A GENETIC ALGORITHM FOR THE MULTI-LEVEL MAXIMAL COVERING AMBULANCE LOCATION PROBLEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MESUT KARAMAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

SEPTEMBER 2008

Approval of the thesis:

A GENETIC ALGORITHM FOR THE MULTI-LEVEL MAXIMAL COVERING AMBULANCE LOCATION PROBLEM

submitted by **MESUT KARAMAN** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ___________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Nur Evin Özdemirel ___________________
Head of Department, **Industrial Engineering**

Asst. Prof. Dr. Sedef Meral ___________________
Supervisor, **Industrial Engineering Dept., METU**

Examining Committee Members:

Assoc. Prof. Dr. Canan Sepil ___________________
Industrial Engineering Dept., METU

Asst. Prof. Dr. Sedef Meral ___________________
Industrial Engineering Dept., METU

Asst. Prof. Dr. Seçil Savaşaneril ___________________
Industrial Engineering Dept., METU

Asst. Prof. Dr. Ferda Can Çetinkaya ___________________
Industrial Engineering Dept., Çankaya University

İnci Yüksel, M.Sc. ___________________
Expert Engineer, ASELSAN

Date: September 5, 2008

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Mesut Karaman

Signature :

ABSTRACT

A GENETIC ALGORITHM FOR THE MULTI-LEVEL MAXIMAL COVERING AMBULANCE LOCATION PROBLEM

Karaman, Mesut

M.Sc., Department of Industrial Engineering
Supervisor: Asst. Prof. Dr. Sedef Meral

September 2008, 116 pages

The emergency medical services (EMS) provide the preliminary assistance and transportation for patients in need of urgent medical care in order to decrease the mortality rate and reduce the non-reversible effects of injuries. Since the objective is directly related to the human life, the value of the proposed solutions in order to improve the performance of EMS is highly welcomed. Mainly, there are three problems that EMS managers face with: location, allocation and redeployment of the EMS facilities/vehicles. Most of the studies in EMS literature focus on accurately modeling the probabilistic nature of the availability of an ambulance when it is called for. However, trivial changes in model parameters or estimates could dramatically change the optimal allocations generated by the probabilistic models and hence make the model invalid. In this study, we formulate the ambulance location problem as a deterministic multi-level maximal coverage model by which the total demand is tried to be covered as many as possible at multiple levels. Both a mathematical programming model and genetic algorithm-based heuristic approaches are proposed for the problem. The results indicate that the genetic algorithm-based solutions give reliable (near-optimal) and robust results in reasonable computational times for the problem. Moreover, the tradeoffs between the two performance measures, ‘responsiveness’ and ‘preparedness’, are searched for; and our approaches with multi-level coverage are compared against the multiple coverage approaches in terms of these performance measures.

Keywords: emergency medical service, ambulance, location-allocation, coverage models, genetic algorithm, metaheuristics.

ÖZ

ÇOK-SEVİYELİ EN FAZLA KAPSAMALI AMBULANS KONUMLANDIRMA PROBLEMİ İÇİN BİR GENETİK ALGORİTMA

Karaman, Mesut

Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Y. Doç. Dr. Sedef MERAL

Eylül 2008, 116 sayfa

Acil sağlık hizmetleri, hasta ve yaralılar için temel acil ilkyardım müdahalelerini yapmayı ve gerekli ulaşımı sağlayarak yaralanmaya bağlı kalıcı hasarlar ile ölüm vakalarını azaltmayı amaç edinmiştir. Konu insan hayatı olduğu için, acil sağlık hizmetlerinin sağlanması ve eniyilenmesi ile ilgili olarak geliştirilen her türlü çözüm yüksek oranda kabul görmektedir. Bu alanda ilgili kişilerin karşılaştığı üç temel problem, acil sağlık hizmeti veren araçların (ambulans) konumlandırılması, tekrar yerleştirme ve sorumlu olduğu bölge tayini problemleridir. Acil sağlık hizmetleri literatüründeki birçok çalışma, bir ambulansın aramaya cevap verememe durumundan kaynaklanan olasılıksal durumun çözümüne odaklanmıştır. Diğer taraftan, bu tür modeller içerisindeki parametre ve tahminlerdeki küçük değişiklikler veya yapılacak hatalar modelin bütününde yol açacağı değişikliklerden ötürü modeli uygulanamaz hale getirebilmektedir. Bu çalışmada, üç düzeyli kapsama standardı önerilerek, ambulans ihtiyacı içerisinde bulunan talep noktalarının kapsanması amaçlanmaktadır. Problem ile ilgili olarak, bir matematiksel model ve genetik algoritma çözümüne dayalı sezgisel yaklaşımlar geliştirilmiştir. Sonuçlar, geliştirilen genetik algoritmanın çok düzeyli kapsama problemi için güvenilir ve hızlı çözümler verdiğini göstermektedir. Ek olarak, aramalara hızlı cevap verebilme ve sistem hazırganlığı gibi iki performans kriteri kullanılarak, geliştirilen çok düzeyli kapsama modelininin çoklu kapsama modelleri ile karşılaştırılması yapılmıştır.

Anahtar Kelimeler: acil sağlık hizmetleri, ambulans, konumlandırma-atama, kapsama modelleri, genetik algoritma, modern sezgisel yöntemler.

To my family

ACKNOWLEDGEMENTS

I would like to express my sincere gratefulness to my supervisor Assist. Prof. Dr. F. Sedef MERAL for her guidance, suggestions and encouragement throughout my studies. Her support and friendly attitude always motivated me and increased my belief in completing the research and coming out with good results.

My special thanks go to Erhan ÖRÜMLÜ and Taner GÜLEZ, for their continuous support in problem coding and detection of the errors in the test runs.

I also appreciate the support of my company, ASELSAN A.Ş. by providing the permission for attending such a great M.S. program. Also, the contribution by TUBİTAK's scholarship program deserves to be mentioned in terms motivating me to be successful.

This thesis would have never been completed if it had not been for Ruken YÜCEL, who was always there with me. I thank her with all my heart for her constant support and belief in me.

Finally, I would like to thank my family for their understanding and patience and faith in me, especially to my mom, grandmother, grandfather for their lovely support and to my uncle who directed me to perform an academic study.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

The aim of emergency medical services (EMS) is to reduce the mortality and irreversible effects of injuries on people by providing urgent transportation and intervention to the patients. However, it is impossible to place an ambulance in every corner of the streets. EMS managers and administrators frequently face with the challenging task of locating a limited number of ambulances in a way that provides the best results in terms of covering the entire population. The term "coverage" is defined in the EMS Act of 1973 as follows: in urban areas 95 percent of calls must be reached in 10 minutes, and in rural areas, calls should be reached in 30 minutes or less. Although the nature of the problem resembles the well-known set covering or maximal expected location coverage problems, the probability that any given ambulance is busy, enforces us to develop improved models by which the uncertainty could be handled as long as coverage and quick response objectives are considered.

The approaches for the ambulance location problem are divided into two categories: deterministic models and probabilistic models. The accuracy of the parameters in probabilistic models could considerably affect the overall model's performance and they are exposed to errors just because of the limited availability of simulating emergency cases. Hence, we prefer to work for a deterministic environment in this study. On the other hand, in order to overcome the shortcomings of considering such a deterministic environment and to handle the conditions of missing calls, we intend to incorporate the idea of multi-level coverage, specifically the triple coverage standards in our study.

As previously stated, the environment we work on throughout this study is deterministic in which the demand at the populated sites and the possible location sites are known in advance. However, the possibility of change in demand patterns, for example in different time zones of the day, is taken into consideration. By detecting the possible movements of the demand points, the need for ambulances is assessed and the model can be used continuously to find the optimal places for the changed conditions. The changes in the system parameters could be because of the different time periods or a meeting in a known square, a sport activity or any considerable coverage decrease in a zone due to heavy ambulance activity just because of a big traffic accident. In addition to these, an ambulance placed at a site could be assigned to any call, if the demand point is in any coverage of the ambulance. So, no demand point is attached specifically to an ambulance before any call.

Imposing some absolute or partial coverage standards may lead to infeasible solutions if they are too tight. In fact, no approach can satisfy a feasible solution at every situation, since demand for ambulances is highly stochastic and conditions might occur where there are not adequate ambulances or the coverage fails in some areas. In order to provide a "good" level of coverage and not to fall in infeasibility, several remedial actions can be taken like: increasing the number of available ambulances or relaxing the higher level coverage standard and partial coverage standards, or not specifying the number of ambulances apriori (Hogan and Revelle, 1986; Marinov and Revelle, 1996; Gendreau et al., 1997).

The set covering and maximal covering problems are classified as NP-hard in the literature while targeting one level of coverage. While trying to overcome the stochastic nature of the missing call problem we introduce the triple coverage standard, making the problem much more extended in terms of the decision variables. That is why classical mathematical modeling approaches seem to be disadvantageous in terms of computation time, hence, we resort to metaheuristics.

The most common approaches in ambulance location studies are based on 'set covering' models, 'maximal expected coverage' location models, queuing theoretic models, hypercube models and the other extended probabilistic location models. In most of these studies, the vehicle type is single, and relocation aspects are ignored.

By using the three-level coverage standards, our aim is to provide a solid coverage for the demand points and increase the probability of quick response to a call by locating service units with respect to coverage levels. In a single coverage standard, a demand point is tried to be covered by a service unit as close as possible. However, the demand points which are probably in a partial coverage range of the ambulance are left out of consideration. Based on the three-level coverage idea, the demand points, which are slightly distant from the tighter coverage standard, are taken into account. We do not consider more than three levels in multiple coverage; because especially in emergency cases, an ambulance station located after some distant point does not contribute to the demand coverage quality of the ambulances, since the response time of that distant unit turns out to be longer than critical response time stated by the EMS Act.

The inability of the mathematical programming solvers to provide optimal or near-optimal solutions in reasonable times even for medium-sized location problems has led us to search for the use of metaheuristics for our three-level coverage problem. Problems with up to 500 demand points and 10 ambulances could be solved by the available capable solvers. When the number of demand points is increased to 1000, the computer memory (Intel Centrino Duo CPU 1.66 GHz, 1 GB of RAM) fails to solve the model. For this reason, we have investigated the principles of some metaheuristics and applied an appropriate method; a genetic algorithm (GA) based solution approach for our three-level coverage idea. Our approach is unique in a sense that it tries to maximize the three levels of coverage for a demand point and utilizes a genetic algorithm. The solution quality of our genetic algorithm then has been compared against both the exact optimal solution of the model and another solution obtained by a well known greedy ADD and DROP heuristic.

As far as the additional issues are considered, our method is applicable to relocation problems because of its rapid solution capability in changing conditions either because of changes in the demand pattern or ambulance conditions. The new locations for ambulances providing three levels of coverage in changing conditions are found in a reasonable computation time and then a redeployment problem needs be solved. Obviously, the redeployment problem of ambulances is an easy problem if the related assignment cost matrix is known in advance. Also, our study contributes to the ambulance location theory with its analysis on the comparison of the multi-level coverage approaches and backup coverage/multiple coverage approaches. In the comparison analysis, the performance metrics 'responsiveness' and 'preparedness' are used in which 'preparedness' corresponds to capability of decreasing the level of missing calls as much as possible. While multi-level coverage idea favors quick response criterion, multiple coverage idea increases the system-wide preparedness. Hence, the comparison results obtained provide useful insights for the decision makers in making use of both multi-level and multiple coverage ideas.

In the following chapters, the details of our study are presented in the sequence of below outline.

Chapter 2 introduces overview of the location and relocation models developed for the EMS services and especially for ambulances. Chapter 3 focuses on the problem environment and presents the mathematical programming formulation. Chapter 4 presents the proposed solution approaches. Chapter 5 presents the results of the computational study performed. Finally Chapter 6 concludes with the discussions on the proposed approaches, presents the remarks derived from the study and suggests some issues for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Deterministic Approaches

The idea of locating emergency service facilities or so called public facilities by using mathematical tools had come out around the beginning of 70s. Prior to this date, the studies based on mathematical analysis had focused extensively on the solution of commercial problems; locating a new machine in a manufacturing environment or a warehouse in a distribution network was among the issues that had been studied. In one of the initial studies ReVelle et al. (1971) explain the reason as follows: in the area of private facility location problems, the starting point is much more identifiable, since the cost elements included in the objective function of the commercial facility location problems can be reasonably estimated and the models can well represent the real location problem they are designed to solve. For example, in the area of private facilities location analysis, an accurate objective function in locating a warehouse is to minimize the cost of manufacturing and distribution.

As far as the public side is concerned, the approach has to be redefined in terms of the objectives. Two different measures which have received attention in location models are: (1) total weighted distance or time from/to the facilities, and (2) distance or time that the most distant user from a facility would have to travel to reach that facility, that is, the maximal service distance. The difficulty of defining optimal service* distance in emergency situations requires much more attention in locating public sector facilities.

---

* Accepted time threshold written in the EMS Act of 1973 requires that in urban areas 95 % of requests be reached in 10 minutes, and in rural areas, calls should be reached in 30 minutes or less.

Among the initial studies on emergency services location problems, the study by Toregas, ReVelle and Bergman (1971) views the location of emergency facilities as a 'set covering' problem with equal costs in the objective and linear programming model is formulated to solve the set covering problem. A single cut constraint is added to resolve the fractional solutions. The model is defined as follows:

Minimize $z = \sum_{j=1}^{i=n} x_j$

Subject to: $\sum_{j \in N_i} x_j \geq 1, \quad (i = 1,2,\ldots,n)$

$x_j = (0,1) \qquad (j = 1,2,\ldots,n)$

where:

$$x_j = \begin{cases} 0, \text{ if no facility is established at point } j \\ 1, \text{ if a facility is established at point } j \end{cases}$$

N_i: the set of points in the distance of 'S' units to the demand point *i*.

The model is basically minimizing the number of facilities to be located while targeting the required service distance, S, for each demand point. The cut constraint used is simply,

$$z = \sum_{j=1}^{i=n} x_j \geq \lfloor m^0 \rfloor + 1,$$

where $\lfloor m^0 \rfloor$ is the integer part of the initial solution.

The first critical point in the model is to allow for a number of ambulances as much as the location sites. However, in real life, the number of ambulances is limited.

Furthermore, the model does not capture the unavailability of the ambulances. That is to say, during a call, the covering ambulance may be busy due to a previous call within the covering distance. On the other hand, the model is valid in the EMS literature in the sense that it provides a solution to the public facility problem via mathematical modeling and at least a bound in order to satisfy the required service level; it is and classified as “Location Set Covering Model” (LSCM).

Another model that needs to be analyzed is the one developed by Church and ReVelle (1974). This model tries to deal with one of the shortcomings of the previous model, i.e. LSCM. Rather than allowing for an unlimited number of ambulances, the model seeks to maximize the population coverage by the limited ambulance usage. This problem is classified as the “Maximal Covering Location Problem” (MCLP).

Defined on a network of nodes and arcs, a mathematical formulation of this problem can be stated as follows:

(MCLP)

Maximize $z = \sum_{i \in I} a_i y_i$

Subject to: $\sum_{j \in N_i} x_j \geq y_i$ for $\forall \ i \in I$

$\sum_{j \in J} x_j = P$

$x_j = (0,1)$ for $\forall \ j \in J$

$y_i = (0,1)$ for $\forall \ i \in I$

where

I : the set of demand nodes,

J : the set of facility sites,

S : distance beyond which a demand point is considered “uncovered”,

d_{ij} : the shortest distance from node i to node j ,

$$x_j = \begin{cases} 0, & \text{if no facility is established at point } j \\ 1, & \text{if a facility is established at point } j \end{cases}$$

$$y_i = \begin{cases} 0, & \text{if the demand node } i \text{ is not covered} \\ 1, & \text{if the demand node } i \text{ is covered} \end{cases}$$

$N_i = \{ j \in J \mid d_{ij} \leq S \}$,

a_i = population to be served at demand node i,

P = number of facilities to be located,

Church and ReVelle (1974) list two solution techniques for the above formulation. One is the heuristic approaches. Greedy Adding Algorithm (GAA) and modified version of it (Greedy Adding with Substitution-GAS) are utilized. Secondly, a linear programming approach is proposed. However, in the linear programming approach, the solution reveals two cases:

Case 1: All x_j, $\overline{y_i}$ = (0,1), which is called an "all integer answer".

Case 2: Some x_j's are fractional, which is called a "fractional answer".

Here, it has to be noted that the linear programming approach is utilized in another version of the problem. In this version of the problem, they define the problem in such a way that rather than maximizing the covered population, they seek to minimize the uncovered population (the decision variable $\overline{y_i}$ is introduced in this second version of the problem definition). When the Case 2 above is faced with, they eliminate the fractional variables either by the method of inspection or the method of Branch and Bound. By assuming the same cost for every establishment of emergency services at location sites, they help in the generation of a cost-effectiveness curve. By increasing the number of ambulances allowed, it is possible to measure the percentage of population covered. This trade-off curve provides valuable information for the decision makers.

They extend their model by the addition of another constraint to their models. By this way, they aim to maximize the population that can be covered within a given service distance S while at the same time ensuring that the users at each point of demand find a facility no more than T units of distance ($T > S$) away. The extended model is also called as MCLP *with mandatory closeness constraints.*

The model does not address the issue of not answering the calls because of busy units. But it has the ability of capturing the real life situations more extensively than the previous model, since it brings limits to the number of ambulances and takes into consideration the mandatory closeness aspect. Eaton et al. (1985) have used MCLP to plan the reorganization of the emergency medical services in Austin, Texas. The developed plan has reduced the total cost of emergency services by $4.6 million in terms of both construction and operation in 1984.

The two approaches above do not still eliminate the problem of missing calls due to the unavailability of medical units. In the deterministic sense, after these two studies are introduced, some models have been developed to handle this issue. In addition to this, by the extended versions of the above models, real life situations have been handled in a broader sense. One of the issues that have been studied is addressing problems with several vehicle types. The model developed by Schilling et al. (1979) is called as tandem equipment allocation model (TEAM). The model was particularly developed for fire fighting purposes. However, it is related with emergency medical services, since there are two basic ambulance vehicles, equipped with different types of utilities, called Basic Life Support Unit (BLS) and Advanced Life Support Unit (ALS). Let p^A and p^B be the number of vehicles of type A and B in use, and r^A and r^B be the coverage standards for each vehicle type, and $W_i^A = \{j \in W : t_{ij} \leq r^A\}$, $W_i^B = \{j \in W : t_{ij} \leq r^B\}$ in which t_{ij} is the time or distance limit for the coverage level corresponding to the vehicle types. These two sets include the possible location sites for the two types of vehicles. Finally, letting x_j^A and x_j^B be the location variables and y_i be the coverage variable if point $i \in V$ is covered by both vehicle types, the TEAM model is developed as shown below:

$$\text{Maximize } z = \sum_{i \in V} d_i y_i$$

$$\text{Subject to: } \sum_{j \in W_i^A} x_j^A \geq y_i \qquad (i \in V)$$

$$\sum_{j \in W_i^B} x_j^B \geq y_i \qquad (i \in V)$$

$$\sum_{j \in W} x_j^A = p^A$$

$$\sum_{j \in W} x_j^B = p^B$$

$$x_j^A \leq x_j^B \qquad (j \in W)$$

$$x_j^A, x_j^B = (0,1) \qquad (j \in W)$$

$$y_i = (0,1) \qquad (i \in V)$$

Actually, this model is the modification of MCLP, and it differs basically from it by the constraint: $x_j^A \leq x_j^B$, which forces the model to create a hierarchy among the two vehicle types. This constraint can be removed according to the conditions, then the model is an extension of MCLP to multiple vehicle types. This extended version (Schilling et al., 1979) is called the FLEET model (facility, location, equipment placement technique). In this model, the hierarchy constraint is removed, but only p location sites can be used. A more advanced model in the same fire fighting area was developed by Marianov and ReVelle (1992). In this model, it is assured that each demand point is covered by an adequate number of pumpers and rescue ladders.

