases to 70.88%. By the arrival of other units, the coverage reaches the maximum,

71.37%. The changes are quite small after the first arrival, since the other two units are already close to their destination posts at that time.

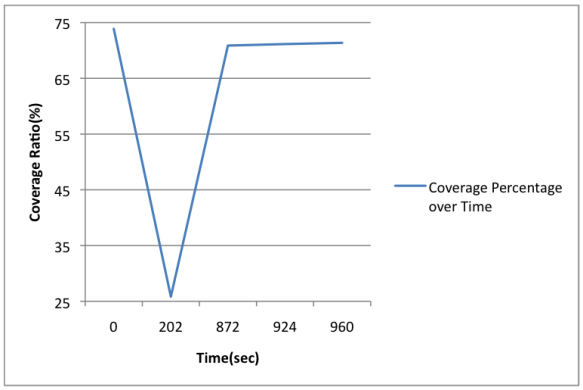

Figure 4.2: Coverage Ratio Variation as Unit Becomes Unavailable

These figures show that we can plot the variations of coverage for each set of relocations and investigate the coverage changes using similar graphs. For different possible relocation sets at time t, all the graphs start at the same coverage value and end up at the same final value. But their fluctuations over time and even their durations are different. In Figure 4.3, we can see two different relocation sets both end up in total coverage of 74.73 %.

Relocation is triggered from 0% coverage when six units start their shift at headquarters. Units follow different movements in each set. The coverage ratio is evaluated every minute. As we see, the coverage in set 1 increases faster. It reaches 56.93% after 840 seconds, where we have only 32.17% coverage in set 2. The nearly 24% difference in coverage is not ignorable, where every accident can cause the risk of death. Also, we see our final coverage after about 24 minutes in set 1, while in set 2 it takes us an extra 22 minutes to reach the 74.73% coverage. This extra time not only increases the risk of missing an accident, but also enforces longer driving time on drivers, which as a result can cause late meal breaks, overtime work, or lower preparedness for paramedics. Considering both graphs, we can see fewer fluctuations in set 1, as it is smoothly increasing, while there are some coverage drops in set 2, *e.g.*, at 1260 seconds. Again, these changes may put us in a situation of not being able to respond to unexpected events during relocations.

According to these insights, relocation set 1 is preferable in this case, providing us with higher coverage during the relocations and also smaller completion time. Dispatchers in headquarters try to take into account these factors as well, while relocating units. However,

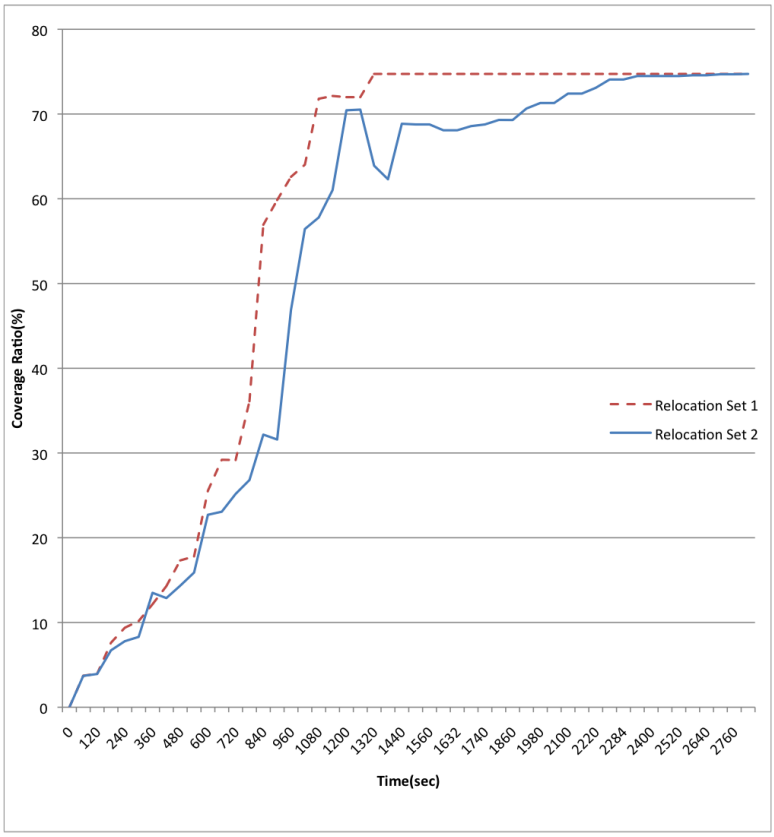

Figure 4.3: Coverage Ratio for Two Different Relocation Sets

under the high pressure, they may not be able to consider them perfectly. In order to help them make better acceptable decisions, therefore, we look at these factors as objectives for searching the space of different relocation sets.

In order to define our objective more precisely, we demonstrate the mentioned factors as measurable values in mathematical formulas. We have seen that one relocation set is superior to the other, based on its coverage variation over time, which results in reducing the probability of missing an accident during relocations. We can assume that the probability of missing an accident at time t is proportional to $100 - CoverageRatio(t)$, which is the uncovered ratio of demands at time t. Accordingly, in our model we minimize the probability of missing an accident during relocations. If a relocation request is triggered at time t_{reloc} and there is a maximum time, t_{max}, required to get to the new best configuration, we can formulate our objective as follows:

$$MissedP = \int_{t=0}^{t=t_{max}} UncoveredRatio(t_{reloc} + t)\, dt \tag{4.6}$$

If we plot the $UncoveredRatio(t)$ for $t_{reloc} \leq t \leq (t_{reloc} + t_{max})$, Equation 4.6 is equal to the area underneath the graph. Then, at each relocation time, we look for the set of repositionings that has the minimum area underneath.

In Figure 4.4, we have replaced the coverage ratio by its complement, uncovered ratio, for the same set of relocations considered in 4.3. The *MissedP* for each set of relocations is equal to the size of the shaded area for that set.

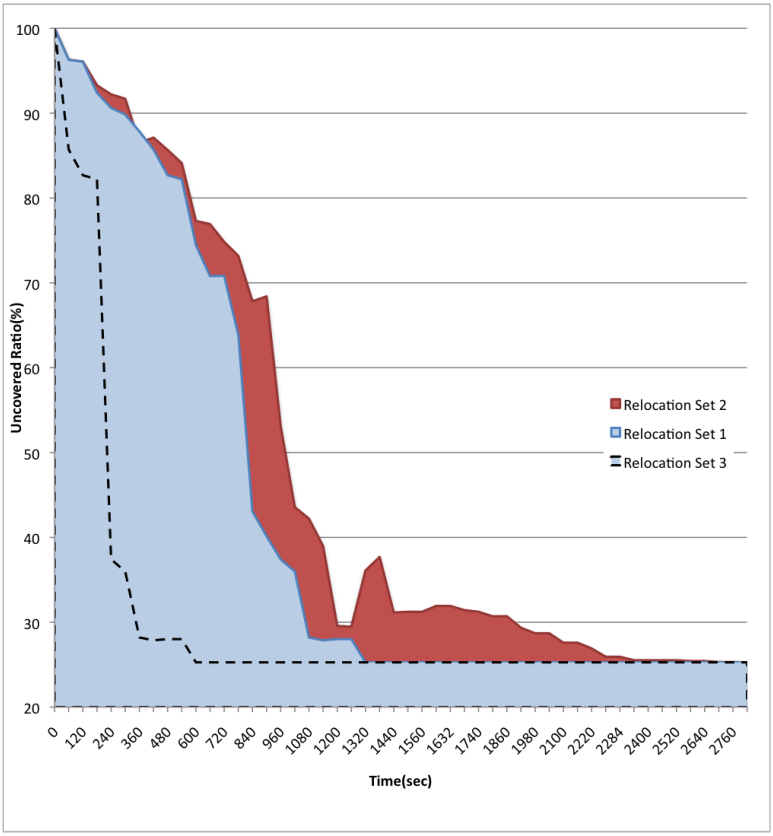

Figure 4.4: Uncovered Area

As we see, relocation set 1, which is preferable, has a smaller *MissedP* than set 2. Also, the imaginary relocation set 3 indicated by the dashed line has the lowest *MissedP*. It is obviously the best set among these three sets, as it goes from 100% uncovered ratio to 25% in 10 minutes in a rapid increase of coverage.