As far as missing calls are concerned with, none of the approaches proposes an appropriate solution. Daskin and Stern (1981) develop a strategy to handle this issue. The strategy is based on MCLP. Rather than increasing the available number of ambulances, the aim is to keep the number as it is and handle the missing calls issue. The solution strategy is based on the idea of multiple coverage. In this approach, a hierarchical objective is used to maximize the number of demand points covered more than once. Similarly, Hogan and ReVelle (1986) use the idea of multiple coverage in their methodology and define their objective as the maximization of the total demand covered twice. Brotcorne et al. (2003) mention in their review paper about the two backup coverage formulations developed by Hogan and ReVelle (1986). The formulations are called as BACOP1 and BACOP2. In these models, a binary variable u_i is incorporated into the model formulations; it is equal to 1 if and only if i is covered twice within the coverage standard, r. The two models are shown below:

(BACOP1)

Maximize $\quad z = \sum_{i \in V} d_i u_i$

Subject to:

$$\sum_{j \in W_i} x_j - u_i \geq 1 \qquad i \in V$$

$$\sum_{j \in W} x_j = p$$

$$0 \leq u_i \leq 1 \qquad i \in V$$

$$x_j \geq 0 \qquad i \in V$$

(BACOP2)

Maximize $\quad z = \theta \sum_{i \in V} d_i u_i + (1-\theta) \sum_{i \in V} d_i y_i$

Subject to:

$$\sum_{j \in W_i} x_j - y_i - u_i \geq 0 \qquad i \in V$$

$$u_i - y_i \leq 0$$

$$\sum_{j \in W} x_j = p$$

$$0 \leq u_i \leq 1 \qquad i \in V$$

$$0 \leq y_i \leq 1 \qquad i \in V$$

$$x_j \geq 0 \qquad i \in W$$

In BACOP2, θ is a weight chosen in [0, 1]. BACOP2 is a model that looks for a balance among demand points covered only once and twice. However, BACOP1 strictly imposes that demand points be covered twice.

In order to solve the problem of missing calls, we see the work by Gendreau et al. (1997) among the most recent approaches. In their work, they aim to maximize the demand covered twice within a time standard of r_1, using p ambulances, at most p_j ambulances at site j. Two coverage standards are used (r_1 and r_2) with $r_1 < r_2$. All demand must be covered by an ambulance located within r_2 time units (broader coverage is guaranteed), and a proportion α of the demand must lie within r_1 time units of an ambulance. The ambulance that covers this α portion could be the same ambulance as the one that covers demand in r_2 time units. Letting $W_i^1 = \{j \in W : t_{ij} \leq r_1\}$ and $W_i^2 = \{j \in W : t_{ij} \leq r_2\}$, the integer variable y_j stands for the number of ambulances located at $j \in W$ and the binary variable x_i^k is equal to 1 if and only if the demand at vertex $i \in V$ is covered k times ($k = 1,2$) within r_1 time units. The formulation below is called the double standard model (DSM).

(DSM)

$$\text{Maximize} \quad z = \sum_{i \in V} d_i x_i^2 \tag{2.1}$$

Subject to: $\sum_{j \in W_i^2} y_j \geq 1 \qquad i \in V \qquad (2.2)$

$$\sum_{j \in V} d_i x_i^1 \geq \alpha \sum_{i \in V} d_i \qquad (2.3)$$

$$\sum_{j \in W_i^1} y_j \geq x_i^1 + x_i^2 \qquad i \in V \qquad (2.4)$$

$$x_i^2 \leq x_i^1 \qquad i \in V \qquad (2.5)$$

$$\sum_{j \in W} y_j = p \qquad (2.6)$$

$$y_j \leq p_j \qquad j \in W \qquad (2.7)$$

$$x_i^2, x_i^1 \in \{0,1\} \qquad i \in V \qquad (2.8)$$

$$y_j, \text{ integer} \qquad j \in W \qquad (2.9)$$

In this context, (2.1) computes the demand covered twice within r_1 time units, constraints (2.2) and (2.3) stand for multiple coverage requirements. The left hand side of (2.4) represents the number of ambulances covering vertex i within r_1 units, while the right hand side is 1 if i is covered within r_1 units and 2 if it is covered at least twice within r_1 units. The combinations of constraints (2.3) and (2.4) ensure that α portion of the demand is covered and the coverage standard must be r_1. Constraint (2.5) ensures that demand point i cannot be covered at least twice if it is not covered at least once. In constraint (2.7), p_j could be set as 2, since the solution always satisfies that condition.

Although the model looks similar to BACOP2, it allows some portion of the demand in the area to be covered again by the same ambulance by the coverage standard approach. DSM enables to construct two level coverage distances or time base rather than using only one type coverage standard r. Gendreau et al. (2001) later extend this model to a dynamic environment and develops the model which is dynamic double standard model (DDSM), and the model is solved by utilizing the tabu search heuristic.

As far as the common properties of these models are concerned with, they are all appropriate for the deterministic and static environment. With the exception of LSCM, the number of vehicles is determined in advance. Some side issues as well can be taken into account as the multiple types of vehicles and extra-coverage. The development of the models is presented below in a chronological order (see Table 1).

Brotcorne et.al (2003) summarizes the deterministic, static and dynamic models as shown in Table 3 (at the end of this chapter).

Table 1 : The chronological development of deterministic ambulance location models

1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982
LSCP			**MCLP**					**TEAM, FLEET**		**Multiple Coverage**	
1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994
			BACOP1 BACOP 2								
1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006
		DSM				**DDSM**					

2.2 Probabilistic and Dynamic Approaches

It is the fact that the cost of not being able to respond to an emergency call sometimes might be immeasurable, since the task, at last, is related with the human life. For this reason, much attention has been given to managing responses to calls in the emergency medical services field. Probabilistic and dynamic methods have been heavily utilized in order to decrease the number of missing calls and increase the performance of ambulance services in this manner.

The maximum expected covering location problem formulation (MEXCLP) by Daskin (1983) is among the first probabilistic models in ambulance location studies.

In this work a "server busy probability" p is assigned to each server (ambulance). The assumption is that this probability is the same for every ambulance. The developed model is as follows:

(MEXCLP)

Maximize $\quad z = \sum_{k=1}^{N}\sum_{j=1}^{M}(1-p)p^{j-1}h_k y_{jk} = \sum_{k=1}^{N}\sum_{j=1}^{M} w_j h_k y_{jk}$

Subject to: $\quad \sum_{j=1}^{M} y_{jk} - \sum_{i=1}^{N} a_{ki}x_i \le 0 \qquad \forall k$

$$\sum_{i=1}^{N} x_i \le M$$

$$x_i = 0,1,\ldots,M \qquad \forall i$$

$$y_{yk} = 0,1,\ldots,M \qquad \forall j,k$$

where

$$y_{jk} = \begin{cases} 1, & \text{if node } k \text{ is covered by at least } j \text{ facilities} \\ 0, & \text{if node } k \text{ is covered by less than } j \text{ facilities} \end{cases}$$

x_i = number of facilities located at node i

$w_j = (1-p)p^{j-1}, \quad j = 1,\ldots,M$

h_k = demand generated at node k

$$a_{ki} = \begin{cases} 0, & \text{if } d_{ki} > D \text{, a facility at } i \text{ does not cover demands at } k \\ 1, & \text{if } d_{ki} \le D \text{, a facility at } i \text{ covers demands at } k \end{cases}$$

M = number of facilities to be located,

N = number of nodes in the network,

D = distance standard for the required coverage,

d_{ki} = distance between facility at i and demand at k.

Briefly, in the model, if vertex $i \in V$ is covered by r ambulances, the expected covered demand is $E_r = h_i(1-p^r)$, and marginal contribution of the r^{th} ambulance to this expected value is $E_r - E_{r-1} = h_i(1-p)p^{r-1}$, and the model tries to maximize their sum. The validity of this model emerges from the fact that the objective function is concave in j, this implies that if $y_{jk} = 1$, then $y_{1k} = 1 = y_{2k} = ... = y_{jk} = 1$; and if $y_{jk} = 0$, then $y_{1k} = 0 = y_{2k} = ... = y_{jk} = 0$.

Daskin (1983) applies the model to a 55-node example by utilizing a single node substitution heuristic. Though the solution procedures yield the same results as MCLP while p approaches 1, it is stated that optimum solution is not guaranteed.

Later the model MEXCLP is revisited by Batta, Dolan and Krishnamurthy (1989). The MEXCLP is criticized in terms of its underlying assumptions such as independent operation of servers, the same busy probability for all servers and for all locations. In order to eliminate these problems, authors apply hypercube queueing model (developed by Larson, 1974) in a single node substitution heuristic optimization procedure. The empirical findings show that the new method (AMEXCLP: adjusted MEXCLP) produces much more reliable results.

An extension of MEXCLP, called TIMEXCLP, is proposed by Repede and Bernardo (1994) and applied to the Louisville-Kentucky data. In TIMEXCLP, variations in travel speed in separate parts of the day are explicitly considered. The method is combined with a simulation module to provide an analysis of the developed solution. The main result is an increase of the proportion of calls covered in 10 minutes or less from 84% to 95%.

Rather than getting interested in busy probabilities of servers, some work also focused on ability of the servers to cover demand with a given probability α. Two probabilistic models are developed by ReVelle and Hogan (1989). These authors formulate the maximum availability location problem (MALP I). The busy fraction

p is assumed to be the same for all potential location sites. The minimum number of ambulances required to serve each demand point k with reliability level α is determined by the constraint:

$1 - p^{\sum_{i \in N} x_i} \geq \alpha$, which can be later linearized as:

$$\sum_{i \in N} x_i \geq \left[\log(1-\alpha)/\log p\right] = b .$$

In order to formulate MALP I, y_{jk} is defined as in MEXCLP, that is:

$$y_{jk} = \begin{cases} 1, & \text{if node } k \text{ is covered by at least } j \text{ facilities} \\ 0, & \text{if node } k \text{ is covered by less than } j \text{ facilities} \end{cases}$$

(MALP I)

Maximize z= $\sum_{k \in N} h_k y_{jk}$

Subject to $\sum_{j=1}^{b} y_{jk} - \sum_{i=1}^{N} a_{ki} x_i \leq 0$

$$\sum_{i=1}^{N} x_i \leq M$$

$$y_{jk} \leq y_{j,k-1} \quad j \in N, \;\; k = 2,...,b$$

$$x_i, y_{jk} \in \{0,1\}$$

Other than the reliability probability, the model also differs from MEXCLP by the constraint $y_{jk} \leq y_{j,k-1}$. It is stated that this constraint is required because of the loss of concavity property observed in MEXCLP. In the second model developed by the same authors, unique busy probability is calculated for every server, and again for every i, a new b_i value is determined. The second model (MALP II) points out the difficulty of working with a busy fraction p_j, specific to each $j \in M$, since these values cannot be known in advance and could be determined correctly after the model output is revealed. Another work on this issue is proposed by Marianov and ReVelle (1994). They propose the queueing probabilistic location set covering problem (QPLSCP) in which busy fractions are site-specific. These authors compute the minimum number of ambulances, b_i, necessary to cover a demand point $i \in V$ in such a way that the probability of all being busy does not exceed a given threshold. This value is then used in MALP II model.

As far as overall reliability revel is concerned with rather than busy probabilities, Ball and Lin (1993) propose an extension of LSCM, called Rel-P. This model incorporates a linear constraint on the number of vehicles required to achieve a given reliability level.

Lastly, it should be noted that all probabilistic models up to now, have tried to resolve the issue of not responding to a call by using especially vehicle specific busy probabilities or ensuring overall system reliability. In addition to these studies, Mandell (1998) describes a two-tiered system in which advanced life support (ALS) and basic life support (BLS) units are available. This two-tiered model (TTM) aims to maximize the expected covered demand by using two vehicle types, one of which has superiority over the other. That is to say, BLS units could also be replaced by ALS units while reverse is not possible.

The development of the probabilistic models in chronological order is shown below in Table 2. The detailed tabulated presentation is in Table 4 (at the end of this chapter).

Table 2 : The chronological development of probabilistic ambulance location models

1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994
MEXCLP						**MALP I** **MALP II** **AMEXCLP**				**REL-P**	**TIMEXCLP** **QPLSCP**
1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006
			TTM								

2.3 Recent Studies in Locating Emergency Services

The detailed models developed so far constitute the major frameworks in locating emergency services. Most of the recent works are based on those well known studies. For instance, Galvao et al. (2001) examine MEXCLP, AMEXCLP and hypercube queueing model (HQM) in terms of their differences and conclude that these three models are not strictly comparable because of their different objectives. Additionally, the work confirms the fact that AMEXCLP produces improved results compared to MEXCLP by relaxing the assumption that the servers operate independently. On the other hand, hypercube location model (HLM) by relaxing other two assumptions of MEXCLP (about server busy probabilities and busy probabilities according to the location) and dealing with queued calls as well, opposed to AMEXCLP, is capable of producing much more reliable results.

Later, Saydam et al. (2002) propose a genetic algorithm (GA) in order to reduce the considerable over- or under-estimated margins among optimal locations found by AMEXCLP and MEXCLP.

Ingolfsson et al. (2003) extend their studies in order to capture the random delays and travel times. In their approach, they address a new side issue that has not been studied extensively. The time period that is spent for the preparation of the medical crew to respond to a call and the other delays in travel times are considered separately in this study. The underlying motivation is such that a node assigned as covered may not be covered when possible delays are considered. A branch and bound algorithm working with a nonlinear programming algorithm is proposed, and the model is run for the city of Edmonton.

Lightner et al. (2004) incorporate the extension of FLEET model (MOFLEET: multiple coverage, one unit FLEET problem) in order to solve locating EMS vehicles in Fayetteville, NC. The study shows that within the existing sites it is possible to increase the coverage by redistributing the vehicles; and when the possible site alternatives are on hand, the coverage increases sharply. The study includes a real-life application.

Peleg et al.(2004) utilize the tool, geographic information system (GIS), in order to reduce ambulance response time.

Galvao et al. (2005) propose a unified version of Daskin's maximum expected covering location problem (MEXCLP) and ReVelle and Hogan's maximum availability location problem (MALP) and by relating these two approaches, they develop extensions to those approaches mainly by incorporating Larson's hypercube model and simulated annealing.

Rajagopalan et al. (2006) analyze the performance of four meta-heuristics applied to a probabilistic location model. The findings show that the tabu search and simulated annealing approaches find the best solutions in the least amount of time. Morabito et al. (2007) extend the hypercube model in order to handle the location and allocation of non-homogeneous vehicle types.

2.4 The Models with Redeployment

The crucial phase in detecting ambulance locations begin when the redeployment and relocation needs arise. Determining the initial position of the vehicle is the core crucial problem that has been studied widely.

Among the works that have been analyzed so far, the population or so-called demand points have been considered stable. However, it is a fact that the population dispersion changes at different time zones of the day. While the population is dispersed in city centers at noon time, it is scattered around the countryside and far from the city centers during night time. So, keeping the position of ambulances always the same is not an efficient policy when the dynamic nature of the population is taken into consideration. Then determining the new settlement of ambulances and the assignment of ambulances to their new places become the issue.

There are several works which have addressed the relocation issue. As mentioned in the review by Richards (2007), the first emergency redeployment problem was proposed by Kolesar and Walker in 1974. They analyzed the relocation of fire companies in the following case: when a fire unit leaves its initial place, the decreased coverage level in that area is compensated for by the other units placed in its “response neighborhood”. The objective in Kolesar and Walker (1974) is to minimize the assignment costs to the new locations by minimizing travel times on account of the pre-defined “good coverage” standard. The work follows four-step sequential solution approaches. With the help of response neighborhoods the availability of relocation is determined, the empty fire fighting houses that are going to be filled are determined. The vehicles are chosen and lastly the assignment problem is solved. The first three stages are handled by heuristic approaches while the last stage is solved as an assignment problem.

Gendreau et al. (2001) enhance the scope of their study conducted in Gendreau et al. (1997) in order to analyze the relocation problem. The model is based on the idea of maximizing the total demand covered at least twice within r_1 dynamically by introducing a penalty term for relocation alternatives. As being different from the DSM (double standard method) approach, DDSM (dynamic double standard model) is solvable in real time by the contribution of parallel computing and tabu search heuristic. The DDSM is proposed as follows:

(DDSM)

$$\max \sum_{i \in V} d_i x_i^2 - \sum_{j=1}^{m} \sum_{l=1}^{p} M_{jl}^t y_{jl}$$

Subject to:

$$\sum_{j=1}^{m} \sum_{l=1}^{p} \delta_{ij} y_{jl} \geq 1 \quad \forall v_i \in V$$ (absolute covering constraint)

$$\sum_{i=1}^{n} \lambda_i x_i^1 \geq a \sum_{i=1}^{n} \lambda_i$$ (proportion a of all demand is covered)

$$\sum_{j=1}^{m} \sum_{l=1}^{p} \gamma_{ij} y_{jl} \geq x_i^1 + x_i^2 \quad \forall v_i \in V$$ (number of ambulances located within r_1 units should be at least 1 if $x_i^1 = 1$ or at least two if $x_i^2 = x_i^1 = 1$)

$$x_i^2 \leq x_i^1 \quad \forall v_i \in V$$ (Demand point can not be covered twice if it is not covered at least once)

$$\sum_{j=1}^{m} y_{jl} = 1 \quad l = 1, \ldots, p$$ (each ambulance is assigned to a possible location site)

$$\sum_{l=1}^{p} y_{jl} \leq p_j \quad \forall v_i \in W$$ (an upper bound for the number of vehicles waiting at a location site)

$$x_i^1 = 0 \text{ or } 1 \quad \forall v_i \in V$$

$$x_i^2 = 0 \text{ or } 1 \quad \forall v_i \in V$$

$$y_{jl} = 0 \text{ or } 1 \ \forall v_i \in W \text{ and } l = 1,...,p$$

In the model the composition of the M^t_{jl} is crucial**.** It is defined as the cost coefficient associated with the relocation of ambulance $l = 1,...,p$ from its current site at time t to location site $v_j \in W$. M^t_{jl} coefficients are updated at each period and penalize redeploying the same ambulance repeatedly, avoiding round trips between two location sites and preventing the large-drive times between initial location and final destination. The other detailed explanation for the variables and cost coefficients are mentioned in the part related with DSM above.

Later Gendreau et al. (2003) examine the relocation issue in the context of maximal expected coverage relocation problem (MECRP). The model concerns the dynamic problem from somehow the static perspective and solves the relocation problems between different time zones by imposing constraints upon the number of redeployments. MECRP assumes the redeployment time to be zero. With this assumption, there are no repositioning costs and ambulances are placed to serve the next call in the best way.

Andersson et al. (2006) propose a model DYNAROC, which solves the dynamic relocation problem in real time. By the concept of "preparedness", DYNAROC detects the regions where there is a decrease in the level of emergency case preparedness and relocate ambulances accordingly. In this sense the model is similar to the one developed by Kolesar et al. (1974). The model uses tree search heuristic.

Lastly, Saydam et al. (2006) propose a comprehensive approach in order to handle the multiperiod set covering issue and dynamic redeployment of ambulances. In their study, the changing nature of the demand patterns in different time zones is considered and the model is enhanced by calculating server specific busy probabilities that are incorporated into a simulated environment. The model seeks to provide a reliability level at each time zone as similar to the case of preparedness.

The detailed tabulated presentation for the model that considers relocation is in Table 5 (at the end of this chapter).

The review by Goldberg (2004) is also a valuable source of information in terms of modelling efforts for the emergency services location and allocation. Goldberg classifies the works according to the dates of studies and modelling approaches. As a result of compact collection of analysis, Goldberg concludes that real time vehicle routing, scheduling and crew shifting issues deserve to be focused intensively as future work. The work in this study aims to focus on the following issues:

- Multi-level coverage perspective
- Rapid solution procedures
- Changing patterns of demand during a certain period of time, and hence the shift of ambulances from one site to another
- Comparison between multi-level and multiple coverage ideas and the interaction among them.

2.5 Issues Considered in Location of Emergency Service Vehicles

Both in deterministic and probabilistic models, there are some common issues and questions. All these issues that have to be addressed while studying EMS are summarized below.

- o How many servers (ambulances) are needed?
- o How long can people involved in an emergency case afford to wait for service before the consequences of a lack of response become intolerable?
- o What does ‘coverage’ or ‘good quality coverage’ mean?
- o What is to be done when servers are not available?
- o How is the distribution of workload among servers?
- o In real life situations:
    - The issue of data collection for the right modeling is crucial.

- The analysis must take into account the nature of the emergency service as public or private.
- Political or real-life feasibility of locating new services or closing the existing ones is another factor while specifying the location sites.
- Handling multiple objectives in decision making always force to produce alternative solutions.

The budget allocated for the emergency services directly affects the number of ambulances available. For this reason, covering the population as much as possible with the existing ambulances is among the first motivations. Though emergency cases vary, being able to respond quickly and on time is another highly valued objective; if this objective is not met, the result might not be endurable.

All in all, it is appropriate to classify the major works on ambulance location and allocation in terms of the following characteristics:

I. Objective
   A. Statement
      i. Covering each demand point
      ii. Maximal coverage
   B. Approach
      i. Single Objective
      ii. Multiple Objective
II. Type of Vehicles
   A. One type
   B. Two types (BLS, ALS)
III. Nature of Demand
   A. Deterministic
   B. Stochastic
   C. Dynamic
IV. Hierarchical Level

    A. Single Stage (Location)
    B. Multiple Stage (Location, Routing)

V. Planning Horizon
    A. Single Period (Static)
    B. Multiple Period (Dynamic)

VI. Types of the Model
    A. Hypothetical
    B. Real-Word Data

2.6 The Link between EMS and Disaster Operations Management (DOM)

The utilization of OR methods and mathematical tools in the field of EMS location goes back to 70s as mentioned in the previous sections. The further enhancement of the methods and tools are also seen in the management of disasters and large scale emergencies. However, the developments in large scale emergencies are much more recent. The work conducted by Altay et al. (2006) covers a wide range of literature on DOM. Emergency management is composed of four phases called as mitigation, preparedness, response and recovery. The 109 articles examined are placed in one of these categories. Altay et al. (2006) also present the statistical summary for the articles' details such as their publishing origin (US or international), publishing time, methodology, disaster type examined and research contribution (theory, model, application). Another review work is conducted by Denizel et al. (2003) who classify the literature by the properties of articles based on mainly the problem setting, data, novelty of the problem, solution approach, generalization of results and future research implications.

The most appropriate phases in which OR tools could be used in large scale emergencies are "preparedness" and "response" stages. The fact is also confirmed by the review works conducted by Altay et al. (2006) and Denizel et al. (2003). In other two stages, that is, mitigation and recovery, rather than mathematical tools, action plans and handbooks are mostly published concerning the planning and coordination of governmental and civil institutions.

The intersection between emergency services location and large scale emergencies lies in the “preparedness” stage. It is the fact that large scale emergencies have their unique features different from daily emergencies. However, the units required to be placed in order to respond to large scale emergencies share the same motivation with daily life emergencies, that is, quick response.

The studies in the context of large scale emergencies focus more on locating facilities that are to be operable for rescuing operations and stocking as well as providing first-aid kit packages through logistic networks. Belardo (1984) is among the initial works that focuses on partial covering approach to locating response resources for major maritime oil spills.

Though Drezner (1987) does not directly consider the location problem in disaster cases, his work on two location problems (*p*-median and *p*-center) analyzing the unreliable facility case is placed in large scale emergencies literature. By considering that facilities may become unavailable in disaster situations, the solution methods developed by the study is applicable for large scale emergencies. Drezner develops heuristic solutions for the problem and presents the results.

Knott (1987) considers the last-mile delivery of food items from a distribution centre to a number of refugee camps, assuming a single mode of transportation that makes direct deliveries to camps.

One of the major studies in large scale emergency case is Batta et al. (1990). The study concerns the location of ambulances in an environment in which a large demand volume often leads to the unavailability of the most desirable response unit. By utilizing the idea of backup coverage and applying the set covering and maximal covering location models, Batta et al. (1990) propose a solution methodology to minimize the congestion in ambulance calls when there is a large volume of demand in large scale emergencies.

Related with DOM, Current et al. (1992) propose solutions for locating emergency warning sirens. During late 90s, the studies mostly focus on simulation models for large scale emergency evacuation activities. Pidd (1996) and Yamada (1996) focus on a city's emergency evacuation planning through simulation models. A network flow approach is used in Yamada's (1996) study. In addition to those studies, Gregory and Midgley (2000) propose a multi-agency planning and coordination framework for disaster operations.

In parallel with the location concepts, we see relief logistics studies beginning from late 90s. Haghani and Oh (1996) and Oh and Haghani (1996) provide detailed routing and scheduling plans for more than one transportation mode facilitating the flow of multiple commodities from multiple supply points in a disaster relief operation. The authors assume that the quantities of commodities are known. They formulate a multi-commodity, multi-modal network flow problem with time windows as a large scale MIP model on a time space network with the objective of minimizing accumulated aggregate cost in the whole set up.

Özdamar et al. (2004) work on another similar problem situation and address an emergency logistics problem for distributing multiple commodities from a number of supply centers to distribution centers next to the affected areas. They formulate a multi-period multi-commodity network flow model to determine pick-up and delivery schedules for vehicles as well as the quantities of loads delivered on these routes, with the objective of minimizing the amount of unsatisfied demand over time. The proposed model allows for regenerating plans based on changing demand, supply quantities, and fleet size. Later Balçık and Beamon (2008) consider both location and stocking issues sequentially, for a humanitarian relief chain responding to quick-onset disasters. In particular, they develop a model that determines the number and location of distribution centers in a relief network and amount of relief supplies to be stocked at each distribution center to meet the needs of people affected by disasters.

Table 3 : Summary of the deterministic models

Reference	Model	Objective	Coverage constraints	Constraints on location sites	Ambulances
ReVelle, Toregas, Bergman (1971)	LSCM	Minimize # of ambulances	Cover each demand point at least once	At most one ambulance per site	One type, number unlimited
Church and ReVelle(1974)	MCLP	Maximize the demand covered	None	At most one ambulance per site	One type, number given
Schilling et al. (1979)	TEAM	Maximize the demand covered	None	At most one ambulance of each type per site. Type A can only be located if Type B is located	Two types, number given
Schilling et al. (1979)	FLEET	Maximize the demand covered	None	At most one ambulance per site. Only p sites can be used	Two types, number given
Daskin and Stern (1981)	Modified MCLP	Maximize the demand covered, then the number of demand points covered more than once	Cover each demand point at least once	At most one ambulance per site	One type, number given
Hogan and ReVelle (1986)	Modified MCLP (BACOP1 and BACOP2)	Maximize the demand covered twice, or combination of the demand covered once or twice	Cover each demand point at least once	At most one ambulance per site	One type, number given
Gendreau et al. (1997)	DSM	Maximize the demand covered at least twice within r_1	All demand covered within r_2 . Proportion α of all demand covered within r_1	Upper bound on the number of ambulances per site	One type, number given
Gendreau et al. (2001)	DDSM	Dynamically maximize the demand covered at least twice within r_1 , minus a redeployment penalty term	All demand covered within r_2 . Proportion α of all demand covered within r_1	Upper bound on the number of ambulances per site	One type, number given

Table 4 : Summary of the probabilistic models

Reference	Model	Objective	Coverage constraints	Constraints on location sites	Ambulances	Busy Period
Daskin (1983)	MEXCLP	Maximize the expected demand covered	None	None	One type.Upper bound given, (always reached)	Same for each ambulance, given
ReVelle and Hogan (1989)	MALP I	Maximize the total demand covered with a probability α	None	None	One type.Number given	Same for all potential sites
ReVelle and Hogan (1989)	MALP II	Maximize the total demand covered with a probability at least α	None	None	One type.Number given	Varies according to each demand point
Batta et al. (1989)	AMEXCLP	Maximize the expected demand covered	None	None	One type.Number given	Varies according to each demand point
Ball and Lin (1993)	Modified LSCM (Rel-p)	Minimize the sum of ambulance fix costs	Proportion a of all demand covered within r_1	At most p_j ambulances at site j	One type.Number given	Upper bound computed on busy period
Repede and Bernardo (1994)	TIMEXCLP	Maximize the expected demand covered	None	None	One type.Number given.Varying speeds	Same for each ambulance, given
Marianov and ReVelle (1994)	QPLSCP	Maximize the total demand covered with a probability at least α	None	None	One type.Lower bound computed for each demand point	Varies according to demand points
Mandell (1998)	TTM	Maximize the expected demand	None	Bounds on each type per site	Two Types	Computed using a queueing model

Table 5 : Summary of the studies on relocation

Reference	Model	Objective	Coverage constraints	Solution Method	Used data	Relocation Constraints
Kolesar and Walker (1974)	Relocation of Fire Companies	Four step Sequential Objectives 1-Decide for relocation (with the help of response neighborhoods) 2-Determine empty houses to be filled. 3-Determining available companies to relocate 4-Solve the assignment targeting minimizing travel times.	Standard minimal coverage constraint, targeting a "good" level.	Heuristic Methods for the first three stages, and assignment algorithm for the last stage.	Hypothetical and Real data.	-Long trips not allowed -too busy units not allowed -units covering big areas not allowed.
Gendreau et al (2001)	Redeployment problem (RP)	-Location Ambulances with respect to double coverage standard -relocate them dynamically to maximize the covered area	Double coverage standard	Parallel tabu search	Hypothetical and Real data.	- too busy units not allowed -round trips not allowed -Long trips not allowed
Andersson (2006)	DYNAROC	Dynamically providing a minimum preparedness level (P_m)	Maximizing the demand in which the minimum preparedness level is met	The tree search algorithm	Hypothetical data	-repeated trips not allowed.
Rajagopalan and Saydam (2008)	DACL	Determine the minimum number of ambulances and their locations for each time cluster	Coverage requirement with predetermined reliability	Reactive Tabu Search and Hybercube Algorithm	Hypothetical data	No constraint

CHAPTER 3

DEFINITION AND MATHEMATICAL MODELING OF THE AMBULANCE LOCATION PROBLEM FOR THREE-LEVEL COVERAGE

In this chapter, starting with the drawbacks of the former formulations of the ambulance location problems like LSCP, MCLP, we define our problem formulation with its environmental characteristics and assumptions. Finally we present the mathematical model of our formulation.

3.1 Covering Problems in General

"The location set covering problem" (LSCP) and "maximal covering location problem" (MCLP) form the foundation in our study. For this reason it is appropriate to elaborate on the characteristics of these two models. In LSCP:

(i) There is no distinction among nodes based on demand size. Each node must be covered based on the specified distance criterion regardless of its demand size.

(ii) If the coverage restriction is tight, this leads to a large number of facilities to be located.

(iii) When an outlying node has a small demand, the cost/benefit ratio of covering this node can be extremely high.

On the other hand, LSCP enables us to determine the minimum number of available resources to ensure a desired service level for every individual or demand area.

The limited resources are taken into consideration in the maximal covering location problem (MCLP), since it seeks to maximize the covered demand in the specified distance standard by the given number of resources. The first attempt to combine LSCP and MCLP was achieved by Revelle et al. (1974) in which the idea of "the maximal covering with mandatory closeness constraint" was proposed, and the location problem was redefined as:

Locate p facilities at possible sites on the network to maximize the population that can be covered within a given service distance S while at the same time ensuring that the users at each point of demand will find a facility no more than T distance $(T > S)$ *away.*

Although the incorporation of mandatory closeness in the formulation improves the coverage quality, there are circumstances where the provision of a service needs more than one "covering" facility (Daskin, Hogan, and ReVelle 1988). This occurs when resources are not available. For example, assume that ambulances are being located at dispatching points in order to serve demand across an area. If the closest ambulance is busy when a call is received, then the next closest available ambulance is assigned to the call. If the closest available ambulance is farther than the service standard, then the call for service is not provided a service within the coverage standard. To handle such issues, models have been developed that seek multiple coverage in the form of stochastic and deterministic formulations. A good example of a stochastic multiple coverage model is the maximal expected coverage model of Daskin (1983).

The back up covering model of Hogan and ReVelle (1986) is a good example of a deterministic coverage model that involves maximizing the second-level coverage. However, in this model it is observed that when the multiple coverage is targeted, the optimal locations are aggregated at the center of the network.

The idea of multiple coverage, at first glance, helps us to think that all demand points are covered in a way that no missing calls exist in the system and quick response objective is handled. However, as it is observed in the work of Hogan and Revelle (1986), collecting the vehicles in a centralized service station may lead to failures in answering an urgent call requiring an immediate access.

In this context, it seems much more appropriate to provide services as close as possible to the demand points, while not failing to cover each demand point in the broader sense. This approach seems to converge to the idea of "the maximal covering with mandatory closeness constraint" approach. However, by defining multi-level coverage quality, the approach is much more a maximal covering location model in the presence of a broader coverage. That is to say, rather than accomplishing only one-level broad coverage, the aim is to maximize the closest coverage that leads to step-by-step broader coverage levels. By this method demand points as many as possible can be covered in terms of tight coverage constraint that leads to the extended step-by-step broader coverage.

We formulate the problem as a multi-level maximal covering location problem the characteristics of which can be stated as follows.

3.2 Characteristics of the Three Level Maximal Covering Location Problem

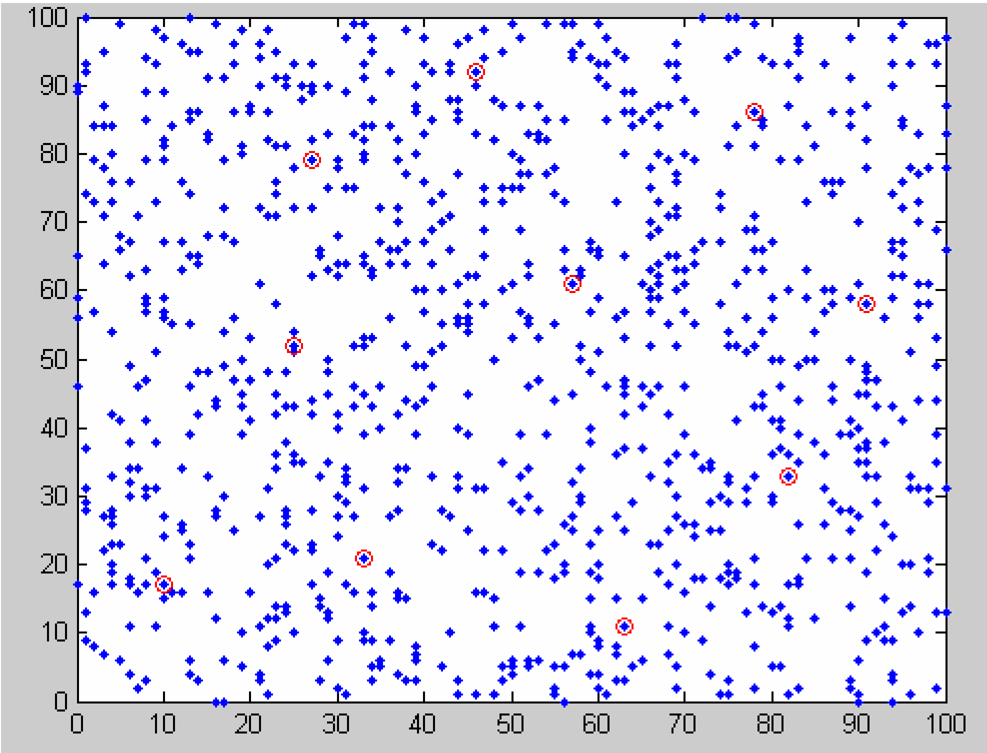
- As the objective function, maximal first level coverage is targeted while aiming at accomplishing to maximize the second and the third degree coverage. However, the problem reduces to a set covering location problem if the mandatory closeness constraint is set as a must.
- Actually multi-level maximal covering location problem could be classified as both single and multiple objective type problems. Since we assign weights for every level of coverage, the weights can be changed, and accordingly the objective function can be changed from favoring closeness to favoring being distant from the demand points level by level. The problem can be reduced to a single objective maximal covering location problem by giving no weights for the second and the third degree coverage.
- Type of resource (ambulance) is single. Number of ambulances is limited.

- ♦ Our model is a single stage location model, since it does not address any allocation or routing problem. Actually, there is an embedded allocation phase in the problem; however, in ambulance location problem, there is no specific allocation of customers to ambulances. Each ambulance is responsible to answer a call if it is in one of the coverage ranges of the demand point in practice.
- ♦ The model's planning horizon is a single period, but the problem can be solved repeatedly according to the changing demand conditions. However, the dispersion of the population must be known for the time periods.
- ♦ The nature of demand is deterministic but dynamic. Both demand points and the ambulances are not static in real life.

The population concentration may shift from the city centers at noon time to the countryside at night time. Also, the ambulances are mostly on duty because of the continuous calls. So, by having a quick and reliable model it is easy to perform repeated runs according to known or anticipated demand and ambulance dispersion patterns. For example, if we know where the population is going to accumulate during the weekends or in a social meeting or in a sports competition by collecting historical data, we can run our model again and again and decide on the new relocation plans. With the developed advanced GIS technology, real time ambulance locations are easy to detect. To sum up, the repeated runs for the new data can find the solutions for the ambulances according to the same multi-level maximal coverage objective.

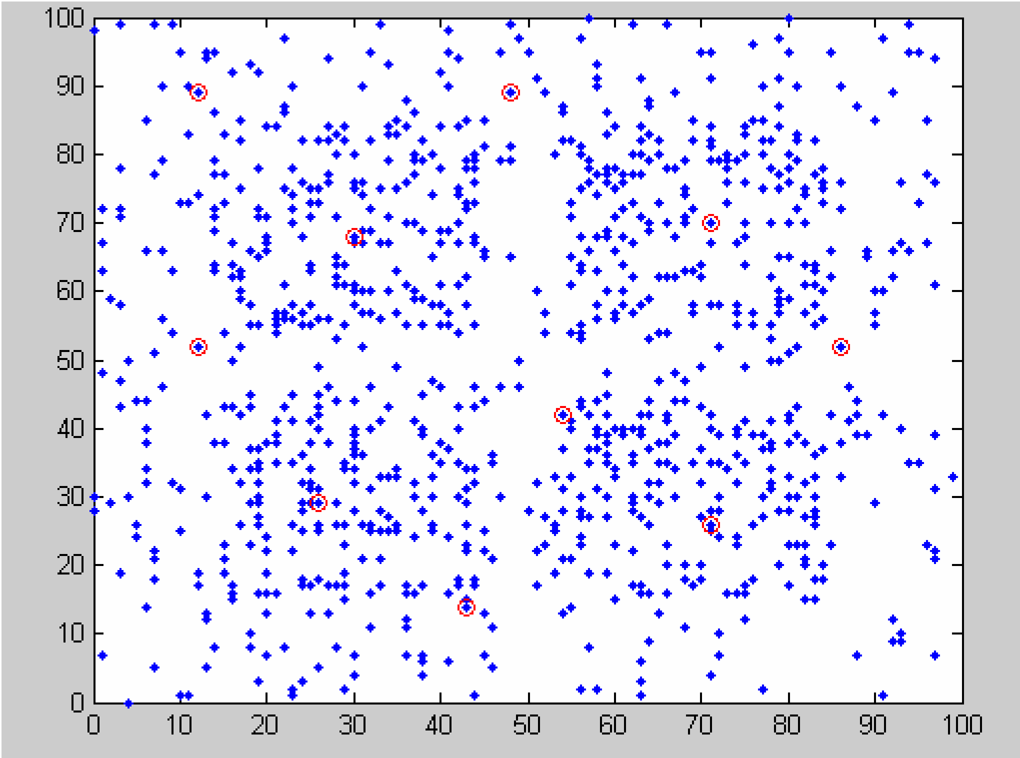
Here, the difficulty lies in the assignment of the cost coefficients to the ambulances in shifting them from their previous locations to the newly found locations. Although it seems to be a trivial assignment problem in terms of distances, that is, an ambulance is assigned to its nearest new location, it is not always the optimal solution to assign the ambulance that is nearest to the new place. An ambulance might be required to undergo several logistics operations before being assigned to its new place, such as renewal of medicine or oxygen tube stocks or gasoline replenishment. So, after finding out assignment cost coefficients and the new places of ambulances, it is an assignment problem that is to be solved by the commercially available solvers. An example node dispersion is presented below in Figure 1 in order to show how demand points could scatter in a different way in different time periods.