Considering our new entity, we formalize our objective as finding a one-to-one mapping of units to posts that minimizes *MissedP*. Let P^* be the set of posts that are selected to be covered. We also call the set of units available for relocation U. Variable v_i takes its value from domain P^*. It shows the assigned post to unit u_i. We have the following formal definition:

$$\min \quad MissedP \tag{4.7}$$

$$\forall u_i, u_j \in U; \quad v_i \neq v_j \tag{4.8}$$

$$\forall u_j \in U; \quad v_j \in P^* \tag{4.9}$$

The best set of assignments found is considered as the best found relocation set, which should be announced to units by dispatchers as soon as possible.

In the following chapters, we consider the algorithms and heuristics used with our optimization model to find the final solution.

Chapter 5

Solving Methods

In this chapter, we describe the methods we applied to the models in order to solve the problem. The solution method for each model is discussed separately. To find the solutions to both models, we use local search to find the suboptimal solution in the search space. Among different search heuristics and techniques, we focus on *Simulated Annealing* [28].

The *Simulated annealing* meta-heuristic uses a transition probability inspired from the *cooling* or *annealing* process in forming crystals. The following transition probability based on the Metropolis criterion is generally used in maximization problems:

$$p(s_1 \rightarrow s_2) = \begin{cases} \exp\left(\frac{\triangle H}{T}\right), & \triangle H < 0; \\ 1, & \text{otherwise.} \end{cases} \tag{5.1}$$

$\triangle H$ shows the difference in the objective value between new point s_2 and previous point s_1. If the difference is positive or zero, we move to the new point as its objective value is not worse than the previous point. But if it is negative, we will make the move according to the probability mentioned. T demonstrates the temperature of the system. A cooling schedule is used to decrease temperature T during the time. Studies on the convergence of simulated annealing toward global optimum have shown that by controlling the cooling schedule, SA will find the global optimum; however, it may require infinite time. The annealing schedule of the form of Equation 5.2 has been proven to lead the search towards global optimum [25].

$$T_t = \frac{\gamma}{\log(t+\beta)} \qquad 2 \leq \beta < \infty \tag{5.2}$$

where γ is a positive constant which depends on the problem specifications, and T_t represents the temperature after t time slots have passed from the beginning of the search,

where each time slot can be assumed as an iteration in the search. According to this formula, the decrease of T over t is very slow. However, we need to produce a solution in finite time. This forces us to have a temperature function that declines much faster. It is important to remember that these functions may not guarantee the convergence. Two of the most commonly used cooling schedules are [28]:

- Linear cooling:
  $T_t = T_0 - \alpha \times t$, where T_0 is the initial temperature which depends on the problem and α is the decrement usually chosen in the interval $[0.01, 0.2]$.

- Exponential cooling:
  $T_t = T_0 \times \alpha^t$, where T_0 is again the initial value and α is the cooling factor, usually in the interval $[0.8, 0.999]$.

After this brief introduction to simulated annealing, we explain in following sections the structure of the search space, the initial state, the neighbourhood definition, and the various heuristics used in applying this search method to our models.

5.1 Finding the Maximum Coverage Ratio

In order to find the maximum coverage ratio, Equation 4.1, we define a greedy approach to find the initial solution. According to constraints 4.4 and 4.5, we want to choose n posts from set P to obtain the maximum coverage. Recall that set P is the set of m posts considered in our relocation. We consider two new sets NP and CP. NP is the set of full posts in our next configuration, and CP is the set of the points covered by the posts in NP. At the beginning, NP is equal to $P\prime$, and CP contains corresponding covered points. $P\prime$ contains posts not included in P. For each post location l in the city and real number $0 < r$, let $D(l, r)$ be the set of demand points that are in distance at most r of l. For each post p_i let

$$w_i = \sum_{\substack{j \in D(p_i, r) \\ and \\ j \notin CP}} d_j \tag{5.3}$$

At each step $1 \leq s \leq n$ we choose a post p_i with maximum w_i, add p_i to NP, and add all the points that are in distance r of p_i to CP. We then need to update w_j for each post

p_j that has not been placed in NP.
The summary of the algorithm is :

Algorithm 5.1.1 *Greedy Initializer*

1. *Set NP equal to $P\prime$.*
   $CP = \{i | \exists p_j \in NP, i \in D(p_j, r)\}$

2. *For $1 \leq s \leq n$ do:*

   (a) *For $p_j \notin NP$, set :*

   $$w_j = \sum_{\substack{k \in D(p_j, r) \\ and \\ k \notin CP}} d_k$$

   (b) *Choose $p_i \notin NP$ where w_i is maximum.*

   (c) *Add p_i to NP and $CP = CP \cup D(p_i, r)$.*

According to the solution found, we assign binary variable x_j in our coverage model equal to 1 if and only if p_j is in set NP. This feasible solution is our initial place in the search space. Note that this greedy approach does not provide the optimal solution.

As an example, consider the case in which six posts are located on the vertices of a hexagon of radius r. There is also a post located in the centre of the hexagon. If the distribution of demand points is consistent in the area, all posts will have the same w. In the case of $n = 3$, the greedy algorithm chooses a random post first. Choosing the post in the centre as the first post is not a good start for our algorithm. However, if we skip the centre at the beginning of the algorithm 5.1.1 and choose another post instead, we can get a better solution. By choosing the three posts that are the non-neighbour vertices of the hexagon, we can completely cover the area in the middle as well.

Our search space consists of all possible solutions each appearing as a $k - ary$ point $(x_1, x_2, ..., x_k)$, where k is the total number of posts in the problem. The neighbourhood is $2 - swap$ neighbourhood, where we can get from one point to its neighbour by swapping the values of 2 attributes x_i and x_j for some i and j. Not all the points in the search space are feasible. Starting from a feasible solution, we need to define some constraints on our moves, so that the next solution remains feasible. For any attribute x_i chosen at any step, p_i must be a member of P. This constraint and simple 2-swap movements guarantee the

feasibility of the next solution. In each move, we will assign 1 to previously 0-valued x_i, while changing the value of another attribute x_j, as an example, from 1 to 0. It can be explained as removing p_j from our set NP and adding p_i instead. The coverage ratio for the new solution is evaluated, and the acceptance of this move is determined based on the transition probability value explained in Equation 5.1. According to the time constraint, the exponential cooling schedule is used throughout the search. After each move, the best found solution is updated. The search terminates after a fixed amount of time, and the best found solution is returned as the final answer. The termination time is determined based on the time constraints in the problem.

At each iteration of the search, we have suggested different strategies and heuristics to make a feasible move to one of the 2-swap neighbours.

- Random-Random: Choosing two attributes at random and swapping their values.
- Random Del-Max Add: Choosing an attribute, x_i, already assigned to 1 by random, change it to 0. Then choose another attribute, x_j, already assigned to 0 with maximum weight. Change its value to 1. In this way, we remove post p_i from set NP by random, update weights for posts not in NP, and choose the one with maximum weight and add it to NP.
- Random Add-Min Del: Choosing an attribute already assigned to 0 by random, change it to 1. Then choose another attribute already assigned to 1, which if it changes to 0 has the minimum effect. Change its value to 0. In this way, we add a post to set NP by random, for each of the posts in NP update their weights assuming that they are not in NP any more and choose the one with minimum weight and remove if from NP.

In the experimental results in Chapter 7, we evaluate the efficiency of this method in finding the maximum coverage based on different parameters.

5.2 Finding the Best Relocation Set

According to our definition in Section 4.2 on relocation, a relocation set is a one-to-one mapping from the set U of n available units for relocation, to set P^* of n posts found in the coverage section. The best relocation set is then determined as the one minimizing

uncovered area during the relocations, Equation 4.7. So, we introduce our search space as a space of $n-ary$ points, each representing a possible permutation of n posts in P^*. This space of size $n!$ contains all the feasible solutions. To find the initial solution, we solve an assignment problem with a cost function $C : U \times P^* \longrightarrow R$. The cost function can be represented by a $n \times n$ matrix as follows:

$\forall u_i \in U, p_j \in P^*$;

$$C(u_i, p_j) = \text{Estimated travel time for } u_i \text{ to get to } p_j \tag{5.4}$$

To find the best assignment with minimum total cost, we use *Munkres Assignment Algorithm* [26]. The algorithm can find the optimal solution in polynomial time $O(n^3)$. This optimal solution includes movements that increase the total driving time as little as possible. This seems to be a good place to start within the search space. Another informative solution that can serve as an initial solution is the set of relocations which minimize the maximum driving time for units involved in the relocations. In previous optimal solutions, although we have minimized the total driving time, we may still have a unit driving for a long time while others drive for only a few minutes. This increases the time required to completely reach our new configuration and new coverage. It may also violate the human comfort issues by enforcing a large workload on just one unit. By controlling the maximum driving time, we can control workload and completion time.