As far as recent studies are concerned with, there are two works that could be seen as inspiring studies for us to work on this multi-level maximal covering location problem. One of them is by Church et al. (2003) named as "multi-level location set covering problem" ML-LSCP. In their work they explain the fact that the ML-LSCP spans the LSCP, which is an applied form of the minimum cardinality set covering problem, and adds that since the minimum cardinality set covering problem is NP-hard, the ML-LSCP is also NP-hard. This means, essentially, that some problem instances are not solvable optimally within a reasonable computational time. Hence, it may be necessary to depend on heuristics to solve large or difficult-to-solve problems.



only a demand point   an ambulance location site

Figure 1.a : The dispersion in the noon time and hypothetical 10 ambulance locations.



only a demand point   an ambulance location site

Figure 1.b : The dispersion in the night time and hypothetical 10 ambulance locations.

Toregas and ReVelle (1973) had found that a specific instance of a LSCP may yield to a reduction algorithm, which could possibly reduce a problem instance in terms of both rows and columns. The work of Church et al. (2003) focuses on these reduction algorithms. By the reduction algorithm, they claim that the work of heuristics can be much easier with less data to work with.

Another study in this context is the one performed by Karasakal et al. (2004). In their work, they aim to detect the performance of the maximal covering location model in the presence of partial coverage. Basically two level coverage standards are presented. The points which lie in the coverage of first level are accepted as fully covered, whereas the points which lie outside of the first level are assumed to be partially covered. They conclude that the maximal covering location problem with partial coverage has a substantial effect on the optimal solution in which the partial coverage is not taken into account.

3.3 Mathematical Model for the Three Level Maximal Covering Location Problem

The mathematical model we develop can be considered a hybrid of the models developed by Karasakal et al. (2004) and Hogan and Revelle's (1986) back up coverage location problem. However, we extend the coverage idea to three levels by incorporating the "weighted linear-stepwise partial coverage" into the objective function.

By using the three-level coverage idea, we let the model first maximize the demand points that are covered by the nearest coverage possible solutions. At the same time, while targeting at the nearest possible solutions, the model tries to maximize the number of demand points uncovered in the nearest possible solution, by covering them in the best manner at the second and the third level coverages. In general, we can define our multi-level coverage approach as follows:

At time period t, given the demand points with their weights, possible location sites are determined. Possible location sites could either be the same as the demand points or could be determined exogenously. The coverage standards, r_1, r_2 and r_3, are defined and the number of ambulances is given. The first level coverage standard r_1 is the shortest and most favored coverage level; it corresponds to the highly critical emergency cases. The first level coverage standard r_1 is followed by the less critical coverage levels r_2 and r_3 in order. At first, the initial coverage r_1 is favored because of the highest weight given to the first level coverage, then the second level coverage and the third level coverages are favored in sequence with the objective of maximizing the weighted total demand covered. Prior to the model formulation, the determination of the coverage levels, the minimum number of ambulances that satisfy at least the most relaxed (r_3) coverage standard, and the coverage capacity constraint for an ambulance are among the crucial issues. The restrictions imposed on the coverage standards could create different types of location problems.

We present our integer programming model in the following sections:

3.3.1 Assumptions, Sets, Parameters, Decision Variables

Assumptions

1. A demand point is weighted in proportion to its demand volume, i.e., population size.
2. Euclidean distance is assumed, rather than the road network.
3. Though ambulances could answer the calls while they are out of the stations, they do not patrol in the streets if a call is not assigned to them. Therefore, there is always a need to define specific ambulance location sites for different time periods.
4. An ambulance site can be the same as a demand point.
5. Only one type of ambulance is used.
6. More than one ambulance can be placed at a possible location site.
7. Rather than set covering the model aims at maximal covering. If a demand point can not be covered even within the defined largest coverage range r_3, then either a coverage standard is relaxed or the available number of ambulances can be increased.

Sets

I set of demand points $i = 1,...,n$

J set of potential ambulance sites $j = 1,...,n$, $I \cap J \neq \phi$

d_{ij} distance between the demand point i and possible ambulance location site j

M_{i1} set of potential ambulance sites that cover demand point i such that $d_{ij} \leq r_1$

M_{i2} set of potential ambulance sites that cover demand point i such that $d_{ij} \leq r_2$

M_{i3} set of potential ambulance sites that cover demand point i such that $d_{ij} \leq r_3$

Here it is worth to note that $r_1 < r_2 < r_3$. This definition implies that $M_{i1} \subset M_{i2} \subset M_{i3}$. The graphical representation of the three-level coverage is shown below in Figure 2.

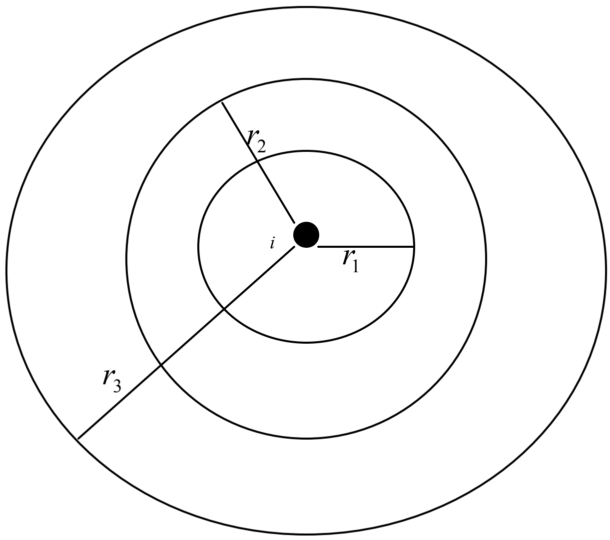


Figure 2 : The coverage circles representing the three coverage levels

Parameters

λ_i demand at point i.

w_k associated weight with the desired coverage level k (k=1, 2, 3); $w_1 > w_2 > w_3$.

d_{ij} distance between demand point i and ambulance site j.

p number of available ambulances.

Decision Variables

$$x_{ij}^1 = \begin{cases} 1 & \text{if } d_{ij} \leq r_1 \text{ (the ambulance at site } j \text{ covers the demand point } i \text{ within } r_1\text{)} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij}^2 = \begin{cases} 1 & \text{if } d_{ij} \le r_2 \text{ (the ambulance at site } j \text{ covers the demand point } i \text{ within } r_2) \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij}^3 = \begin{cases} 1 & \text{if } d_{ij} \le r_3 \text{ (the ambulance at site } j \text{ covers the demand point } i \text{ within } r_3) \\ 0 & \text{otherwise} \end{cases}$$

$y_j =$ number of ambulances located at node j

3.3.2 The Mathematical Model

(Three Level Coverage)

$$\max\ f = \sum_{i \in I} (w_1 \sum_{j \in M_{i1}} \lambda_i x_{ij}^1 + w_2 \sum_{j \in M_{i2}} \lambda_i x_{ij}^2 + w_3 \sum_{j \in M_{i3}} \lambda_i x_{ij}^3) \qquad (3.1)$$

Subject to:

$$\sum_{j=1}^{n} y_j = p \qquad (3.2)$$

$$x_{ij}^1 \le y_j \qquad \forall i \in I,\ j \in M_{i1} \qquad (3.3)$$

$$x_{ij}^2 \le y_j \qquad \forall i \in I,\ j \in M_{i2} \qquad (3.4)$$

$$x_{ij}^3 \le y_j \qquad \forall i \in I,\ j \in M_{i3} \qquad (3.5)$$

$$x_{ij}^1 \le x_{ij}^2 \qquad \forall\ i, j \qquad (3.6)$$

$$x_{ij}^2 \le x_{ij}^3 \qquad \forall\ i, j \qquad (3.7)$$

$$\sum_{j = M_{i1}} x_{ij}^1 \le 1 \qquad \forall i \in I \qquad (3.8)$$

$$\sum_{j = M_{i2}} x_{ij}^2 \le 1 \qquad \forall i \in I \qquad (3.9)$$

$$\sum_{j = M_{i3}} x_{ij}^3 \le 1 \qquad \forall i \in I \qquad (3.10)$$

$$x_{ij}^1 = 0 \text{ or } 1 \qquad \forall i, j \qquad (3.11)$$

$$x_{ij}^2 = 0 \text{ or } 1 \qquad \forall i, j \tag{3.12}$$

$$x_{ij}^3 = 0 \text{ or } 1 \qquad \forall i, j \tag{3.13}$$

$$y_j \geq 0, \text{integer} \qquad \forall j \in J \tag{3.14}$$

The objective function (3.1) maximizes the total weighted demand covered in the first-level, second-level, and third-level coverages. Constraint (3.2) satisfies that the total number of ambulances sited is equal to *p*. Constraints (3.3), (3.4) and (3.5) control the x_{ij}^k ($k = 1,2,3$) variables so that if an ambulance is not sited at *j*, all x_{ij}^k 's for site *j* are forced to be zero. The constraints (3.6) and (3.7) are the consequences of the coverage level definition, $r_1 < r_2 < r_3$; so, if a demand point is covered in r_1, it is also automatically covered in r_2 and r_3; the same relationship holds between r_2 and r_3. Constraints (3.8), (3.9) and (3.10) have a similar function such that the demand point is taken as covered by at most one of the ambulance locations sited around itself for every coverage level. If there are more than one ambulances sited that cover the demand point, only the ambulance that can provide the maximum contribution to the objective function is selected. Constraints (3.11), (3.12), (3.13) and (3.14) are the integrality constraints for the decision variables. Figure 3 is a helpful representation of what the objective function aims at by the three-level coverage idea.

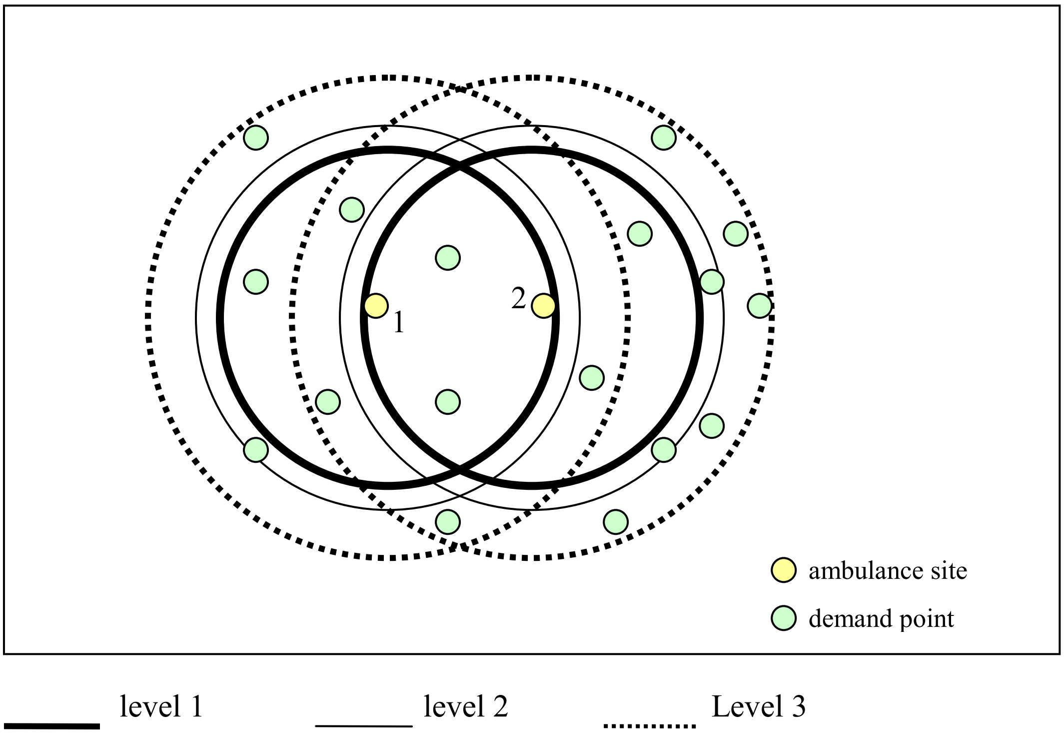


Figure 3 : The graphical representation of three-level coverage idea

In the example in Figure 3, if we take the weights for each corresponding coverage level as $w_1 = 2$, $w_2 = 1$ and $w_3 = 0.5$, then for the two possible location sites, the objective function values are f_1=20.5 and f_2=21, respectively. In this solution, while site 1 covers 5, 6 and 9 demand points in r_1, r_2 and r_3 distance units, respectively, site 2 covers 4, 6 and 14 demand points in r_1, r_2 and r_3 distance units, respectively. If we had solved this problem only for one or two-level coverages, the optimal solution would have been site 1, however, as it is observed in the figure, with three-level coverage approach, site 2 stands as the optimum. Here, on the other hand, it should be noted that a change in the relative weights of the coverage levels could change the optimal solution in the example. In our study, since we would be favoring the closest coverage more, the weight given to the demand covered within r_1 units of distance is the highest among all the coverage level weights ($w_1 > w_2 > w_3$).

3.4 Problem Complexity and Model Validation

The NP-hard nature of the *p*-median, set covering and maximal covering location problems are a well known fact and has been studied extensively. One of the studies is the one performed by Garey and Johnson (1979). For example, if we have an *N*-node network and possible *P* facilities to site, the number of possible location configurations amount to:

$$\binom{N}{P} = \frac{N!}{P!(N-P)!}$$

For a 1000-node network with 10 location sites, the number becomes 2.6341E+23, a-non traceable number to write. Since a one-level coverage problem is NP-hard, the three-level coverage is also NP-hard. Also, when there is no restriction in terms of the number of ambulances located at a site, the possible number of combinations raises to N^P, which makes the solution even harder.

Fortunately, with the advanced computer and software technology, moderate sized NP-hard and NP-complete problems are solvable to optimality. For this reason we code our exact model in Genetic Algebraic Modeling System (GAMS) which is capable of solving both IP (integer programming) and MIP (mixed integer programming) models. We utilize the CPLEX solver embedded in GAMS environment and produce results for moderate sized problems. As it is expected, when the problem size exceeds 1000 nodes, we start to cope with computer memory restrictions.

The code of the three-level maximal covering location model in GAMS environment is given in Appendix A. The GAMS model has been used for two purposes: (i) to validate our model and (ii) to obtain an optimal or near-optimal benchmark solutions to compare the performances of the heuristics we developed for large size three-level coverage problems. The validity of the GAMS model was tested for small-sized problem instances with 10-20 nodes and 2-3 ambulances for which the optimum solutions were known in advance.

The following chapter presents the heuristic approaches we develop for the large size three-level maximal coverage location problem.

CHAPTER 4

THE SOLUTION APPROACHES PROPOSED

Three solution approaches are available in the literature for the location problems. One is the complete enumeration, which is easy to implement when the problem size is small. Since every possible solution is traced in the complete enumeration, the computational burden is high. The other solution approach is mathematical programming. Though the mathematical models guarantee the optimum solutions, even with the advanced general-purpose mathematical solvers and commercial software, they may not solve large-size problems. The other solution approach is heuristics, which are capable of solving large-size problems, but fail to prove optimality. Several heuristics that are utilized for location problems are “greedy adding”, “tabu search”, “simulated annealing”, “genetic algorithms” and “langrangean relaxation based heuristics”.

After reviewing the solution approaches in the literature, we decided to build up our own solution methodology based on the genetic algorithm which is proved to be producing reliable results in location network problems (Altman et al., 2002 and Jamaa et al. 2004). On the other hand, we have developed a greedy ADD-and-DROP algorithm as well to determine whether a genetic algorithm does really outperform an-easy-to-implement heuristics in this three-level coverage approach.

In this chapter, the heuristics we develop are explained in detail. The fundamentals related with different phases and parameters of GA (Genetic Algorithm) are defined.

4.1 ADD and DROP Heuristic

The ADD algorithm, also referred to as “greedy” or “myopic” algorithm, was first suggested by Kuehn and Hamburger (1963) for a slightly different problem as well as by Feldman et al. (1966).

It follows a very simple strategy: within each iteration it locates a new ambulance at the location that contributes most to the objective in our problem. It starts with an empty configuration and stops if the configuration reaches the desired number of p ambulances. Once an ambulance site is established in the configuration, it is never moved. The algorithm always terminates and the computation time is known and small. However, the ADD algorithm is very likely to get caught in a local optimum. Consider the linear network of five nodes in Figure 4 with a demand of one for each point and a distance of one between every two neighbour points.

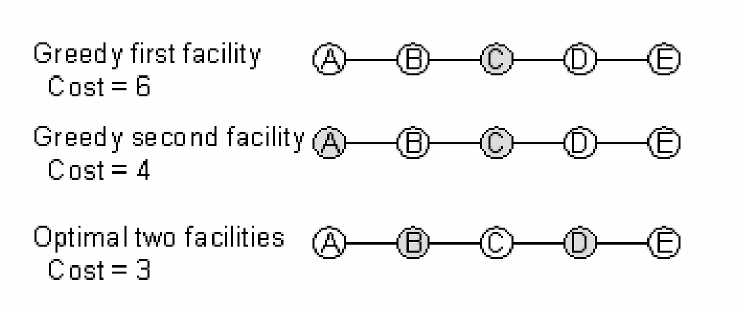


Figure 4 : The ADD algorithm’s local optima

All points are ambulance site candidates. The first point the algorithm chooses is C, because it has the minimal distance of a one-point configuration of 6. The second point can be any of A,B,D, and E, since a new service at any of these points results in a total cost of 4. However, the global optimum of a two-point configuration is B and D and has a total distance of only 3.

The ADD algorithm is simple, and rather fast; the computational time needed is determined only by the size of the input data such as the number of candidate location sites and demand points. So every problem of the same size needs the same time, that is, the algorithm does not need a longer computational time, if problem as described by the data is difficult.

The ADD algorithm is coded in MATLAB environment for the three-level maximal covering location problem and presented in Appendix B.

The DROP algorithm is based on the naive drop heuristic developed by Chardaire and Lutton (1993). Initially, all candidates are part of the configuration. In each iteration one candidate is removed until only the given number of p candidates is left in the configuration. At each iteration the candidate site that produces the smallest decrease (deterioration) in the objective function value is dropped.

This strategy is the reverse of the ADD heuristic approach that adds one candidate to the configuration at each iteration. Like the ADD algorithm, it is simple and tempted to return a sub-optimal result.

The DROP strategy can be observed in reality, if a company reduces the number of its branches by always shutting down the one, which affects the overall accessibility least at that stage.

The DROP algorithm is coded in MATLAB environment for the three-level maximal covering location problem and presented in Appendix C.

4.2 Genetic Algorithm Based Heuristics for Three-Level Maximal Covering

After first introduced by Holland (1975) and disciplined by Goldberg (1989), genetic algorithms (GA) have been applied to various problem situations. Basically GAs are adaptive heuristic search algorithms based on the evolution idea of natural selection and genetics. In the optimization environment, they represent a useful way of random search via keeping the past information of the solution set in genes.

GAs encourage the survival of the fittest among the individuals over periods for solving a problem. Each generation consists of a population character string that has a similar representation with our DNAs. Each individual stands for a solution in a search space. These individuals are then subject to an evolutionary process to form the best individual in terms of “fitness”.

There are several components which must be defined properly in order to form a well working GA. They are basically:

- Chromosome coding.
- Methods to generate an initial population.
- Determining the fitness function.
- Defining the evolutionary process including genetic operators, parent selection technique, crossover, mutation and replacement strategies.
- Defining the parameters for the evolutionary process by which the duration for the process is defined, and crossover and mutation rates.

These components of the GA which are related to our solution procedure for the three-level maximal covering location problem are explained in the following sections.

4.2.1 Chromosome Coding

While developing a GA, chromosome coding is critical in terms of its ability in representing every possible solution and also avoiding infeasible solutions in the population.

Mainly there are two representation schemes for chromosome coding in GAs: (i) binary representation, and (ii) non-binary representation. The binary representation uses base 2 and any value or character could be converted into base 2 from base 10. An example of an operation coded in binary representation is shown below.

0: **0000**
1: **0001**
2: **0010**
3: **0011**
4: **0100**
5: **0101**
6: **0110**
7: **0111**
8: **1000**
9: **1001**
+: **1010**
-: **1011**
*****: **1100**
/: **1101**

For example, with the above representation, the expression (6+5*4/2+1) could be stated as:

0110	**1010**	**0101**	**1100**	**0100**	**1101**	**0010**	**1010**	**0001**
6	**+**	**5**	*****	**4**	**/**	**2**	**+**	**1**

The mathematical value of the above gene is 17. The later operations with the above offspring and other population members could produce better values in terms of fitness.

However, in several instances it is much more useful to use non-binary representations in order to handle information tracking on genes much more easily and not to cause difficulties in genetic operations over offspring (Özgönenç, 2006).

To implement the genetic algorithm (GA) for our problem, non-binary representation is chosen and the coding of the information is carried out in the following manner: a chromosome corresponds to a particular set of available ambulances and the location indices they are assigned to. For example, with 60 possible location sites and 5 available ambulances, a possible chromosome coding for a feasible solution can be seen in Figure 5; the first ambulance is located at the location site indexed by 25, and the second ambulance is located at site 42, and so on.

Ambulance no	1	2	3	4	5
Assigned location site	25	42	5	52	11
Chromosome	25	42	5	52	11

Figure 5 : Chromosome representation

4.2.2 Fitness function

The fitness function of the GA heuristics to solve the three-level maximal covering location problem is the same as the objective function defined in the integer programming model. That is to say, the individuals which have higher total coverage values with respect to coverage levels and the weights are said to have higher fitness values.

4.2.3 Initial population generation

A gene pool that has a wide variety is much more preferable than a small gene pool; because in this way, the probability of forming the best individuals is increased. This makes initial population generation in GAs crucial. The objective is to have an optimal population size in which every possible solution (chromosome representation) can be attained through the genetic operators.

One way of forming an initial population is forming it randomly. During the random generation process, the population is controlled such that it is checked whether it includes every possible gene that can contribute to the fitness function or not. So, with an initial population generation method which is not fully randomized, and with control on genes included in the population, a gene pool is created, which is a starting point to generate the best individuals. However, when the overall size of the problem is increased, a predetermined constant population size might prevent keeping genetic variety in the population, and the initial population size might require modifications in that sense. In general, the population size is defined as: n/p, where n is the possible number of sites, and p is the number of available vehicles.

In the GA developed to solve the three-level maximal covering location problems, we have used two different approaches to test the effect of initial population size and its formation on the performance of the algorithm. In the first approach, we apply the traditional way which is defined above *(n/p)* while in the second approach we adopt the idea developed by Erkut et al. (2003). The population size is based on the "density" concept which is defined by the possible number of sites (n) and the available vehicles p. $S = \binom{n}{p}$ is the total number of possible solutions for the problem, while $d = \lceil n/p \rceil$ is the rounded density of the problem. Then the initial population size is defined as :

$$P(n,p) = \max\left\{2, \left\lceil \frac{n}{100} * \frac{\ln(S)}{d} \right\rceil\right\} * d$$

In this sense, for a 1000-node network problem with 8 ambulances the initial population size would be:

$$P(1000,8) = \max\left\{2, \left\lceil \frac{1000}{100} * \frac{\ln(\binom{1000}{8})}{125} \right\rceil\right\} * 125 = 375$$

The effects of p and n on the initial population size is shown below in Figures 6 and 7.

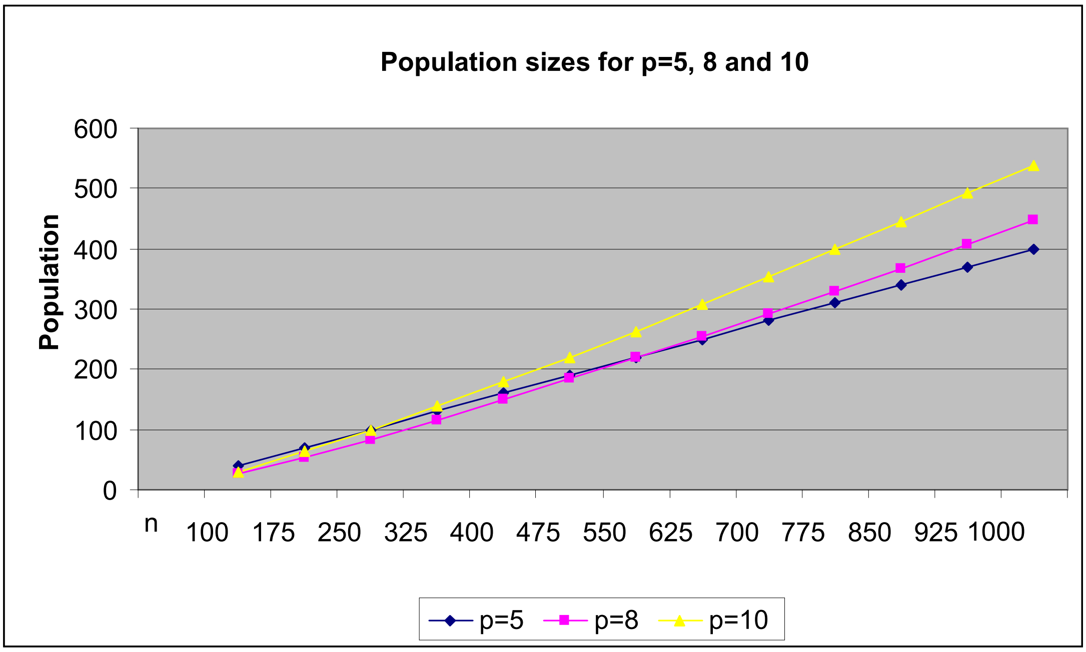


Figure 6 : Population size as a function of *n* for three different values of *p*

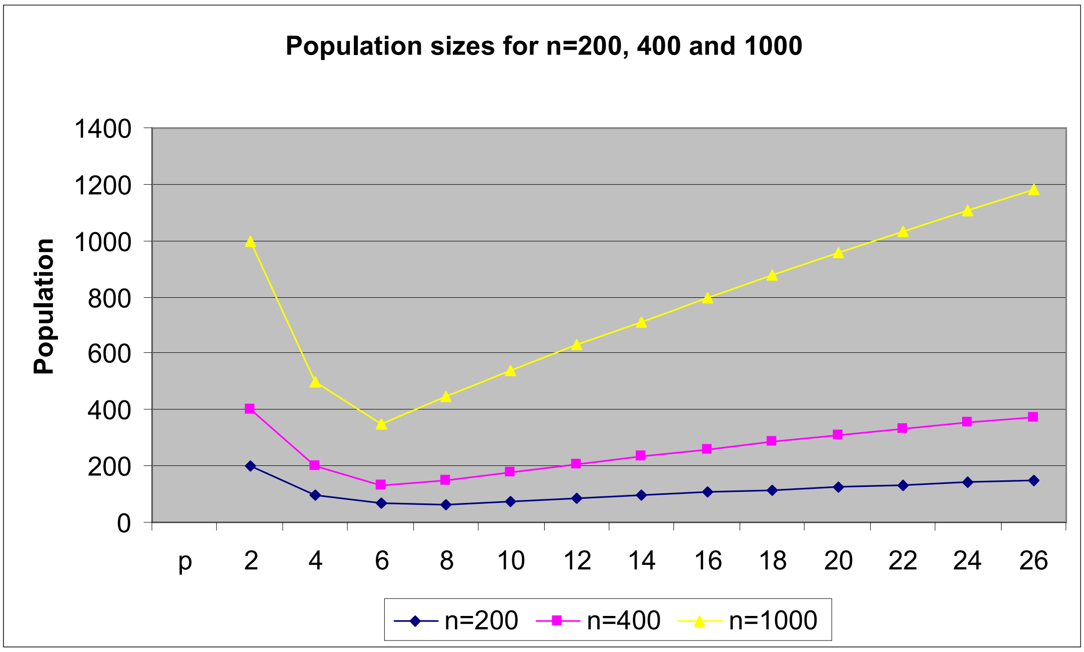


Figure 7 : Population size as a function of *p* for three different values of *n*

After the population size is determined, the next step is to initialize the population in such a manner that the gene pool is unbiased. Hence, in this phase, we adopt the idea based on the uniform probability distribution so that each gene has the same chance of appearing in the initial population.

4.2.4 Selecting the parents

The genetic heritage is carried to an offspring by two parents. Although it is a common fact in humans that two perfect parents are required for a perfect child, it is not always the case in GAs. Sometimes a weak parent could enable the production of an offspring that has rich treats in terms of gene design. This is due to the fact that a weak parent might hide some powerful chromosomes in its design waiting to show up.

For this reason it is not always preferred to give chance to the stronger parents in terms of fitness value. There are several parent selection techniques which can outperform the other techniques in some aspects and become an underperformer in some other aspects. The selection techniques are summarized below.

- *Roulette-wheel selection*: each individual possesses a part of area according to its fitness in the selection scheme, and the higher the individual fitness, the higher the probability of that individual to enter the offspring production is.
- *Stochastic universal sampling*: similar to roulette wheel selection, this time the fitness scheme for individuals is divided again according to the fitness values, then the mating population is chosen one by one starting from the first pointer in the fitness scheme. Pointers are equally positioned in the scheme and the number of pointers is as many as the number of required parents. This method enforces production of an offspring which is closer to what is desired, compared to the roulette wheel selection.
- *Local selection*: every individual is mated with an individual around its neighbourhood defined with specific characteristics on purpose.

- *Truncation selection*: this is an artificial selection method by which only some percent of individuals are subject to mating within the fitness scheme. For example, an individual may be forced to be chosen as a parent from the last 20 % of the best group.
- *Tournament selection*: a number "*Tour*" of individuals is chosen randomly from the population and the best individual from this group is selected as a parent. This process is repeated as often as individuals must be chosen. These selected parents produce offspring. The parameter for tournament selection is the tournament size *Tour*. *Tour* takes values ranging from 2 to total number of individuals.
- *Random selection*: parents are selected uniform randomly from the population. So, every individual has the same chance to be selected for the crossover. Though convergence to the best value takes time in this way, this method increases the genetic diversity in the offspring by preserving the weak genes in the gene pool.

4.2.5 Generating new members (Crossover)

In a classical GA application, selected parent's genes are merged with a prescribed crossover operation. Generally, the chromosomes of the parents are split into two or three, creating four or six partial chromosomes, and then these partial chromosomes are combined with the defined procedures to create two new members. The classical way of producing a new member is shown below in Figure 8.

Parent 1	5	3	8	1	2	7
Parent 2	11	6	9	4	13	10
Child 1	5	3	8	4	13	10
Child 2	11	6	9	1	2	7

Figure 8 : A single point crossover operator

In one of our GA heuristics, we adopt this method, whereas in the other GA heuristic we utilize the idea developed by Erkut et al. (2003). In the latter method, instead of the traditional crossover technique, a greedy procedure is followed. The chromosomes of parents are unified causing an infeasible gene combination with the total of m genes where $2p \geq m \geq p$ (repeating genes counted only once). After that, a greedy deletion heuristic is applied to the infeasible gene in order to make it a feasible one. The technique we apply is somehow a genetic engineering approach to the offspring formation. Besides, while applying the deletion heuristic, the repeated genes in both parents are preserved in the new offspring. The procedure is shown below in Figure 9.

<table>
<tr><td>Parent 1</td><td colspan="2">5</td><td colspan="2">3</td><td colspan="2">8</td><td colspan="2">1</td><td colspan="2">2</td><td colspan="2">7</td></tr>
<tr><td>Parent 2</td><td colspan="2">11</td><td colspan="2">6</td><td colspan="2">9</td><td colspan="2">4</td><td colspan="2">13</td><td colspan="2">10</td></tr>
<tr><td>Infeasible Child</td><td>5</td><td>3</td><td>8</td><td>1</td><td>2</td><td>7</td><td>11</td><td>6</td><td>9</td><td>4</td><td>13</td><td>10</td></tr>
<tr><td>Feasible Child, after deletion</td><td colspan="2">5</td><td colspan="2">8</td><td colspan="2">1</td><td colspan="2">11</td><td colspan="2">2</td><td colspan="2">10</td></tr>
</table>

Figure 9 : A single point crossover operator

As it is observed in the figure, the genes 3, 7, 6, 9, 4, and 13 are removed from the infeasible child one by one in order to make it a feasible and fittest offspring.

4.2.6 Mutation

One of the crucial problems in GA is to get out of the local optima. Mutation is a component to make jumps from local optima in GA. In our two GA approaches we use mutation idea in order to get away from the local optima after a prescribed number of iterations. In this context, iteration is to apply fully the new member generation procedure and to evaluate its fitness value. We perform mutation in such a way that a new member (gene) is randomly added into the chromosomes of a predefined number of parents, while the member that is going to be removed from the chromosomes of the selected parents in a uniform random manner.

4.2.7 Replacement

The new offspring in both GAs is evaluated according to its fitness values and compared with the existing ones in the population. The worst member(s) of the population are replaced by the new member(s) if the new ones outperform the existing ones.

4.2.8 Termination

In most of the GA applications, a convergence based stopping criterion is designed. The algorithm terminates after observing a defined number of iterations. For example one of the used stopping criterions in the literature is $n\sqrt{(n-p)}$ (Erkut et al. 2003). In our GA we take the iteration limit as 1000.

We summarize the properties of the two GAs below in Table 6.

Table 6 : Properties of applied genetic heuristics.

	GA 1	*GA* 2
Chromosome Coding	Non-binary	Non-binary
Fitness Function	The same with the Three-Level Maximal Covering Location Problem	The same with the Three-Level Maximal Covering Location Problem
Initial population generation (size)	*(n/p)*	$P(n,p)=\max\left\{2,\left\lceil \frac{n}{100}*\frac{\ln(S)}{d}\right\rceil\right\}*d$
Initializing the population	Uniform Random	Uniform Random
Selecting the parents	Random Selection	Random Selection
Generating new members	Classical Crossover	Creating an infeasible offspring then making it feasible by greedy deletion

Mutation	20% of the population are subject to mutation in each generation	20% of the population are subject to mutation in each generation
Replacement	Only the best 6 members from the previous population preserved	Only the best 6 members from the previous population preserved
Termination	After 200 successive iterations by which no improvement is observed or 1000 generations	After 200 successive iterations by which no improvement is observed or 1000 generations

The MATLAB codes for GA1 and GA2 and flowchart of the GAs proposed are given on Appendix D, E and F, respectively.

CHAPTER 5

COMPUTATIONAL STUDY

5.1 Preliminary Results

The solution approaches to the multi-level coverage problem have been tested on several problem instances which are classified according to the number of demand points and the available ambulances. The total number of demand points is 200, 400 and 1000 in the problem instances. The GAMS model could solve the instances with up to 500 demand points optimally using the computer, which is intel R CPU, 1,66 Ghz, with 1 GB RAM. So, the problem instances sized as 200 and 400 are solved by the GAMS model and the heuristics performances are compared with the optimal results. The problem instances of 1000 demand points could only be solved by the heuristics methods.

The demand points are randomly generated in the [0, 100^2] square based on the continuous uniform distribution. The weight for each demand point is assumed to be equal to 1, though the model allows them to be different for possible modifications in real life usage. The number of ambulances is 5, 8 and 10 for all problem instances. Euclidean distances between demand points and possible location sites are assumed.

In each problem instance the maximum distance among the demand points is found out and the third level coverage standard is set as the half of the maximum distance in the system, that is, $r_3 = \dfrac{\max(d_{ij})}{2}$. The simple justification for this arrangement is that the central point among those two points is able to cover all demand points with respect to r_3 . In fact, the coverage levels are subject to change according to real life conditions. For example r_3 can be set as the maximum distance or time limit in which the ambulance service still keeps its effectiveness. After that r_2 can be set as half of r_3, and r_1 as half of r_2. It should be noted that forcing coverage levels not to be tight may cause failure in evaluating the performance of solution methodologies; this is because when the coverage levels are set loose, every solution technique is able to find the optimal solutions by which every demand point can be covered in r_1 standard. In real life conditions, for example, r_1 could be defined as the maximum distance criterion in which a heart attack emergency must be responded, and r_2 is used as the distance limit in accessing the site of a traffic accident, and lastly r_3 is the least serious emergency case such that relatively a late response to the call does not considerably affect the patients' survival chance, like a knee injury or loosing consciousness.

The contribution of coverage levels to the objective function in all problem instances are set as 2, 1 and 0.5 for weights w_1, w_2 and w_3, respectively. This enforcement is made to encourage the hierarchical coverage that gives more weight to a closer coverage. The resulting effect of the coverage weights (w_1=2, w_2=1, w_3=0.5) on the coverage levels in any problem instance is shown in Figure 10 below.

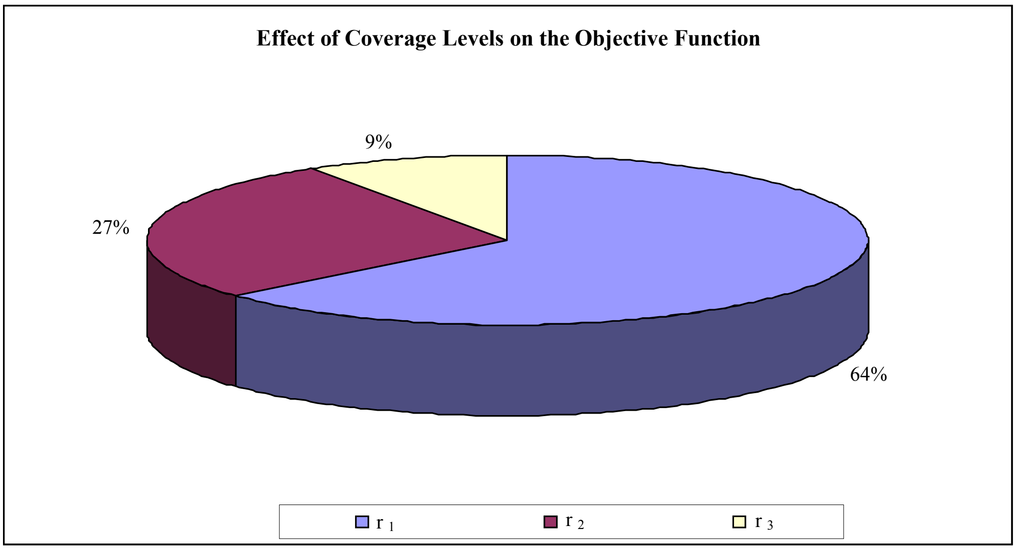


Figure 10 : Effect of weights of coverage levels on the objective function

Greedy ADD and DROP heuristics, and GA heuristics (GA1, GA2) are coded in MATLAB (2008r version) environment, and integer model is coded in GAMS environment and solved by using both MIP (mixed integer programming) and CPLEX solvers.

The initial results reveal that both greedy ADD heuristic and GA2 outperform the other two heuristics in terms of the objective function values. The test run results obtained for 200 and 400-demand point environments with 5, 8 and 10 ambulance servers are shown at the end of section 5.1 (Tables 14 and 15).

Greedy ADD and GA2 managed to obtain 0.74% and 1.96% close results to the optimum objective value, respectively, in 200-demand point environment with 5 ambulances. On the other hand, it is observed that when the number of available ambulances is increased, GA2 becomes much more efficient compared to greedy ADD heuristic. This result could be totally related to the myopic nature of the ADD heuristic. When the number of ambulances is not high, greedy ADD has a comparative advantage; it has to locate only a few ambulances and hence could find the global optima with a high probability in a short time.

As far as run time performances are considered, the DROP heuristic performance is the worst in 400-demand point environment, while it is relatively better in small sized, like 200-demand point environments. Since the DROP heuristic starts to eliminate the possible location sites from the entire set of solution space with respect to their contributions to the objective function, when the problem size gets larger, the run time performance of the DROP heuristic starts to be worse due to the fact that it has to examine a larger solution space. The performance of the ADD heuristic is outstanding such that it is able to find "good" results in a short period of time. Although GA2 performs best in terms of the solution quality as the problem size is increased, its run time performance becomes worse. This may be due to both the "crossover" method utilized in GA2 and the stopping criteria imposed at the beginning of the computer run. Fortunately, later examinations have revealed that the performance increase of GA2 is possible through some modifications in some specifications of the algorithm. We discuss these improvements in the following sections.

Also in the preliminary test runs, we keep the relevant data in order to compare the multi-level coverage and backup coverage approaches. This is an interesting analysis because the findings can direct us in the comparison of multiple coverage idea that encourages gathering vehicles at central stations, and multi-level maximal covering idea that aims to be closer to the demand points by spreading out the vehicles all over the network.