In our problem, we are able to find an optimal assignment whose maximum cost is minimized by the use of the Munkres algorithm. Let us assume a complete assignment by $A \subset U \times P^*$, where for every $u \in U$ there is exactly one pair (u, p) in A and for every $p \in P^*$ there is exactly one pair (u, p) in A. The total and maximum cost of A are

$$totalC(A) = \textstyle\sum_{(x,y)\in A} C(x, y) \tag{5.5}$$

$$maxC(A) = \max_{(x,y)\in A} C(x, y) \tag{5.6}$$

To find a complete assignment A^* which has the minimum $maxC$, the following algorithm is used. Let function $Munkres(C)$ return an optimal solution A which has the minimum total cost. Assume we have a set of marked pairs initially empty. At each iteration of the algorithm, we mark an unmarked pair (x, y) with maximum cost and try to find a complete assignment not including the marked pairs. The value of the recently marked pair represents an upper bound on the maximum cost of the assignment. By marking more pairs

we are reducing the upper bound until no more assignments can be found. The last found assignment is assumed to be our optimum solution.

Algorithm 5.2.1 *Finding A^* with minimum maxC*

1. $A \longleftarrow Munkres(C)$.
2. *Set* $A^* = A$.
3. $m = ((\max_{(x,y)} C(x,y)) \times n) + 1$.
4. *Choose* (x,y) *from* C *which has the maximum value smaller than* m.
5. *If no(x,y) found, return* A^*.
6. *Else,*
   (a) $C(x,y) = m$.
   (b) $A = Munkres(C)$.
   (c) *If* $totalC(A) < m$
      i. *Set* $A^* = A$, *go to step 4.*
   (d) *Else return* A^*.

In algorithm 5.2.1, we mark the maximum pair in step $6-(a)$ by assigning its value to a large number m which is greater than the total cost of any possible complete assignment. If there exists a complete assignment $A\prime$ without including marked pairs, the total cost of this assignment is smaller than m. The assignment with the smallest possible total cost is generated in step $6-(b)$ by the Munkres algorithm. The total cost of this assignment is compared to m in step $6-(c)$. If the value is smaller than m, we continue marking the pairs; otherwise, we will return the best solution found so far in step $6-(d)$. Having $O(n^2)$ pairs to be marked, the running time of the algorithm is $O(n^5)$.

However, instead of marking pairs one by one, we can mark a number of pairs in each round. Let us assume in each round we have an upper bound ub and we mark all the pairs with costs higher than ub. We want to find the minimum ub that enables us to have complete assignment with unmarked pairs. To search for this optimum upper bound, we use a binary search on the sorted array of cost values $SortedC = \{c_1, c_2, ..., c_{n^2}\}$, where

$\forall i \leq j; \quad c_i \leq c_j$. In each round, we assign ub equal to the median of search domain. If a complete assignment exists, we continue to search for optimum ub in the lower half of the domain; otherwise, we search for it in the upper half of the domain. During the search, whenever a complete assignment is found, the assignment and the corresponding ub are kept as best found solutions so far. Starting the search from a search domain of size n^2, $SortedC[1..n^2]$, search terminates when the domain size becomes equal to 1. The best found assignment and ub is considered as final solution. According to the running time of binary search $O(\lg n)$, we run Munkres $O(\lg n)$ times, which gives us the total running time of $O(n^3 \lg n)$.

A faster algorithm can also be presented if we model our problem as a bipartite graph $G(U \bigcup P^*, U \times P^*)$, where any complete assignment is a perfect matching and *vice versa*. Omitting the edges representing the marked pairs in each round from the graph, we can replace the Munkres algorithm with the maximum matching algorithm [14]. If the size of a maximum matching is n, then we have a perfect matching and a complete assignment. Recalling the running time of the maximum matching algorithm in bipartite graphs as $O(|E|\sqrt{|V|})$ [24], we can reduce the total running time to $O(n^{\frac{5}{2}} \lg n)$.

After finding the optimum assignment $A^* = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, we start our search from the corresponding initial point $(y_1, y_2, ..., y_n)$ in space. We define our neighbourhood as $2 - swap$ neighbourhood. For a neighbour point, the new assignment is generated and variable v_i is assigned a value:

$$\forall (u_i, p_j) \in A; \quad v_i = p_j \tag{5.7}$$

According to these values, the objective *MissedP* mentioned in Equation 4.7 is evaluated. Based on the transition probability used in our simulated annealing search, we may either move to this new point or stay at our previous point. A fixed amount of time is considered as termination criteria, and the best found solution during the search is returned as final solution. In the experimental results in Chapter 7 we discuss the results of applying this search method to our problem.

Chapter 6

Case Study: RISER

Rapid Intelligent Scheduling for Emergency Responders (RISER), is a Precarn Inc. optimization project on the dynamic relocation of emergency resources. The specification of this project indicated it as a good choice for a case study for the models and algorithms we have described here for solving the dynamic relocation of ambulances. The project basically concerns the emergency medical services in the city of Ottawa. The data sets gathered from the Ottawa Paramedic Service (OPS) are used as the input data to the project. We ran our experiments on the sample problems generated on the basis of these data sets. In this chapter, we describe various data sets involved in the RISER project.

6.1 Post

There are 27 posts scattered in the city of Ottawa. Each post is either an actual facility building or just a corner of two streets which is called a *mobile* post. Based on the call volume in the city, posts are either located in low or high density areas. Fig. 6.1 shows the locations of posts in the city using Google Earth. Each location is defined in terms of the Universal Transverse Mercator (UTM) coordinate system. The UTM coordinate system is a grid-based method of specifying locations on the surface of the Earth. The UTM zone for all the locations in the case study is 18. We convert the locations from UTM to latitude and longitude to demonstrate them using Google Earth.

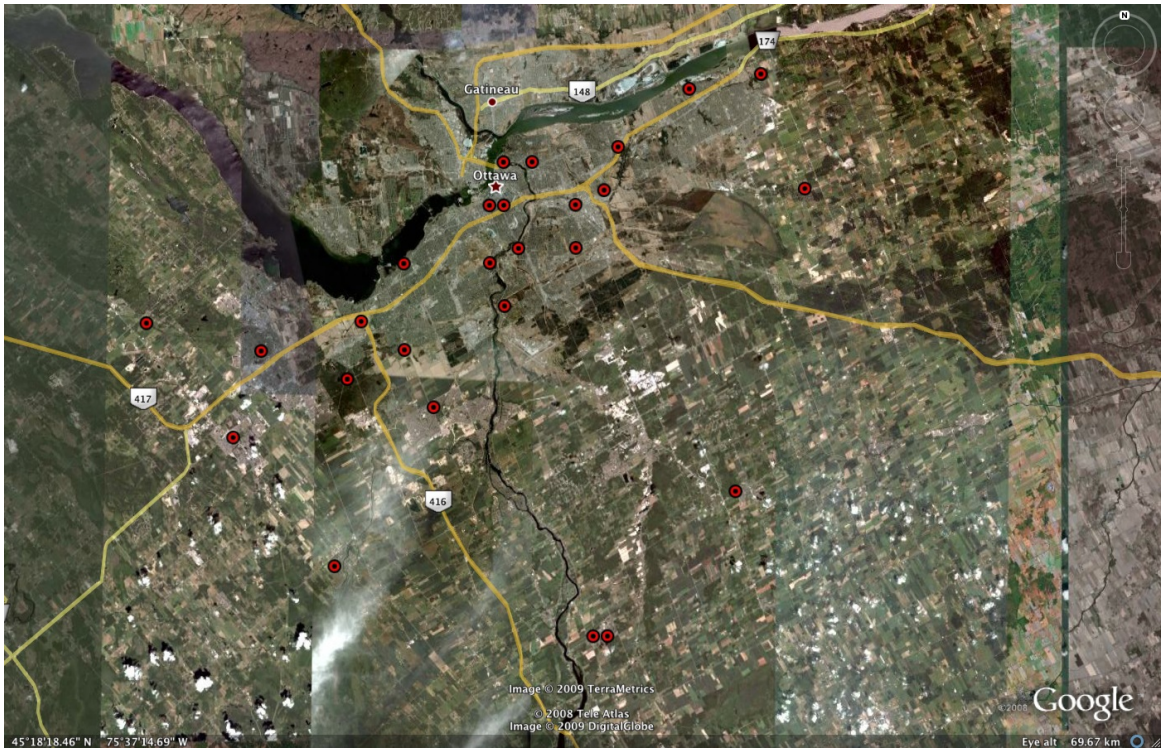

Figure 6.1: 27 Posts in Ottawa (Image courtesy of Google Earth)

6.2 Unit

Units work for 13-hour or 8-hour shifts. Approximately every four hours, paramedics are allowed to have a half-hour break. So, based on their shift duration, they may have either one or two breaks during their shifts. Starting a shift from its base, each unit must return back to its base by the end of that shift. Bases are either OPS headquarters or posts in the low density areas. Units starting their shifts in low density areas are preferred to be kept in these areas. However, they may travel to high density areas if required. Figure 6.2 shows the location of posts as red circles and the bases as blue circles.

6.3 Historical Accident Data

The historical data of accidents that occurred during the five months starting 1 January 2006 and ending 31 May 2006 are used as input data in the project. Based on this information, each accident was assigned a priority number by the experts in the call centre. High-risk

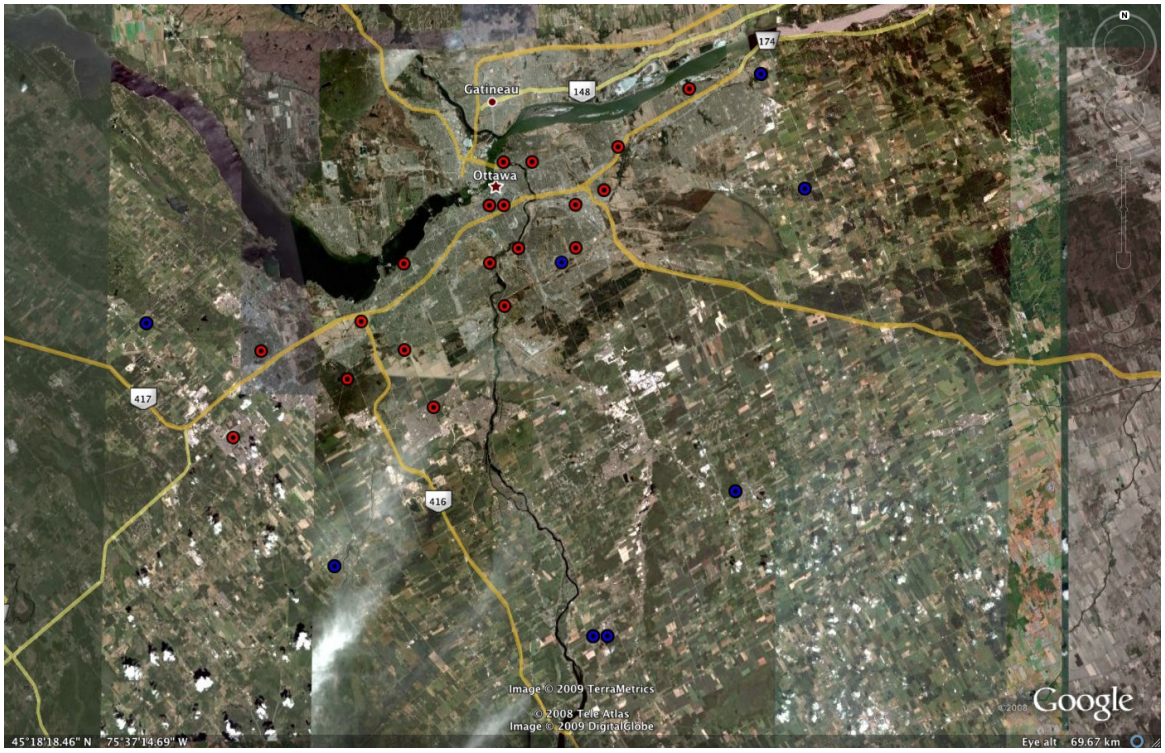

Figure 6.2: 27 Posts and OPS-HQ in Ottawa (Image courtesy of Google Earth)

accidents were assigned 4, while the less risky ones were assigned 1. To get useful information from this data set, we consider the distribution of high-risk accidents with priority 4 or 3 in the area. This distribution can be helpful in relocating ambulances more efficiently in the city to be more prepared to respond to high-risk accidents. The data set includes 21012 high priority accident data over 147 days. Fig.6.3 shows the distribution of these accidents in the city. Accidents are shown as black points.

Each day, accidents are monitored during the 24-hour period starting from 12.00 am. The average number of accidents in each day of the week is shown in Table 6.1. It shows

Mon	Tue	Wed	Thu	Fri	Sat	Sun
145	146	139	145	146	133	138

Table 6.1: Average Number of Accidents

fewer accidents during the weekends in the city, as expected. According to the paramedic

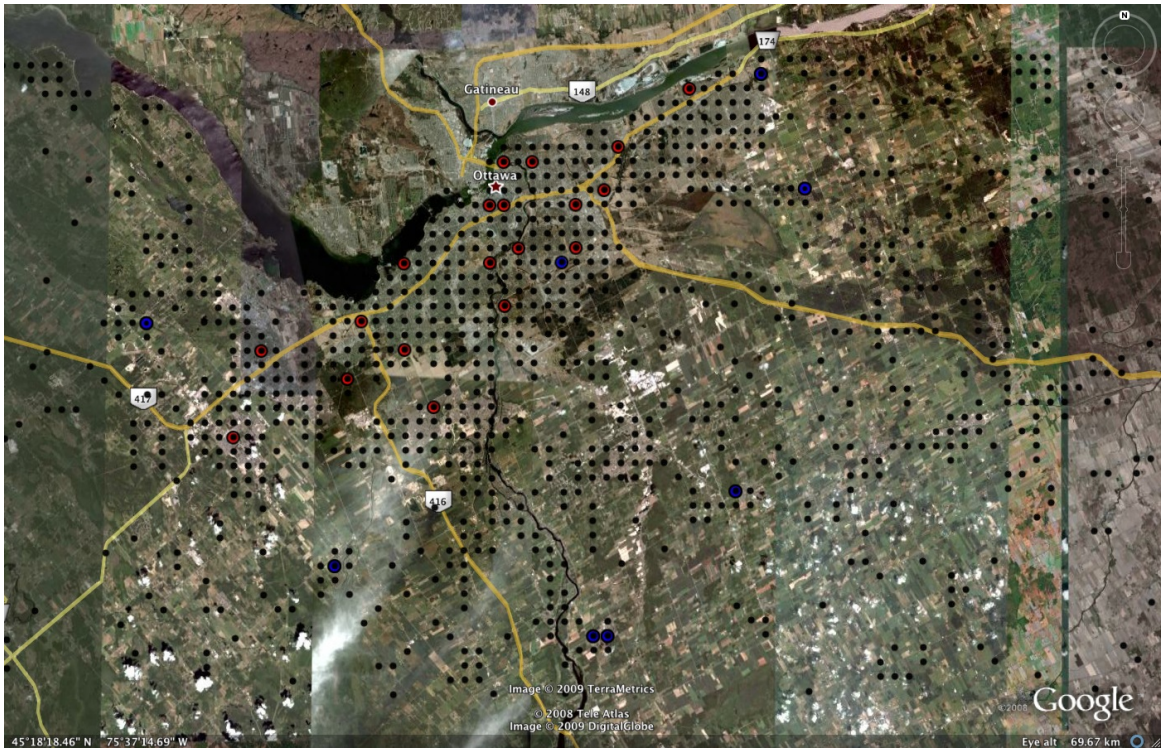

Figure 6.3: Distribution of Accidents (Image courtesy of Google Earth)

centre, there is a maximum time limit of 10 minutes for responding to an accident. After 10 minutes, the probability of losing the patient is very high, especially in priority-4 accidents. So dispatchers in the call centre prefer to make relocations that give them a high probability of having a unit available in the 10-minute time distance of accidents. Considering the average speed for units, the area in the 10-minute time distance of each available unit is considered as the area under coverage.

Chapter 7

Experimental Results and Analysis

We have run a number of experiments to evaluate the efficiency of our models and solution methodology. Based on the RISER case study and its data set, we generate sample days with a number of accidents. The status of posts, units, and demand points are set up for each day. As we start simulating each sample day, we need to dynamically make relocation decisions during the simulation, where we use our models and solution algorithms to find relocations. In the following sections, we provide details on the specification of experiments and the results. We also provide our analysis of these results.

7.1 Specifications

7.1.1 Machine Specifications

All experiments are performed on the Mac OS X, with 2.4 GHz processor dual core and 4 GB of RAM. The code is written in java programing language.