The comparison between the multi-level maximal covering location approach and the back up coverage location approach is performed through the concept of "**preparedness**". Since the concept is used as a qualitative measure, two people might not define the term always the same. However, while using the concept of preparedness, we adapt the term defined by Andersson and Varbrand (2007). Then the preparedness measure for demand point i can be calculated as:

$$p_i = \frac{1}{\lambda_i} \sum_{l=1}^{L_i} \frac{\gamma_i^l}{d_i^l},$$

where L_i is the number of ambulances that contribute to the preparedness for demand point i, d_i^l is the distance for ambulance l to demand point i, and γ_i^l is the contribution factor for an ambulance l to the preparedness measure of demand point i. λ_i is a weight associated with the demand for ambulances at demand point i. In this context, in our study the following assumptions hold for this factor:

- Each demand point is equally weighted like in the LSCM, so λ_i is equal to 1 for every demand point. However, our model allows λ_i to be greater than 1.
- The contribution factor γ_i^l for each ambulance is the same and equals 1.

As a result of these two assumptions, the preparedness for a demand point increases if an ambulance moves closer to the demand point. On the other hand, preparedness for a demand point may increase if the number of ambulances is increased in a specific distant point.

During the test runs, the optimal solutions found by GAMS/CPLEX reveal worse preparedness values, whereas heuristic approaches are able to produce much better preparedness values. These first conclusions helped us to gain insight and encouraged us to make comparison between multiple coverage and multi-level coverage approaches; we discuss the results in detail in the next sections.

As far as the framework of the test runs is considered, four problem instances are generated for each problem type, that is, for each (number of sites & number of ambulances) pair. All 4 runs are recorded for a problem type. The average values of four problem instances are recorded in Tables 7 through 9, and Figures 11, 12, 13 for *n*=200 problems, and in Tables 10 through 12 and Figures 14, 15, 16 for n=400 problems. The individual run results are provided in Tables 14 and 15, for *n*=200 and 400, respectively.

Table 7 : Average Objective Function values for *n*=200

averages *n*=200	Optimum solution (CPLEX)	**ADD**	**DROP**	**GA1**	**GA2**
# of ambulance	Objective fuction value				
5	510,00	506,25	469,75	482,00	500,00
8	604,50	592,00	565,00	569,00	592,75
10	645,50	627,50	602,25	599,50	630,00

Table 8 : Average Run time Values for *n*=200

averages *n*=200	Optimum solution (CPLEX)	**ADD**	**DROP**	**GA1**	**GA2**
# of ambulance	Run time (in seconds)				
5	26,45	5,88	77,33	18,23	52,48
8	26,00	4,00	74,43	20,98	192,75
10	51,35	4,25	74,68	22,63	335,00

Table 9 : Average Preparedness Index Values for *n*=200

averages *n*=200	Optimum solution (CPLEX)	**ADD**	**DROP**	**GA1**	**GA2**
# of ambulance	Preparedness				
5	27,05	27,25	27,00	26,50	26,50
8	42,15	42,75	42,50	44,50	43,00
10	50,75	51,50	50,25	50,75	51,00

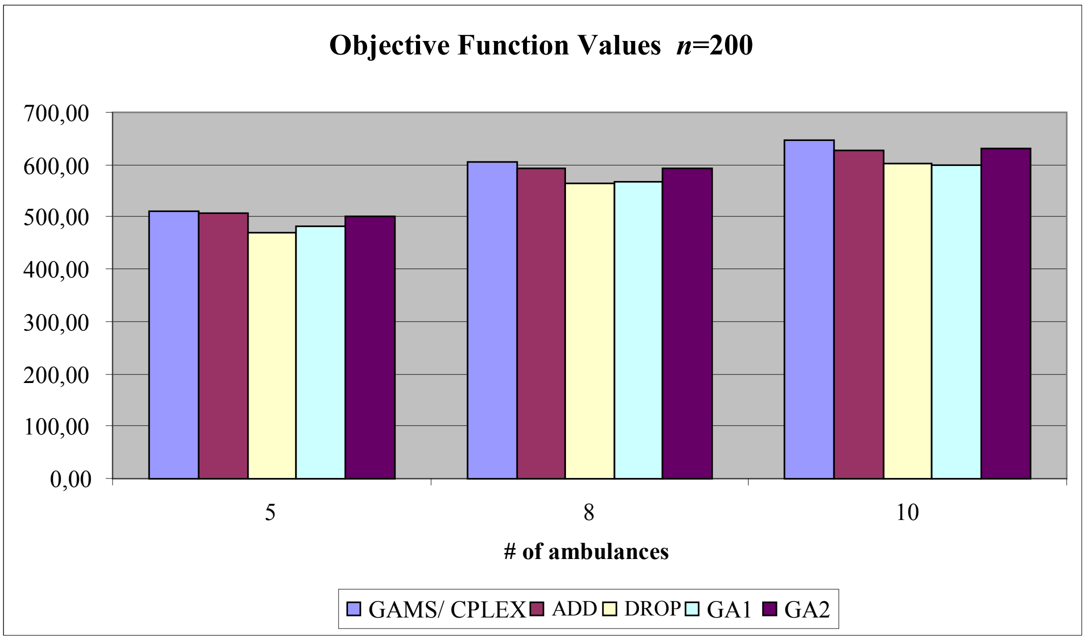


Figure 11 : Objective function values for *n*=200

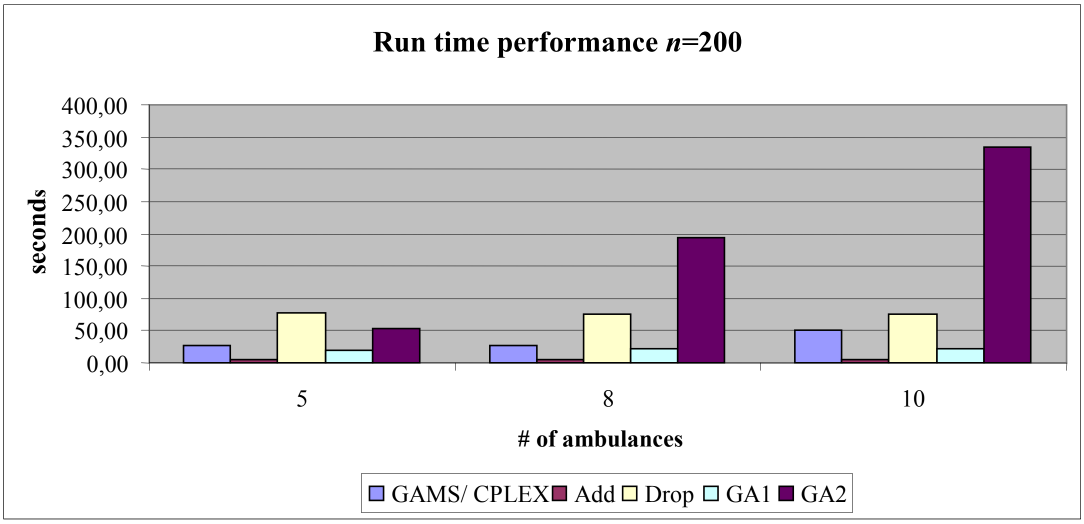


Figure 12 : Run time performance for *n*=200

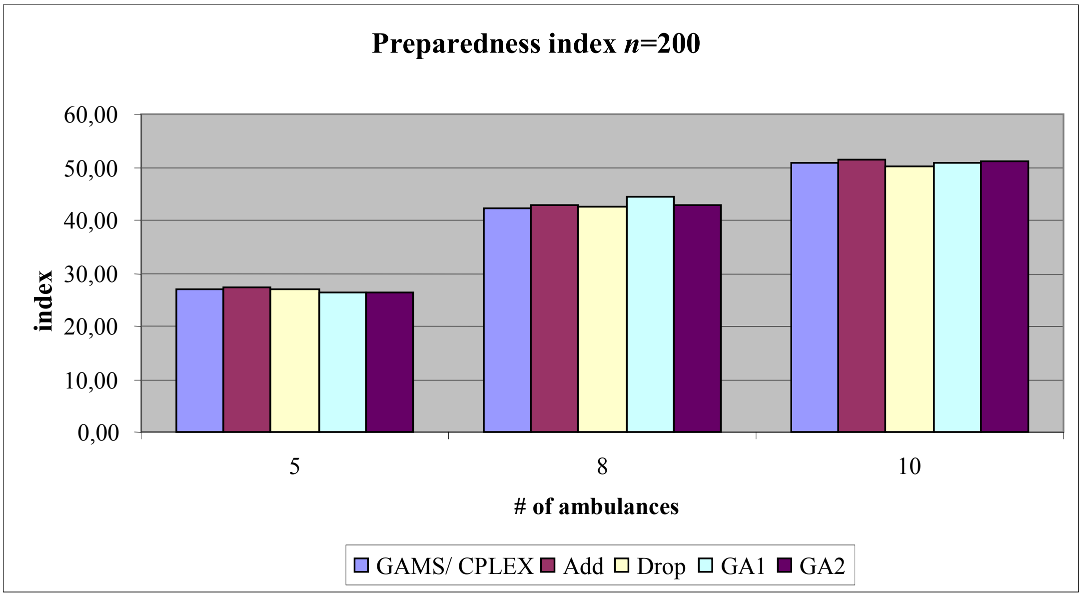


Figure 13 : Preparedness Index Values for n = 200

Table 10 : Average Objective Function Values for n=400

averages n=400	Optimum solution (CPLEX)	**ADD**	**DROP**	**GA1**	**GA2**
# of ambulance	Objective value				
5	1198,00	990,50	918,75	945,50	977,75
8	1313,00	1186,25	1126,00	1135,75	1177,25
10	1347,50	1252,50	1185,25	1196,00	1264,75

Table 11 : Average Run time Values for n=400

averages n=400	Optimum solution (CPLEX)	**ADD**	**DROP**	**GA1**	**GA2**
# of ambulance	Run time (in seconds)				
5	328,25	4,76	1209,50	31,75	90,76
8	300,25	5,64	1179,50	31,51	359,50
10	355,50	6,53	1186,75	29,38	597,25

Table 12 : Average Preparedness Index Values for n=400

averages n=400	Optimum solution (CPLEX)	**ADD**	**DROP**	**GA1**	**GA2**
# of ambulance	Preparedness				
5	48,60	56,00	52,25	53,75	55,00
8	63,63	87,25	85,25	87,25	89,00
10	76,80	107,25	107,25	107,25	105,50

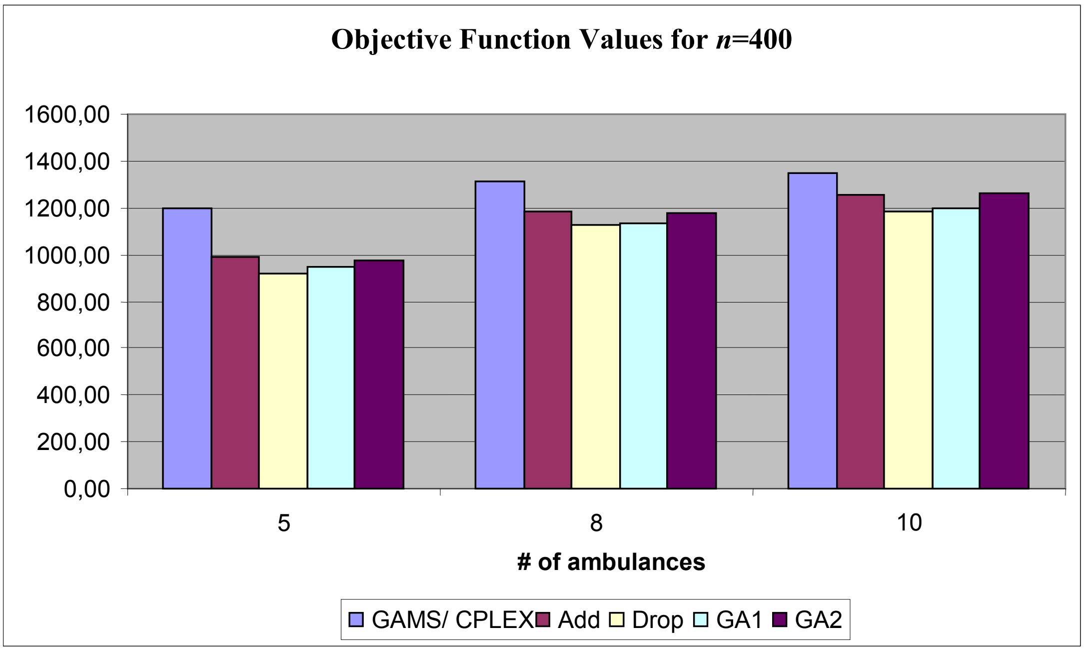


Figure 14 : Objective Function Values for n=400

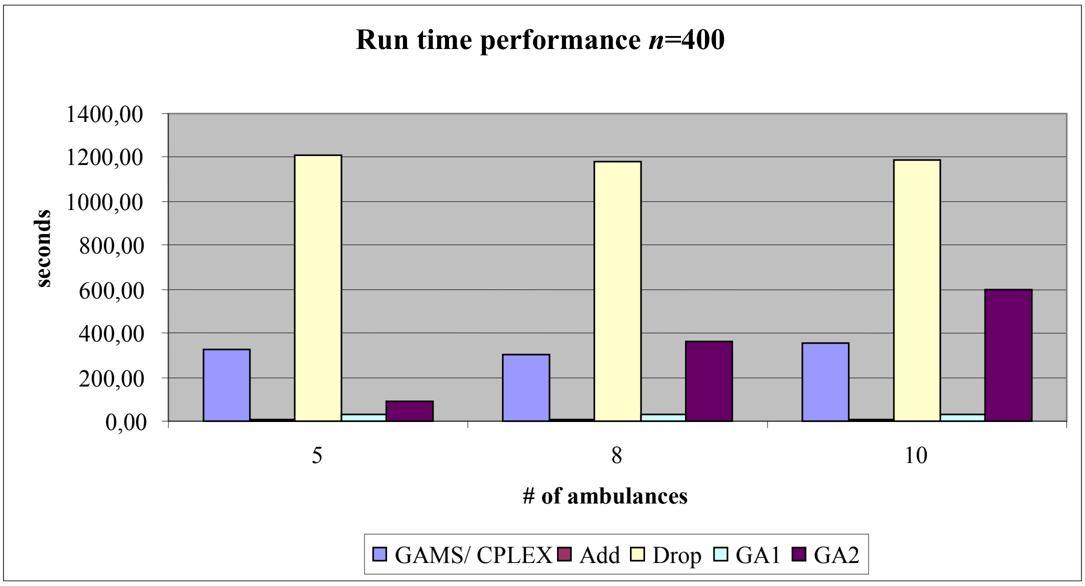


Figure 15 : Run time performance for *n*=400

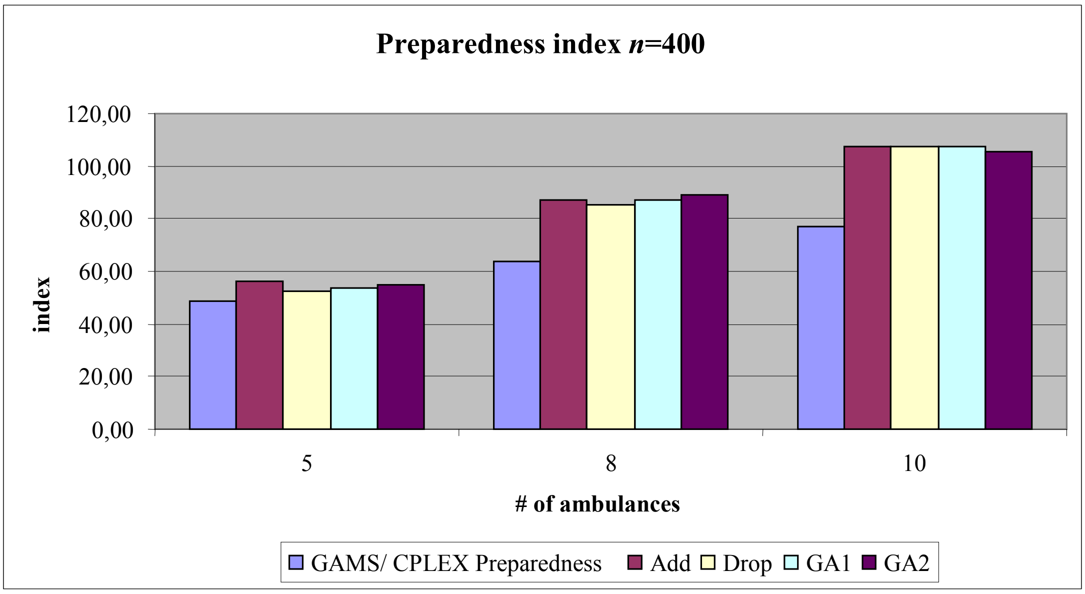


Figure 16 : Preparedness Index Values for *n* = 400

To sum up, GA2 and greedy ADD heuristics are the best performers in terms of objective function values. While DROP heuristic is the worst performer, GA1 moderately outperforms DROP heuristic. Although the run time of GA2 increases as the problem size gets bigger, its performance in terms of objective function value gets better. The average percentage gaps between the optimum objective function value (GAMS/CPLEX) and the values found by the heuristics are shown below in Table 13.

Table 13 : Performance of the methods with respect to the optimum solution

averages n=200	GAMS/ CPLEX	ADD	% gap with the optimum	DROP	% gap with the optima	GA1	% gap with the optimum	GA2	% gap with the optimum
# of ambulance	Objective value								
5	510.00	506.25	0.74%	469.75	7.89%	482.00	5.49%	500.00	1.96%
8	604.50	592.00	2.07%	565.00	6.53%	569.00	5.87%	592.75	1.94%
10	645.50	627.50	2.79%	602.25	6.70%	599.50	7.13%	630.00	2.40%
averages n=400	GAMS/ CPLEX	ADD	% gap with the optimum	DROP	% gap with the optimum	GA1	% gap with the optimum	GA2	% gap with the optimum
# of ambulance	Objective value								
5	1198.00	990.50	17.32%	918.75	23.31%	945.50	21.08%	977.75	18.38%
8	1313.00	1186.25	9.65%	1126.00	14.24%	1135.75	13.50%	1177.25	10.34%
10	1347.50	1252.50	7.05%	1185.25	12.04%	1196.00	11.24%	1264.75	6.14%

These results in Table 13 provide us some useful insights. First of all, when the problem size is small and the available number of ambulances is also small, the greedy ADD heuristic produces closer results to the optima. On the other hand, as problem size gets bigger and available number of ambulances increases, GA2 becomes the outperformer.