7.1.2 Experiments Specifications

The high-level view of the testing process consists of a simulator and a solver which interact with each other. The simulator reads the initial state of units and posts from input files. It then starts simulating the accidents happening throughout a sample day, reading from another input file. During the simulation, a relocation request is triggered whenever a unit is sent to an accident or becomes available after responding to an accident. At these situations, the simulator calls the solver and passes the required information including the

updated status of units and posts and asks the solver for the best suggested relocations. On the other side, the solver uses the information passed by the simulator as well as the information about demand points to make the best decision. The simulation then continues based on the suggested relocations. As we can see in Fig. 7.1, different algorithms for finding best coverage and best relocation sets can be plugged into the solver to be evaluated during the simulation.

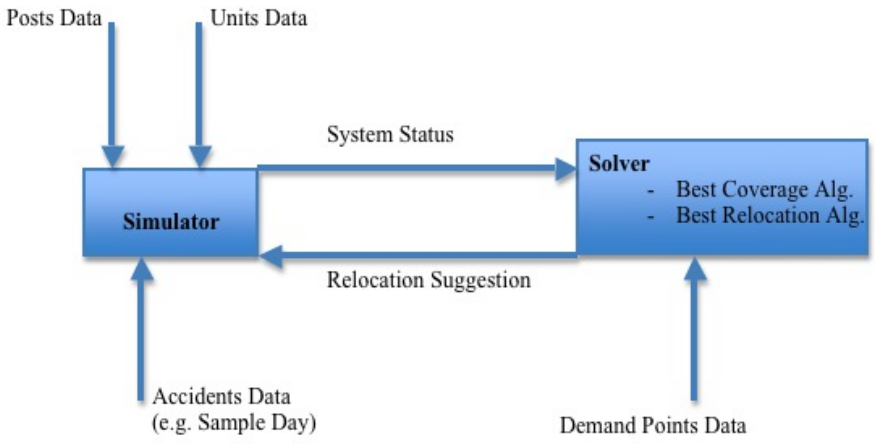

Figure 7.1: Tester Parts

In order to test our approaches, we divide the historical accident data of the RISER case study into two parts. The first part includes the accident data of the 31 days of January, which are used as input to the simulator. All the approaches are tested against these 31 sample days to analyze their efficiency. The second part includes the data from the remaining 116 days, which are used to generate demand points as input to the solver. To create the demand points from the historical data, the area is divided by a grid and all the accidents occurring inside a grid cell are aggregated and represented by a single demand point in the center of that cell. As described in Section 2, demand points can be $time - tagged$ or time-independent. We evaluate the effect of using each type of demand point along with the use of different grid cell sizes.

Experiments are divided into two parts. The first part comprises the experiments on coverage. Simulated annealing, hill- climbing, and mixed-integer programming are compared based on their efficiency in finding the best coverage in different situations. In each situation, the coverage ratio of the found solution is compared to the coverage ratio of the optimum

solution found by exhaustive search. The evaluation criteria for this part are

- **Quality of Solution**: QOS shows how close is the solution to the optimum one. It is calculated as follows:

$$QOS = \frac{CoverageRatio_{solution}}{CoverageRatio_{optimum}}$$

- **Uncovered Ratio%**: This shows the percentage of demand points not covered in the solution.

In the second part, we focus on finding the best set of relocations given the next best configuration. Objectives such as minimizing the *MissedP* based on coverage variation as in Equation 4.6, time, and workload are used to guide the search in finding the best relocations. Algorithms based on these objectives are plugged into the solver, and the simulation results on each of the 31 sample days are used for evaluation. The evaluation criteria used in this part are as follows:

- **Missed Accidents**: Number of accidents that have a response time greater than 10 minutes, also accidents not responded to because of lack of units.
- **Avg. response time**: Average response time for accidents covered within 10 minutes.
- **Stdev. response time**: Standard deviation of response times for accidents covered within 10 minutes.
- **Avg. Driving Time**: Average driving time during the relocations for units. The time units spend at the accident scenes or on their way to the hospital is not considered in this part; we are more concerned about the influence of different relocation strategies on human comfort issues.
- **Max. Driving Time**: Maximum driving time during the relocations for units.

The value of each of these parameters is calculated for each pair of (sample day, algorithm) given as input to our testing process. The overall performance of an algorithm on all days is used for the analysis.

7.2 Results and Analysis

This section presents further detail on all the experiments and their results for each part. In all experiments, the initial 1000 temperature is set for SA and uses an exponential cooling schedule with a cooling factor of 0.995. Travel times are computed using Euclidean metric and a constant unit speed of $60km/h$.

7.2.1 Best Coverage

The first experiment in this section focuses on the performance of simulated annealing in finding the best coverage. The inputs of the experiment are the location of 27 posts in the city, time-independent demand points each representing an area of 2×2 km^2, the maximum number of posts that can be chosen, and the type of heuristic for choosing a neighbouring solution during the search. As described in Section 5.1, three heuristics, Random-Random, Random Del-Max Add, and Random Add-Min Del are suggested for use.

Experiments show that after 25 ms of search using SA, we obtain good improvements in the solution quality. So, for each set of inputs, we let the SA search for 25 ms and save the best visited solution. Fig.7.2 demonstrates the quality of best found solution for each of the heuristics for different problem sizes. According to the chart, all the approaches show good performance with QOS higher than 0.97. Random Del-Max Add is superior to other

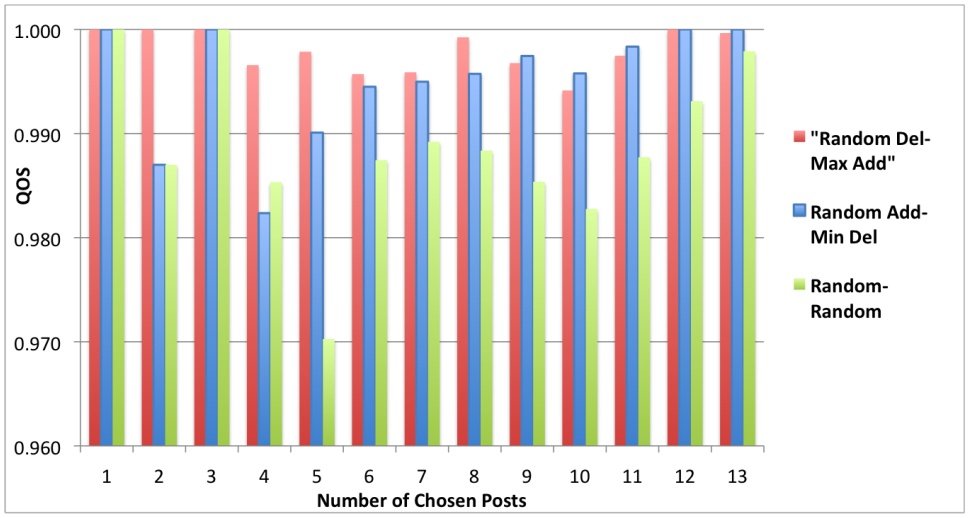

Figure 7.2: Different Neighbourhoods

approaches over all while Random-Random Moves is the worst. There are four cases where Random Add-Min Del outperforms Random Del-Max Add; however, the difference is less than 0.001.

Also we can see that for a small number of chosen posts, SA can find the optimum solution. The quality of the solution declines as the number of chosen posts increases, since the search space as well as the neighbourhood size are increasing. However, the solution quality improves to 1 again as the number of posts is greater than 11. This outcome results from the smoothness of the search space. For the small number of chosen posts, two neighbouring solutions may have very different coverage ratios, resulting in the bumpy search space. For a larger number of chosen posts, the difference between the coverage ratio of two neighbouring solutions decreases. This smoothness will help the search algorithm to easily escape the local optima and continue its move towards the global optimum.

In the next experiment, we let SA (Random Del-Max Add) to search for a longer time, 100 ms, to find the best solution. In this experiment we are not concerned about the running time. Instead, we want to investigate the effect of different grid cell sizes on the coverage

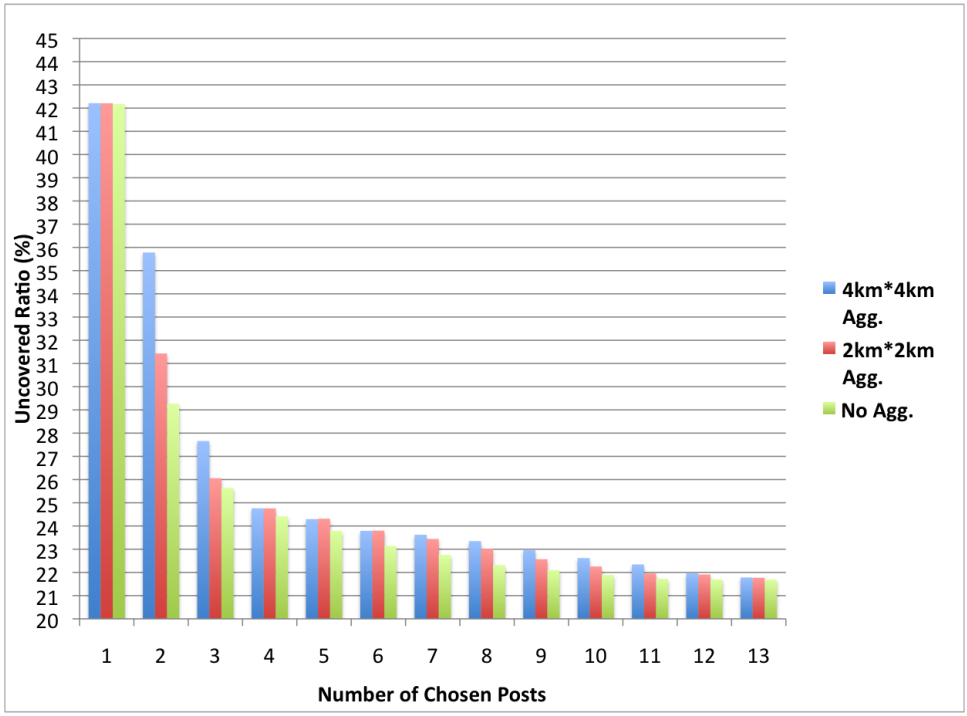

Figure 7.3: Different Aggregations

problem. As specified, historical data are aggregated into grid cells to generate demand points. In Fig.7.3, accidents are aggregated into 4×4 km^2 and 2×2 km^2 grid cells. In another case, each accident is represented by a demand point and no aggregation is used.

According to this chart and our expectations, for all problem sizes, the finer aggregation we use, the less uncovered ratio we obtain. Finer aggregation means more demand points to deal with. It shows the desirability of more efficient and quick algorithms for finding the best coverage.

In the next experiment, we investigate the time efficiency and quality of solution of algorithms other than SA in finding the best coverage. As one of the approaches in solving the optimization model described in Section 4.1 for finding the best coverage, we use the mixed integer programming solver from CPLEX 8.0 [2] student edition along with the AMPL[1] modeling language. The student version restricts us from experimenting with a large number of demand points. However, we are able to solve the problem for 4×4 km^2 and 2×2 km^2 aggregation size, where the number of demand points are 172 and 270. Results show that on this size of data set CPLEX can find the optimum solution quickly. The time efficiency of LP usually considered as the relaxation for MIP is approximately the 4th power of the number of variables [14]. This implies the lower bound on the running time

	Greedy	SA(Random D- Max A)			HC		
n		25 ms	50 ms	100 ms	25 ms	50 ms	100 ms
1	1	1	1	1	1	1	1
2	0.943	1	1	1	0.975	0.994	1
3	1	1	1	1	1	1	1
4	0.955	0.997	1	1	1	1	1
5	0.951	0.995	1	1	1	1	1
6	0.987	0.996	0.997	1	0.995	0.995	0.997
7	0.987	0.998	0.997	1	0.996	0.996	0.999
8	0.988	0.997	1	1	0.997	0.998	0.998
9	0.985	0.998	0.998	0.999	0.996	1	1
10	0.983	0.993	0.999	0.999	0.998	0.998	1
11	0.984	0.998	0.999	1	1	1	1
12	0.992	1	1	1	1	1	1
13	0.998	1	1	1	1	1	1

Table 7.1: QOS for 27 Posts

of MIP and specifically CPLEX, which shows CPLEX is slow on the large data set. Besides this general method, we also look at other *ad hoc* approaches. The greedy approach, SA (Random Del-Max Add) and Hill Climbing (HC) heuristics are evaluated. Both SA and HC start their search from a greedy solution. HC looks for the first best neighbouring solution while examining the whole neighbourhood in order. Getting stuck in the local optima, HC makes a random jump.

Table 7.1 demonstrates the quality of average best found solution for our *ad hoc* approaches in 10 rounds on different problems where n is the number of chosen posts. For all the experiments, accidents are aggregated in $2 \times 2 \ \ km^2$ grid cells.

Based on this table, we can see the greedy approach can find solutions with quality of 94% of the optimal. Also by performing more movements in the search space, SA can increase QOS to 0.993 in 25 ms, and to 0.999 in 100 ms, where 84% of problems are solved to optimality. HC, on the other hand, reaches optimality in 76% of problems in 100 ms. For $n = 2$, it shows weak performance despite the small search space. As outlined earlier, it may be caused by the bumpy search space in this problem, which traps HC in local optima. Results show that SA is more capable in dealing with these situations.

To further investigate the search space, we calculate the probability distribution of uncovered ratio in the neighbouring solutions of the greedy solution. In the case where 2 posts

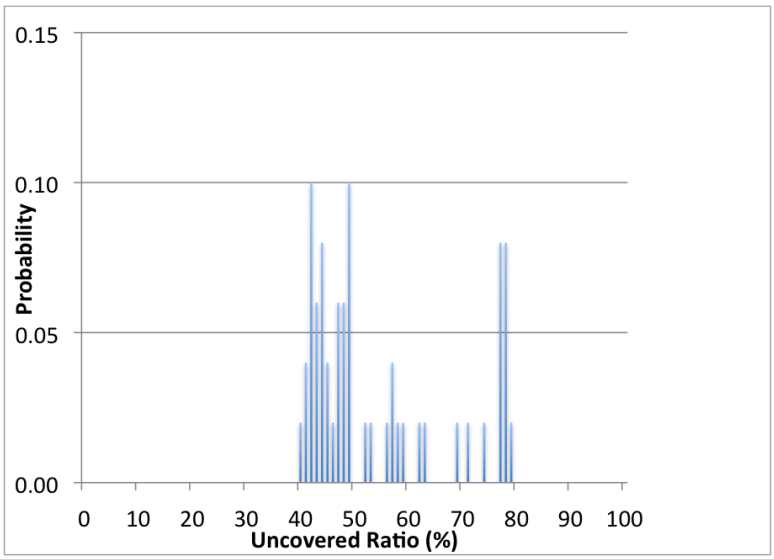

Figure 7.4: Probability Distribution of Uncovered Ratio in the Neighbourhood Area (n=2)

need to be chosen from 27 posts, the greedy solution has an uncovered ratio of 39%. Figure 7.4 shows the probability distribution of uncovered ratio in 50 neighbouring solutions. It is observed that the uncovered ratio changes in the wide range of [40%, 80%] in the neighbourhood area.

However, in Figure 7.5, which considers the case of choosing 10 posts, we see that the uncovered ratio of more than 90% of the neighboring solutions varies in the small range of [23%,30%]. This shows that in this case there is a much smaller difference between the neighbouring solutions than in the previous case. These results confirm our previous idea

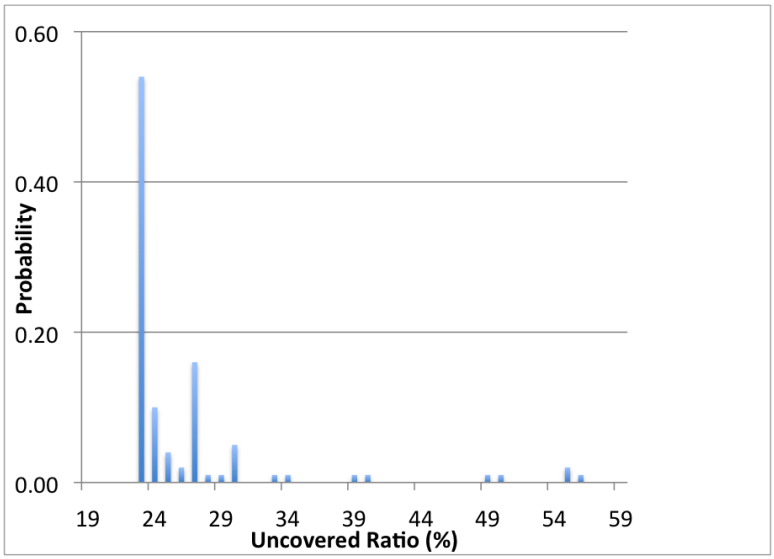

Figure 7.5: Probability Distribution of Uncovered Ratio in the Neighbourhood Area (n=10)

about the smoothness of the search space in different cases.

Back to our results of different algorithms: We see that simple yet fast approaches like SA or even a greedy approach find a solution that is very close to the optimum. Table 7.2 shows the results of the same runs of algorithms, but this time considers a total of 57 post locations in the city. The increase in the number of post locations expands the search space. The table demonstrates the uncovered ratio of the best found solution for each run. Starting from the greedy solution, we can see improvements as the time of the search increases. Apart from $n = 5$, HC outperforms SA. In the case of $n = 5$, HC seems to be trapped in a local optimum.