Table 14 : Computational results for $n = 200$

problem instance	n	p	**Objective function value**					**Run time (in seconds)**					**Preparedness index**				
			GAMS/ CPLEX	**ADD**	**DROP**	**GA1**	**GA2**	**GAMS/ CPLEX**	**ADD**	**DROP**	**GA1**	**GA2**	**GAMS/ CPLEX**	**ADD**	**DROP**	**GA1**	**GA2**
1	**200**	**5**	**513**	**512**	**468**	**481**	**504**	**28.8**	**11.7**	**76.5**	**16.3**	**52.3**	**27.3**	**28**	**26**	**28**	**27**
2	**200**	**5**	**519**	**519**	**485**	**504**	**505**	**26.7**	**4.1**	**77.3**	**18.6**	**54.4**	**27**	**27**	**25**	**27**	**26**
3	**200**	**5**	**507**	**503**	**460**	**478**	**502**	**24.7**	**3.9**	**78.3**	**17**	**52.3**	**26.2**	**27**	**29**	**25**	**26**
4	**200**	**5**	**501**	**491**	**466**	**465**	**489**	**25.6**	**3.8**	**77.2**	**21**	**50.9**	**27.7**	**27**	**28**	**26**	**27**
5	**200**	**8**	**598**	**593**	**550**	**569**	**584**	**23.5**	**4.0**	**74.3**	**23.4**	**199**	**44**	**44**	**42**	**42**	**42**
6	**200**	**8**	**584**	**572**	**555**	**551**	**571**	**26.8**	**4.1**	**74.1**	**22.3**	**187**	**40.8**	**41**	**43**	**42**	**42**
7	**200**	**8**	**619**	**595**	**579**	**574**	**612**	**21.9**	**3.9**	**74.7**	**23.4**	**196**	**43**	**42**	**41**	**48**	**42**
8	**200**	**8**	**617**	**608**	**576**	**582**	**604**	**31.8**	**4.0**	**74.6**	**14.8**	**189**	**40.8**	**44**	**44**	**46**	**46**
9	**200**	**10**	**648**	**638**	**600**	**606**	**630**	**82**	**4.2**	**74.7**	**24.2**	**320**	**49**	**52**	**47**	**47**	**52**
10	**200**	**10**	**654**	**638**	**615**	**602**	**644**	**50.8**	**4.2**	**74.5**	**24.6**	**377**	**52.7**	**55**	**52**	**52**	**52**
11	**200**	**10**	**640**	**618**	**592**	**594**	**618**	**46**	**4.3**	**75.2**	**16.4**	**320**	**52.7**	**51**	**53**	**54**	**53**
12	**200**	**10**	**640**	**616**	**602**	**596**	**628**	**26.6**	**4.3**	**74.3**	**25.3**	**323**	**48.6**	**48**	**49**	**50**	**47**

Table 15 : Computational results for $n = 400$

problem instance	n	p	GAMS/ CPLEX	ADD	DROP	GA1	GA2	GAMS/ CPLEX	ADD	DROP	GA1	GA2	GAMS/ CPLEX	ADD	DROP	GA1	GA2
			Objective function value					**Run time (in seconds)**					**Preparedness index**				
1	**400**	**5**	**1,206**	**995**	**930**	**942**	**979**	**430**	**4.80**	**1,233**	**38.80**	**88.50**	**49.60**	**55**	**53**	**57**	**55**
2	**400**	**5**	**1,195**	**985**	**903**	**952**	**980**	**321**	**4.96**	**1,209**	**29.00**	**89.15**	**49.90**	**55**	**51**	**51**	**56**
3	**400**	**5**	**1,188**	**986**	**927**	**948**	**974**	**307**	**4.66**	**1,201**	**29.20**	**88.70**	**52.90**	**61**	**52**	**54**	**55**
4	**400**	**5**	**1,203**	**996**	**915**	**940**	**978**	**255**	**4.63**	**1,195**	**30.00**	**96.70**	**42**	**53**	**53**	**53**	**54**
5	**400**	**8**	**1,301**	**1,155**	**1,125**	**1,126**	**1,152**	**330**	**5.70**	**1,180**	**32.50**	**348**	**67**	**87**	**86**	**95**	**87**
6	**400**	**8**	**1,324**	**1,233**	**1,130**	**1,154**	**1,206**	**194**	**5.66**	**1,175**	**28.55**	**390**	**64.90**	**89**	**84**	**85**	**92**
7	**400**	**8**	**1,318**	**1,176**	**1,090**	**1,096**	**1,181**	**341**	**5.60**	**1,179**	**30.60**	**353**	**60**	**88**	**86**	**85**	**90**
8	**400**	**8**	**1,309**	**1,181**	**1,159**	**1,167**	**1,170**	**336**	**5.60**	**1,184**	**34.40**	**347**	**62.60**	**85**	**85**	**84**	**87**
9	**400**	**10**	**1,362**	**1,284**	**1,195**	**1,216**	**1,288**	**342**	**6.30**	**1,176**	**33.20**	**597**	**76.70**	**105**	**107**	**105**	**100**
10	**400**	**10**	**1,322**	**1,230**	**1,202**	**1,208**	**1,247**	**330**	**6.40**	**1,180**	**35.00**	**586**	**77**	**111**	**110**	**112**	**111**
11	**400**	**10**	**1,356**	**1,242**	**1,168**	**1,174**	**1,252**	**420**	**7.10**	**1,189**	**22.60**	**604**	**77**	**107**	**108**	**109**	**102**
12	**400**	**10**	**1,350**	**1,254**	**1,176**	**1,186**	**1,272**	**330**	**6.30**	**1,202**	**22.70**	**602**	**76**	**106**	**104**	**103**	**109**

5.2 A Hybrid Heuristic Approach

The preliminary results directed us towards a hybrid heuristic by which the greedy ADD heuristic and GA2 are combined in such a way that the solution found by the greedy ADD heuristic is placed in the initial population of the GA2 method. For this purpose, twelve test runs are performed and the results have revealed that the hybrid heuristic is able to produce much more closer results to the optima by improving GA2's performance by at least 1% on the average (see Tables 16 and 17).

Table 16 : Performance of the Hybrid Heuristic

Problem instance	*n*	*p*	**ADD**	**GA2**	**Hybrid**	**ADD**	**GA2**	**Hybrid**
			Objective function Value			Run Time (in seconds)		
1	**400**	**5**	**983**	**950**	**983**	**4.60**	**89**	**94**
2	**400**	**5**	**1004**	**998**	**1004**	**7.10**	**86**	**86**
3	**400**	**5**	**1021**	**999**	**1021**	**4.60**	**85**	**90**
4	**400**	**5**	**1042**	**1003**	**1042**	**4.60**	**87**	**93**
5	**400**	**8**	**1185**	**1194**	**1203**	**5.50**	**358**	**349**
6	**400**	**8**	**1146**	**1174**	**1178**	**5.50**	**339**	**343**
7	**400**	**8**	**1167**	**1133**	**1167**	**5.50**	**344**	**345**
8	**400**	**8**	**1198**	**1204**	**1200**	**5.50**	**395**	**383**
9	**400**	**10**	**1270**	**1268**	**1270**	**6.20**	**586**	**649**
10	**400**	**10**	**1261**	**1250**	**1275**	**6.30**	**665**	**584**
11	**400**	**10**	**1246**	**1272**	**1280**	**6.30**	**581**	**638**
12	**400**	**10**	**1264**	**1250**	**1270**	**6.30**	**582**	**597**

Table 17 : The percentage improvement by the Hybrid Heuristic

	ADD	**GA2**		**Hybrid Heuristic**	**% gap with the best performer**
Objective Function Value Averages					
5	1012.50	987.50		1012.50	0.00
8	1174.00	1176.25		1187.00	0.91
10	1258.00	1260.00		1273.75	1.08

Having developed the hybrid heuristic method, we have tried to make a progress in the run time of it. After including the solution of the ADD heuristic in the initial population, the run time results of the GA2 are still not satisfying. For this reason, two crucial factors of GA2 that may cause run time to be longer are examined: the stopping criteria and the initial population size.

In GA1 all the runs are performed with the initial population size of (*n/p*) and in GA2 the initial population size are calculated based on the formula below:

$$P(n,p) = \max\left\{2, \left\lceil \frac{n}{100} * \frac{\ln(S)}{d} \right\rceil \right\} * d$$

In both genetic heuristics we use two stopping criteria: (i) the generation limit, 1,000, (ii) the stopping condition, that is, no improvement in fitness value after 200 generations. It is a fact that the initial population size and the stopping criteria would considerably affect the run time performance of the heuristics, since they directly decrease or increase the problem's solution space.

In order to test the effects, a two-factor factorial design environment is used. A total of 12 runs (problem instances) are performed and the results are analyzed in MINITAB statistical software package. The results are shown below in Table 18.

Table 18 : Test runs for the factorial design n=400 p=8

Run	Beginning Population size	Termination condition	Objective Function Value
1	50	1*	1168
2	50	2**	1126
3	50	1	1142
4	50	2	1156
5	50	1	1156
6	50	2	1178
7	150	1	1199
8	150	2	1206
9	150	1	1178
10	150	2	1198
11	150	1	1166
12	150	2	1171

*Termination condition 1: total generation=500, stop if there is no improvement after 100 generations.
**Termination condition 2: total generation=300, stop in there is no improvement after 50 generations.

As shown in the MINITAB output in Figures 17 and 18 (the detailed statistical results given in Appendix G), change in the stopping criteria conditions does not have a dramatic effect on GA2 performance; therefore, by this experiment, it is decided to decrease the population generation limits in the stopping criteria. The effect of this reduction is observed in the following runs in Table 19 such that more than 75% of improvement in run time is achieved. The beginning population size is left the same as the proposed formula dictates.

As the results in Table 19 indicate, by making reductions in the stopping criteria parameters, a reasonable amount of run time is saved. As it is mentioned in detail in the following sections, in the problems with 1000 demand points and 10 ambulances, the modified GA2's (hybrid heuristic) run time is about 15 minutes, while it is impossible to track an optimal solution in GAMS environment.

Table 19 : The effect of termination condition on run time n=400 p =10

Run	Termination Condition	Run time	Obj. Value
1	1000/200*	1449	1280
2	300/50**	352	1278
3	1000/200	1350	1264
4	300/50	345	1261

*Termination condition 1000/200: total generation=1000, stop if there is no improvement after 200 generations.
**Termination condition 300/50: total generation=300, stop in there is no improvement after 50 generations.

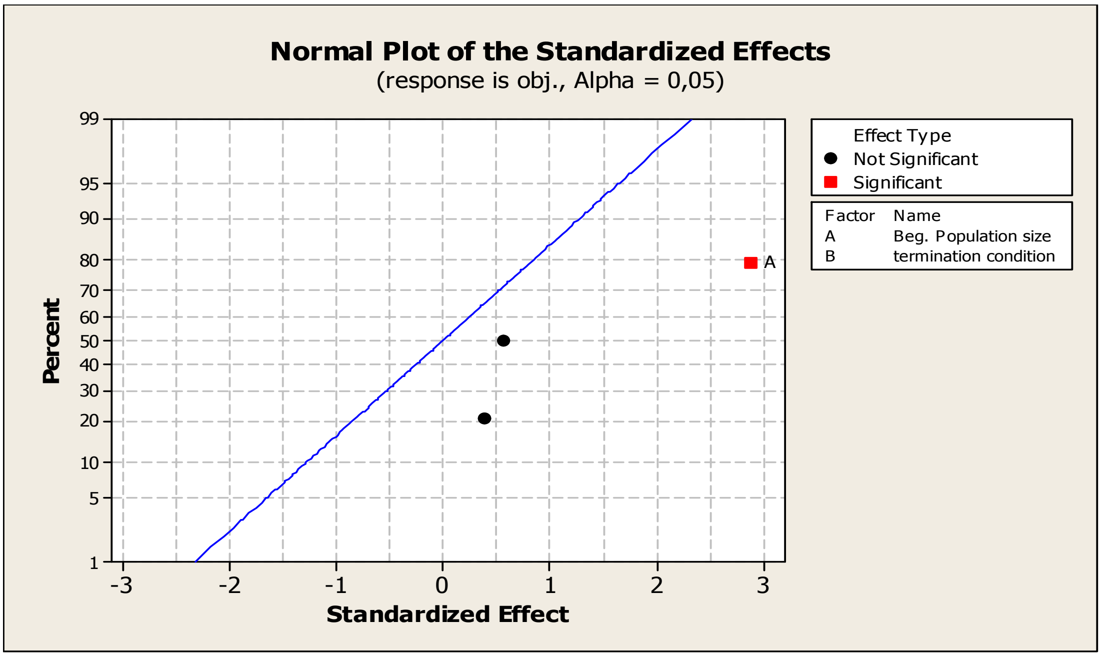


Figure 17 : The Minitab Effects Plot

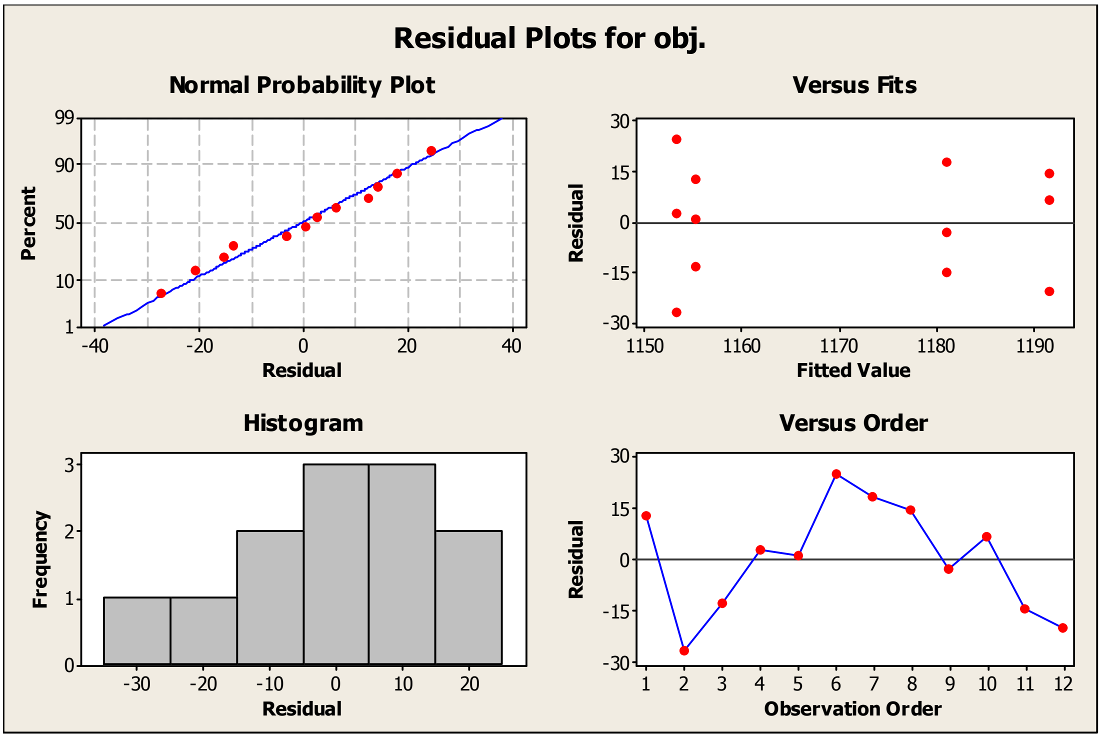


Figure 18 : The minitab Data analysis plot

5.3 Comparison between multi-level and multiple coverage approaches

In the mathematical model of multi-level coverage approach when the constraints (3.8), (3.9) and (3.10) are removed, the model turns out to be a multiple coverage model in which a demand point can be covered more than once. The solutions reveal that multiple coverage approaches tend to find 'centralized' solutions as shown in Figure 19, whereas multi-level coverage approach tries to disperse the located ambulances in such a way that their coverage circles at the third level, (r_3) are almost tangent to each other.

For example, in the examples shown in Figures 19 and 20 with 400 demand points and 10 ambulances, the multiple coverage approach locates the ambulances in a centralized manner in a densely populated area. On the other hand, multi-level coverage approach locates ambulances in such a dispersed manner that each ambulance is located in a dense populated area with its coverage circle (r_3) at the third level being tangent to another ambulance's third-level coverage circle.

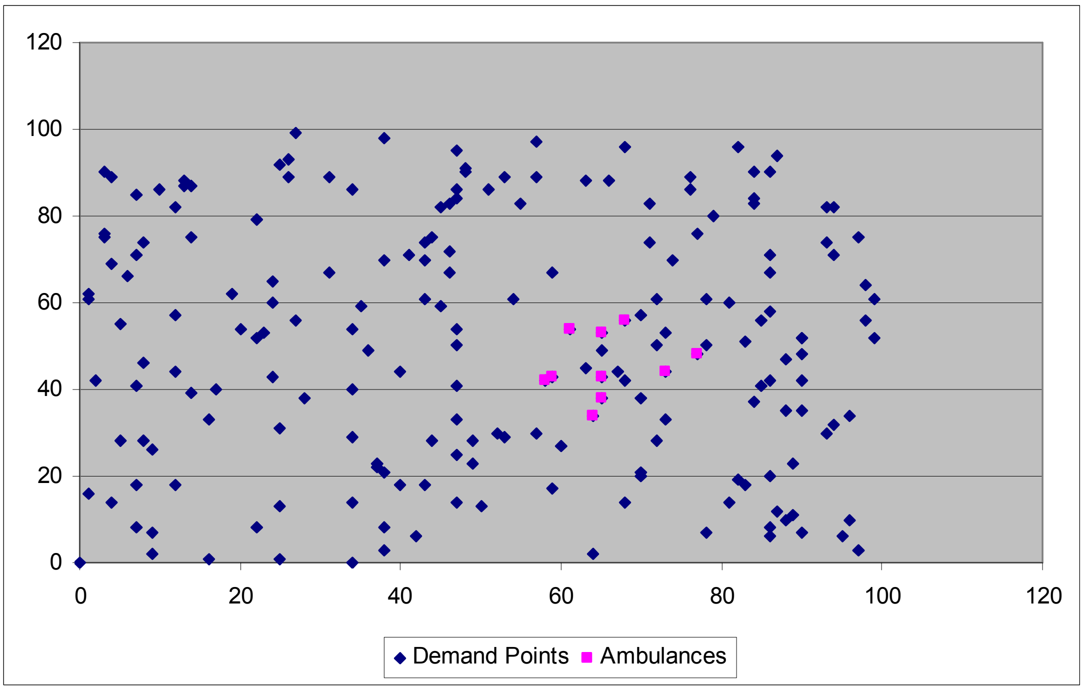


Figure 19 : Location of ambulances with the multiple coverage approach, n=400 p=10

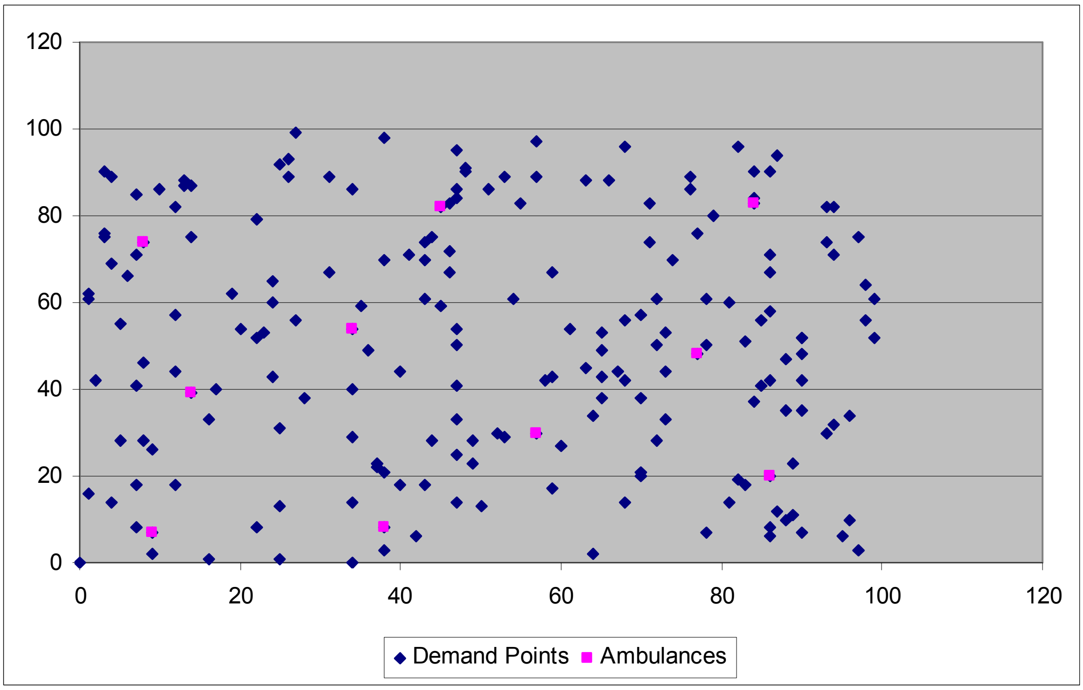


Figure 20 : Location of ambulances with multi-level coverage approach, n=400 p=10

In the above example, the preparedness index values for multi-level and multiple coverage approaches are 79 and 88, respectively. These results are derived according to the 'preparedness' measure defined in Chapter 3; and the higher the index value is, the better the preparedness condition is. For every demand point, all ambulances are taken into consideration that the demand point falls in the coverage range of (r_3), and the sum of the reciprocals of the Euclidean distance between a demand point and each ambulance is calculated. Further analysis with a more refined preparedness measure is presented in the following sections.