In the last experiment, we want to investigate the effect of different demand point types

n	Greedy	SA(Random D- Max A)			HC		
		25 ms	50 ms	100 ms	25 ms	50 ms	100 ms
1	49.48	49.48	49.48	49.48	49.48	49.48	49.48
3	29.77	29.77	29.77	29.77	29.77	29.77	29.77
5	26.37	25.01	24.99	24.98	25.07	25.04	25.01
7	23.87	23.75	23.64	23.58	23.56	23.52	23.49
9	22.79	22.67	22.61	22.55	22.50	22.46	22.43
11	22.07	22.05	22.05	22.07	22.01	22.01	21.99
13	21.77	21.75	21.73	21.70	21.71	21.66	21.66
15	21.60	21.56	21.52	21.56	21.52	21.51	21.49
17	21.50	21.47	21.47	21.45	21.45	21.44	21.43
19	21.44	21.42	21.40	21.40	21.40	21.39	21.37
21	21.39	21.38	21.38	21.38	21.37	21.37	21.37
23	21.38	21.37	21.37	21.37	21.37	21.37	21.37

Table 7.2: Uncovered Ratio for 57 Posts

and the corresponding coverage on the performance of the system throughout the day. For this purpose, we assume that at the time of accident, the best coverage is achieved and all the available units are residing at the posts. According to this assumption, the relocation time is zero. The historical accidents are aggregated into 2×2 cells to generate demand points. The results of the simulation of 31 sample days are shown in Table 7.3. The first two rows correspond to the cases in which historical data are aggregated without considering the time. In the rest of the rows, we see the output of the cases in which data are aggregated based on different time intervals.

It turns out that if we partition the whole life cycle into 8-minute intervals, we can get better coverage according to fewer missed accidents while the average response time is almost the same. We have not captured any specific relation between the 8-minute time intervals and the data set of accidents. So, we are not sure why the system is producing better results using 8-minute intervals. But we can suggest that looking at different time intervals or even simulating their effects on the system can be helpful in making the system more efficient.

Type	Total Missed Accidents	Avg. Respond Time (m)	Stdev. Respond Time (m)
CPLEX Time Independent	110	4.15	0.4
SA Time Independent	110	4.15	0.4
SA Time Interval=4 min	114	4.16	0.35
SA Time Interval=8 min	103	4.17	0.39
SA Time Interval=16 min	111	4.16	0.4
SA Time Interval=32 min	111	4.13	0.43

Table 7.3: Effect of Time-Tagged Demand Points

7.2.2 Best Relocation Set

In this part of the experiment, we consider three different approaches for finding the best set of relocations, given the set of best posts to cover. As described previously, each set of relocations is basically a one-to-one mapping from the set of available units to the set of these posts. In the first approach, Max Time, we are looking for the set of relocations whose maximum travel time is minimized. The algorithm 5.2.1 based on Munkres is used to find the optimum solution. The second approach, Max WorkLoad, has more focus on workload of the units. This time, we are looking for the set of relocations which imposes less workload on the units. The transportation algorithm is used to find the optimum solution, where the cost of each pair of (u, p) is the total driving time of unit u up to the time of the current relocation, plus the time required by u to get to p. In the third approach, Coverage Variation, we use our coverage variation based model introduced in Section 4.2 to find the best relocation set.

Table 7.4 shows the result of simulating 31 sample days while plugging different approaches into the solver. For all the experiments, we aggregate historical accidents in 4×4 km^2 squares without considering the time. The solver uses pre-computed optimum solutions in its best coverage finder part. For each sample day, the simulation is run for 13 hours starting from 05:30 am as units start their 13-hour shifts. Different numbers of units,

12, 17 and 22, are considered to be available in the morning to evaluate the result of adding units to the system.

Specification		Missed Accidents	Avg. Respond Time (m)	Avg. Driving Time (m)	Max. Driving Time (m)
Units=12	Max Time	120	4.21	125.15	196.33
	Max WorkLoad	124	4.19	110.22	127.09
	Coverage Variation	115	4.18	149.42	220.72
Units=17	Max Time	77	3.78	97.26	177.6
	Max WorkLoad	76	3.83	84.31	115.80
	Coverage Variation	73	3.93	105.27	186.20
Units=22	Max Time	66	3.51	66.33	130.86
	Max WorkLoad	70	3.55	61.48	88.64
	Coverage Variation	63	3.52	68.81	131.32

Table 7.4: Comparison of Different Approaches for Relocation Based on Time, WorkLoad, and Coverage Variation. (All the available units take part in the relocation.)

It is observed that the coverage variation approach manages to save more accidents while keeping the average response time the same. However, the average and maximum driving times of the units are increased. Max WorkLoad, on the other hand, manages to decrease the driving time in comparison to the two other approaches, but the number of missed accidents is greater.

Adding more units to the system, as we expect, decreases the number of missed accidents. It has a favourable impact on the other parameters as well. By just adding five more units to the system, the number of missed accidents drops by 30%.

To justify the effect of coverage variation on relocation decisions, we capture the changes of the coverage over time at the time of one random relocation. Starting from the same initial state and the same set of posts to be covered, Max Time and Coverage Variation return two different sets of possible relocations. The simulator follows two scenarios. In each, it relocates units based on one of these sets. The uncovered ratio is evaluated every minute. Fig.7.6 demonstrates the two plots captured during these two scenarios.

The set of relocations suggested by the coverage variation approach outperforms by keeping the uncovered ratio lower during the relocations. It gives the system a higher chance of saving the accidents occurring during the relocations. And this is what we observed in

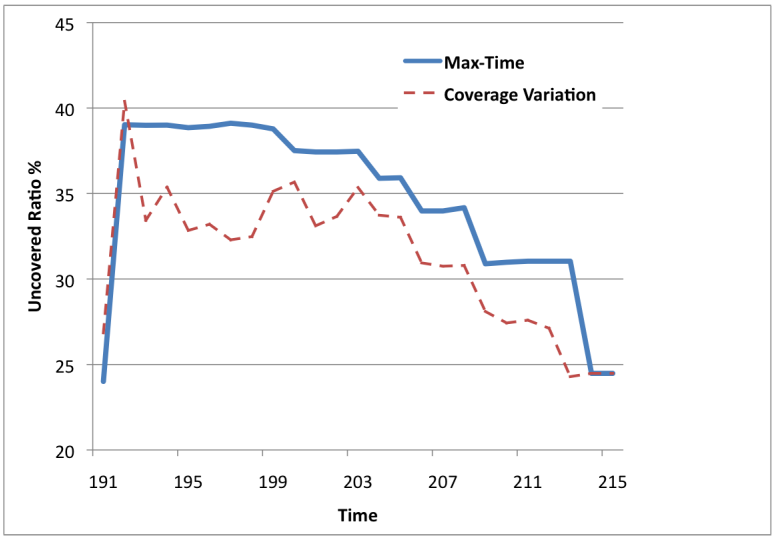

Figure 7.6: Coverage Variation over Time for Max Time vs. Coverage Variation

our results in Table 7.4 as well.

To do more investigation and to get a visual overview of how coverage variation can avoid missed accidents, we concentrate on the status of the system (location of units and their destinations if moving) just before the occurrence of an accident that is saved by this approach. We also investigate the status of the system at the same time using the Max Time approach. We find out that in Max Time the only unit in the coverage area of the

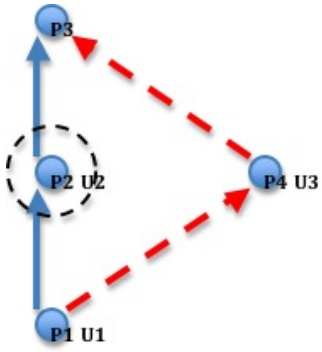

Figure 7.7: Two Different Relocation Outputs. (Dashed lines show the relocation suggested by Coverage Variation. Relocations suggested by Max Time are demonstrated by the solid lines.)

accident has left the area based on the relocation decisions. However, Coverage Variation

will manage to keep the unit in the area, which results in saving the accident. In both cases, a relocation request is triggered minutes before the accident. Fig. 7.7 shows the status of the system and the different relocations suggested.

According to this figure, at the time of the relocation, units U_1, U_2 and U_3 should relocate from corresponding posts P_1, P_2 and P_4 to posts P_2, P_3 and P_4. Knowing that U_2 residing at P_2 is covering a very high demand area inside the dashed circle, it is not wise to relocate this unit and leave the area uncovered for a period. As we see in the figure, Max Time suggests unit U_1 to go to P_2 and U_2 to go to P_3. This set of relocations is very quick, but it may cause the accident to be missed in the high-demand circle. However, Coverage Variation chooses the longer relocations by sending U_1 to P_4 and U_3 to P_3. This relocation lets U_2 remain in P_2.

Chapter 8

Conclusion

In this thesis, we considered the real-time relocation problem of emergency medical services. A two-phase approach was described to dynamically find the best set of relocations based on the status of ambulances and potential waiting sites throughout the city at the time of the relocation. A set of demand points in the city was used to represent the distribution of previous and, it is hoped, of future accidents.

In the first phase, the goal was to find the next best set of posts to be covered in order to maintain the maximum coverage. Local search techniques such as Simulated Annealing and Hill Climbing were used to search the solution space starting from an initial greedy solution. Different heuristics were used to move in the neighbourhood area. The performance of these methods was evaluated against the standard MIP techniques as well as optimum solutions found through an exhaustive search.

The output of the first phase was then used as an input to the second phase. In this phase, we considered the available units and their current locations and developed a relocating schedule that assigned each unit to either a new post or reassigned them to their current post. The goal was to cover the posts found in the first phase. We found routes that gave us higher coverage during the transition of ambulances from one configuration to the new one. We showed the effect of our method by experiments using a real data set. The overall outcome of this thesis showed that this new approach provided a better solution.

8.1 Contributions

We can summarize the contributions of this thesis as follows:

- We compared the performance of local search heuristics vs. MIP on finding the best coverage. Experiments showed that local search heuristics were able to find solutions 99% close to optimum in less than 25 msec. Increasing the time of search to 100 msec, they would find optimal solutions. The results confirmed that by replacing these *ad hoc* algorithms with standard methods (*e.g.*, MIP), we were still able to find optimum or very close to optimum solution very quickly, whereas standard methods (*e.g.*, MIP) might require long processing time on large data sets.

- We investigated the impact of historical accident data aggregation into demand points on the performance of the system in dealing with unknown future accidents. Demand points served as the leaders in our search space and the goal was to increase their coverage in both phases. So they should be carefully generated.

  Results using different aggregation sizes showed that the finer aggradation we had, the better we could represent the distribution of accidents. Also if we considered the time of occurrence of accidents in our historical data, we could generate time-tagged demand points that could lead the search more efficiently in different time intervals.

- Knowing the best set of posts to cover as destinations, there were multiple relocation sets, which could move units from their current locations to destinations. Different objectives could be used to define the best set of relocations, such as time, workload, *et cetera*. We defined a new objective based on the coverage variation during the relocations. The goal was to minimize uncovered area during the relocations (movements). In the imaginary ideal world, units could appear at destinations in a second, which would give us zero uncovered area during the move. The experimental results showed that we could miss fewer accidents by using this objective vs. minimizing the driving time or workload. Results demonstrated that we would have a higher chance of saving accidents that occur during the relocations by applying our new objective to the search methods.

8.2 Future Work

For future work we are thinking of adding the real city map into our system. Now, we assume the route between two locations to be the direct line between them. However, in the real city, there can be multiple routes between two locations. This adds more complexity

in finding the best relocations set, in which we should not only define the destination of each unit, but also the best route to get to the destination. We want to investigate the performance of our objective in this situation in comparison with time or workload-based objectives. Also, we can expand our approach and look at the changes of other parameters such as workload during the relocations. In this way, we can control the human comfort issues as well.

The idea of dividing the problem into two phases, trying to find the set of best posts independently of the current locations of units at posts and then trying to attain this new configuration can be examined in more detail in future. Is it really efficient to partition the problem into two independent subproblems, or is there a way to combine the subproblems and get better results?

Bibliography

[1] Ampl, http://www.ampl.com/index.html.

[2] Cplex, http://www.ampl.com/downloads/cplex80.html.

[3] O.I. Alsalloum and G.K. Rand. Extensions to emergency vehicle location models. In *Computers and Operations Research*, volume 33, pages 2725–2743. Elsevier, 2006.

[4] T. Andersson, S. Petersson, and P. Värbrand. Calculating the preparedness for an efficient ambulance health care. In *Proceedings of 7th International IEEE Conference on Intelligent Transportation Systems*, pages 785– 790, Washington, DC, 2004.

[5] T. Andersson, S. Petersson, and P. Värbrand. Dynamic ambulance relocation for a higher preparedness. In *Proceedings of 35th Annual Meeting of Decision Sciences Institute*, Boston, MA, 2004. DSI.

[6] T. Belshe, J. Goodhue, and J. Yeh. Emergency medical services vehicle redeployment recommendation based on complete system state analysis (ems redeploy). In *Systems and Information Engineering Design Symposium,IEEE*, pages 298–303, April 2006.

[7] R. Bent and P. Van Hentenryck. Regrets only! online stochastic optimization under time constraints. In *AAAI*, pages 501–506, San Jose, CA, 2004. AAAI Press / The MIT Press.

[8] R. Bent and P. Van Hentenryck. The value of consensus in online stochastic scheduling. In *ICAPS*, pages 219–226, Whistler, BC, 2004. AAAI.

[9] R. Bent and P. Van Hentenryck. Online stochastic optimization without distributions. In *ICAPS*, pages 171–180, Monterey, CA, 2005. AAAI.

[10] L. Brotcorne, G. Laporte, and F. Semet. Ambulance location and relocation models. In *European Journal of Operational Research*, pages 451–463. Elsevier, 2003.

[11] S. Budge, A. Ingolfsson, and E. Erkut. Optimal ambulance location with random delays and travel times. In *Health Care Management Science*, volume 11, pages 262–274. Springer, 2008.

[12] H. S. Chang, R. Givan, and E. K. P. Chong. On-line scheduling via sampling. In *AIPS*, pages 62–71, Breckenridge, CO, 2000.

[13] R.L. Church and C.S. ReVelle. The maximal covering location problem. In *Papers of the Regional Science Association*, volume 32, pages 101–118. Regional Science Association, 1974.

[14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. The MIT Press, 2001.

[15] M.S. Daskin. A maximum expected location model: Formulation, properties and heuristic solution. In *Transportation Science*, volume 7, pages 48–70. INFORMS, 1983.

[16] M.S. Daskin. Location, dispatching, and routing model for emergency services with stochastic travel times. In *Spatial Analysis and Location Allocation Models, A. Ghosh and G. Rushton (eds.)*. Van Nostrand Reinhold Company, 1987.

[17] G. Erdogan, E. Erkut, and A. Ingolfsson. Ambulance deployment for maximum survival. In *Naval Research Logistics Quarterly*, volume 55, pages 42–58. Wiley, 2008.

[18] E. Erkut, A. Ingolfsson, T. Sim, and G. Erdogan. Computational comparison of five maximal covering models for locating ambulances. To appear in *Geographical Analysis*.

[19] M. Gendreau, G. Laporte, and F. Semet. Solving an ambulance location model by tabu search. In *Location Science*, volume 5, pages 75–88. Elsevier Science, 1997.

[20] M. Gendreau, G. Laporte, and F. Semet. A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. In *Parallel Computing*, volume 27, pages 1641–1653. Elsevier Science, 2001.

[21] M. Gendreau, G. Laporte, and F. Semet. The maximal expected coverage relocation problem for emergency vehicle. In *Journal of the Operational Research Society*, volume 57, pages 22–28. Palgrave Macmillan, 2006.

[22] P. Van Hentenryck and R. Bent. *Online Stochastic Combinatorial Optimization*. The MIT Press, 2006.

[23] P. Van Hentenryck, R. Bent R., and E. Upfal. Online stochastic optimization under time constraints. Technical report, Brown University, Providence, RI, 2005.

[24] J. Hopcroft and R. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. In *SIAM Journal on Computing*, volume 2, pages 225–231. SIAM, 1973.

[25] D. Mitra, F. Romeo, and A. Sangiovanni-Vincentelli. Convergence and finite-time behavior of simulated annealing. In *Advances in Applied Probability*, volume 18, pages 747–771. Applied Probability Trust, 1986.

[26] J. Munkres. Algorithms for the assignment and transportation problems. In *Journal of the Society for Industrial and Applied Mathematics*, volume 5, pages 32–38. SIAM, 1957.

[27] C.S. ReVelle and K. Hogan. The maximum availability location problem. In *Transportation Science*, volume 23, pages 192–200. INFORMS, 1989.

[28] J. Schneider and S. Kirkpatrick. *Stochastic Optimization*. Springer, 2006.

[29] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley and Sons, 1998.