The results in the previous figures direct us towards an important conclusion that when the ambulances are centralized, the total system preparedness increases. It was noted before that the heuristic approaches had yield better preparedness results than the real optimal solutions. This was due to the fact that the non-optimal solutions in multi-level approach causes to locate ambulances closer to each other and this closeness causes the preparedness to be higher.

From this point on, the unification of both multi-level and multiple coverage ideas could provide useful real life applications both in ambulance location and relocation problems as well as in many other strategic location problems. To make it clear, multiple coverage idea locates the ambulances in such a centralized manner that many demand points are out of the critical coverage ranges, though the total system preparedness is increased (only the closer points to the ambulances get much of the benefit). On the other hand, multi-level coverage approach can locate the ambulances as close as possible to the demand points. So, when our problem is to identify a number of strategic ambulance stations in a city for a specific time interval, the multi-level approach seems to be the best. On the other hand, if the question is where the ambulances should be located in these areas in which stations are opened, the answer is stations itself.

These findings are another justification of the “risk pooling” concept that is faced with frequently in supply chain management problems. The solution methodologies developed in this study (the three-level coverage idea solved by a hybrid heuristic method) could be adapted to several application fields such as placing recycle collection baskets or placing sensitive radars or ATM machines for banks. To illustrate, a regular citizen is generally motivated to keep the materials that can be recycled. However, the recycle bins are not usually located close enough to the citizens, and because of this, the citizens throw the material into ordinary garbage collection points. On the other hand, when the recycle bins are closer, they get full in a short period of time. The possible solution is to place the recycle bins as close as possible to the citizens and with multiple units at the same place. By this way, when a citizen wants to throw the material, most probably he/she will find a closer point in which an empty bin exists.

5.4 The Extended Analysis

It could be questioned whether the developed methods could give different results with the specifically designed data. Actually, this argument is valid since the dispersion of people in a region resembles a patterned data rather than uniformly distributed data in real life. In addition to this point, the defined preparedness measure could be modified in such a manner that being far away from the demand point does not worsen the preparedness contribution of the ambulance to the demand point that much. For this reason, the distance between the ambulance and the demand point could be divided by the square root of the Euclidean distance between the two, rather than the distance itself. So, the refined preparedness measure, p_i, could be stated as:

$$p_i = \frac{1}{\lambda_i} \sum_{l=1}^{L_i} \frac{\gamma_i^l}{\sqrt{(d_i^l)}}$$

In order to test the above aspects and run time performance of the developed hybrid heuristic, firstly "*Extended SOLOMON's VRPTW instances*" are used. Although the data is designed specifically for vehicle routing problems (VRP), the dispersion of the demand points are suitable since they are classified as clustured, uniform and uniform-clustured dispersions. The data are available at http://www.fernuni-hagen.de/WINF/ touren/ menuefrm/probinst.htm.

SOLOMON's problem instances are desgined for 200-400-600-800 and 1000-demand point environment. Although the modifications according to the vehicle routing problems create many possible problem instances, in terms of demand point dispersion, there are five possible patterns for each demand size. In problem instances, *r* stands for uniformly distributed data, *c* stands for clustured data and *rc* stands for uniform-clustured data. An example of SOLOMON's problem instances with 400 demand points and in form of *rc* is shown in Figure 21. The hybrid heuristic solution is shown in Figure 21 with available 10 ambulances. It should also be noted that, the set of demand points and the set of possible location sites are still the same.

Only 200, 400 and 600-demand points sized SOLOMON's instances are analyzed since the exact algorithm in GAMS environment is not able to solve the problems sized more than 600 to optimality. The results are shown in following Tables 20, 21 and 22.

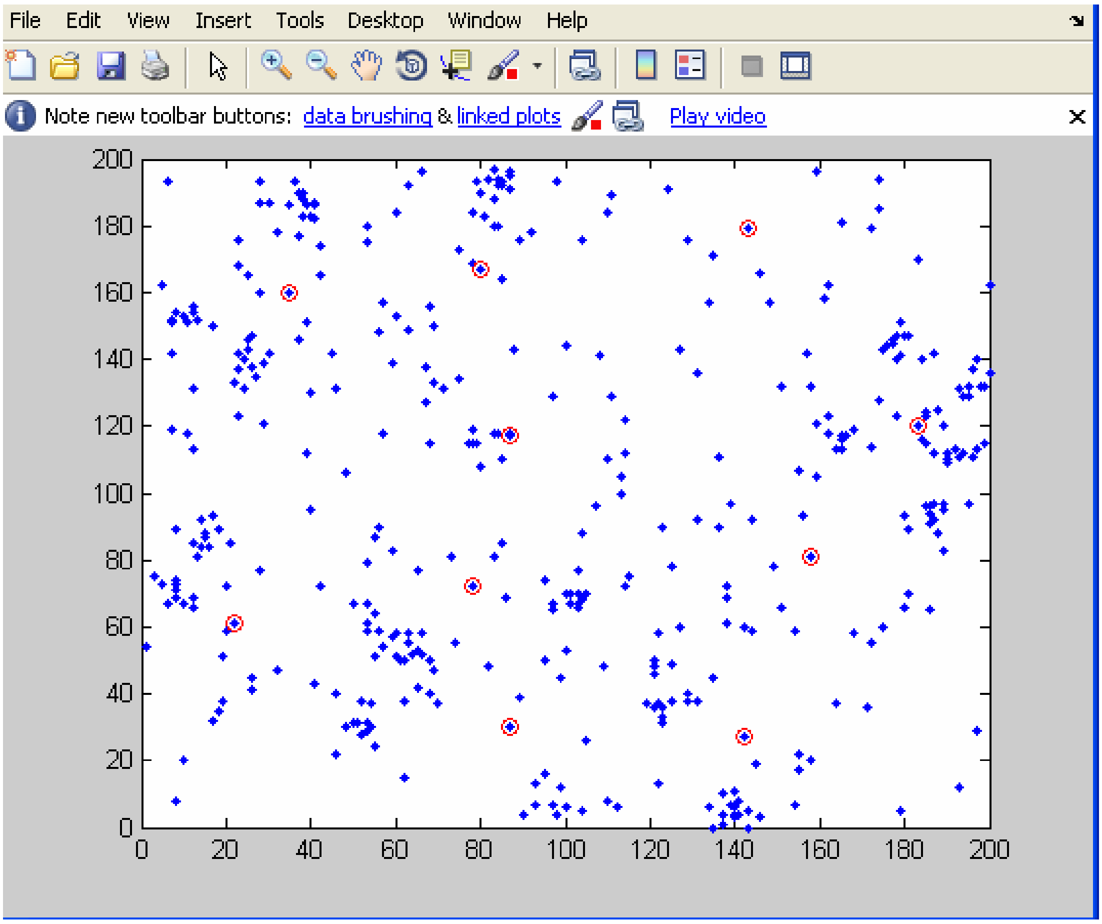


Figure 21 : The solution for SOLOMON's *rc* problem instance, *n*=400, *p*=10

Table 20 : SOLOMON's problem instance n=200

n=200

problem	#of ambulance	Objective			Run time			Preparedness		
		ADD	Hybrid	CPLEX	ADD	Hybrid	CPLEX	ADD	Hybrid	CPLEX
c1	5	572	572	572	7.26	17.2	28.9	110	110	109
c2	5	590	590	591	6.9	17.2	26.8	124	124	121.3
r1	5	514	514	515	6.9	17.5	21.7	118	118	112
r2	5	514	514	515	6.9	17	24.9	115	115	116
rc1	5	550	554	566	6.9	17	27	122	114	114
c1	8	656	656	664	7.2	54	30	176	176	157
c2	8	666	666	674	7.1	55	31.2	193	193	194
r1	8	598	600	608.5	7.2	54.7	36	179	168	170
r2	8	596	606	608.5	7.2	54.6	45	177	168	173
rc1	8	644	644	650	7.2	54.2	25	184	184	185
c1	10	694	694	700	7.4	89	38	222	222	217
c2	10	680	682	692	7.3	90	41	237	247	235
r1	10	632	632	648	7.3	89.5	27	215	215	218
r2	10	632	632	648	7.4	90	30	215	215	215
rc1	10	664	664	670	7.4	90	36	222	222	233

Table 21: SOLOMON's problem instance n=400

n=400

problem	#of ambulance	Objective			Run time			Preparedness		
		ADD	Hybrid	CPLEX	ADD	Hybrid	CPLEX	ADD	Hybrid	CPLEX
c1	5	1013	1013	1023	7.9	51.5	167	168	168	183
c2	5	1050	1065	1079	18.6	52.7	285	220	212	200.4
r1	5	1002	1011	1021	7.9	51.4	284	195	205	187
r2	5	1002	1008	1021	8.1	51.4	213	195	195	187.7
rc1	5	1074	1079	1088	10.22	80.1	171	183	191	184
c1	8	1205	1205	1209	12.54	210	131	277	277	271
c2	8	1264	1264	1302	14.9	199	251	321	321	335.4
r1	8	1173	1180	1221	38.8	290	636	318	294	318.4
r2	8	1173	1184	1220	9.3	230	428	318	317	316
rc1	8	1239	1241	1265	8.6	290	331	307	304	309
c1	10	1284	1284	1294	13	349	132	353	353	336
c2	10	1314	1346	1362	9.6	347	326	408	399	404.7
r1	10	1259	1264	1305	18.6	355	1052	391	377	350
r2	10	1259	1265	1305	9.7	522	1052	390	388	345
rc1	10	1308	1310	1333	29.6	504	290	371	369	372

Table 22 : SOLOMON's problem instance n=600

n=600

problem	#of ambulance	Objective			Run time			Preparedness		
		ADD	Hybrid	CPLEX	ADD	Hybrid	CPLEX	ADD	Hybrid	CPLEX
c1	5	1548	1548	1550	16.3	167	309	238	238	237.71
c2	5	1671	1671	1704	9.8	65	358	238	238	239.3
r1	5	1488	1497	1520	10.1	60.5	377	252	246	242.3
r2	5	1488	1488	1520	9.8	57	376	252	252	242.3
rc1	5	1560	1560	1588	9.8	68	420	260	260	252.5
c1	8	1830	1834	1840	12.16	240	319	390	383	387.4
c2	8	1938	1952	1990	11.3	253	361	385	391	392
r1	8	1795	1795	1827	12	239	785	381	381	387.9
r2	8	1795	1795	1827	12.3	236	785	380	380	388
rc1	8	1838	1838	1882	12.1	236	446	393	393	385.8
c1	10	1958	1958	1976	13.9	373	314	457	457	468.4
c2	10	2044	2044	-	13.7	385	-	472	472	-
r1	10	1908	1908	-	13.8	364	-	471	471	-
r2	10	1908	1908	-	13.8	371	-	471	471	-
rc1	10	1954	1954	-	13.8	337	-	486	486	-

It should be noted that the problems with 600 demand points and available 10 ambulances, exact algorithm could not find an optimum solution. In order to solve problem instances other than the SOLOMON's problem instances and to observe the run time performance of the hybrid heuristic, we also develop some specially designed problem instances with 1000 demand points and available 10 ambulances. The problem instances are called *D1*, *D2*, *D3* and *U* which stand, respectively, for design type 1, type 2, type 3 and uniformly distributed demand data. For each design type, 4 random problem instances are generated and runs are performed with only ADD heuristic and hybrid heuristic. Examples of design types solved by hybrid heuristics are shown in the following Figures 22, 23, 24 and 25.

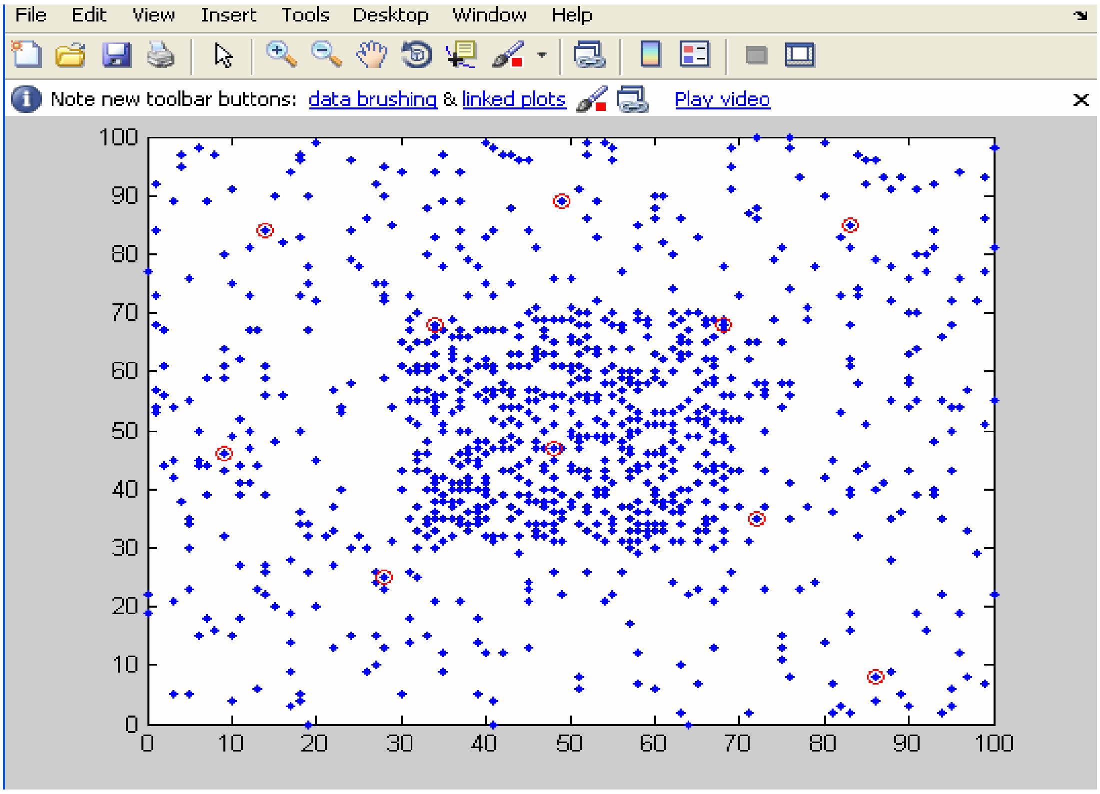


Figure 22 : Specially designed data, *D1,* solution with the hybrid heuristic

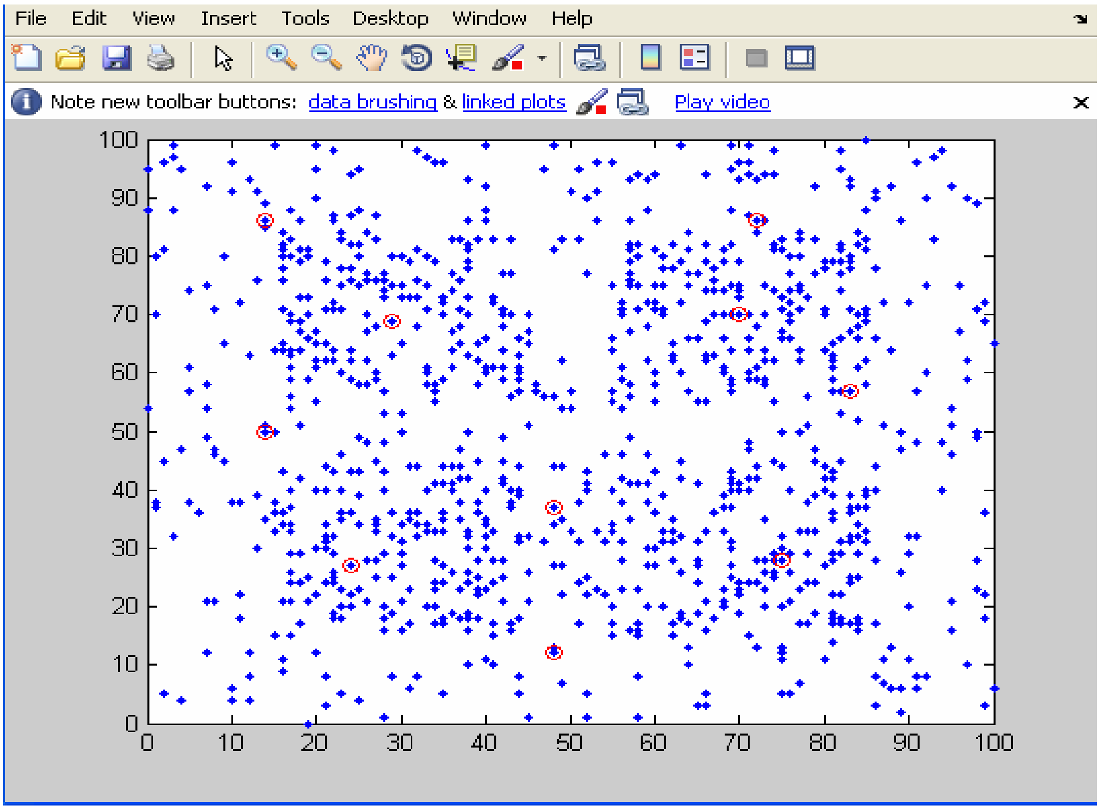


Figure 23 : Specially designed data, *D2,* solution with the hybrid heuristic

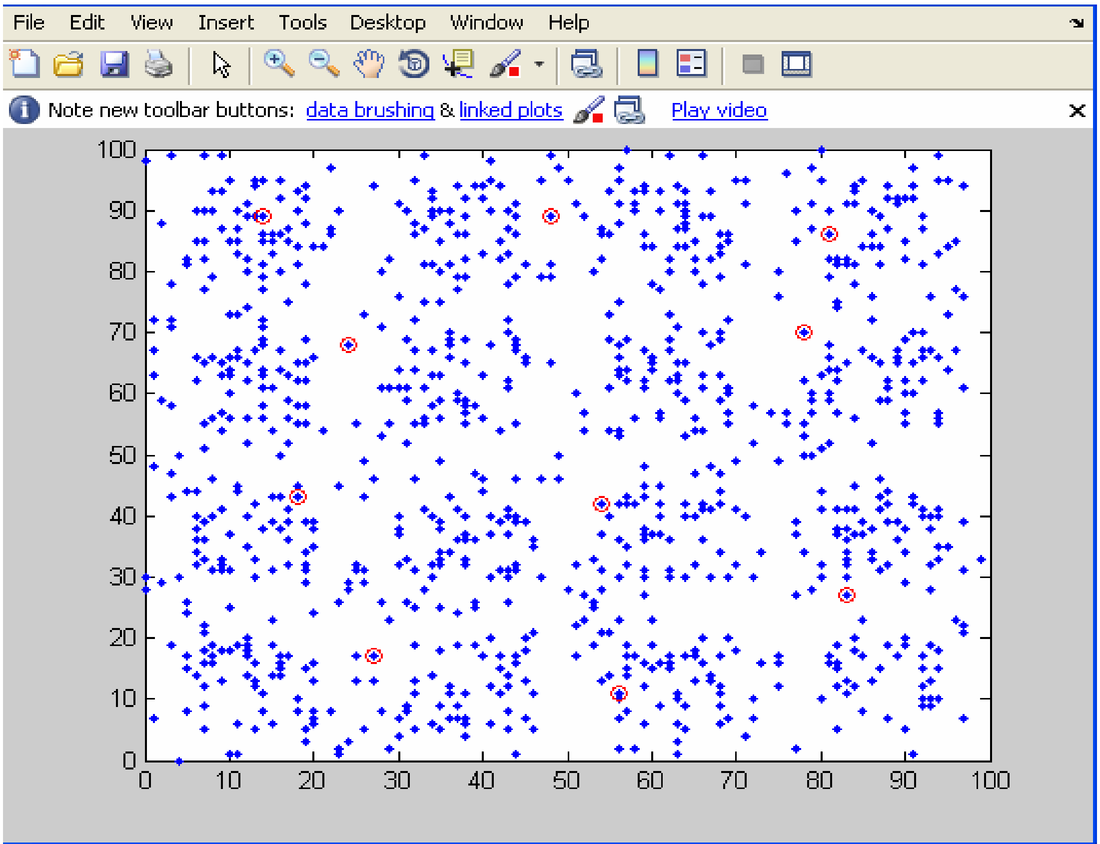


Figure 24 : Specially designed data, *D3,* solution with the hybrid heuristic

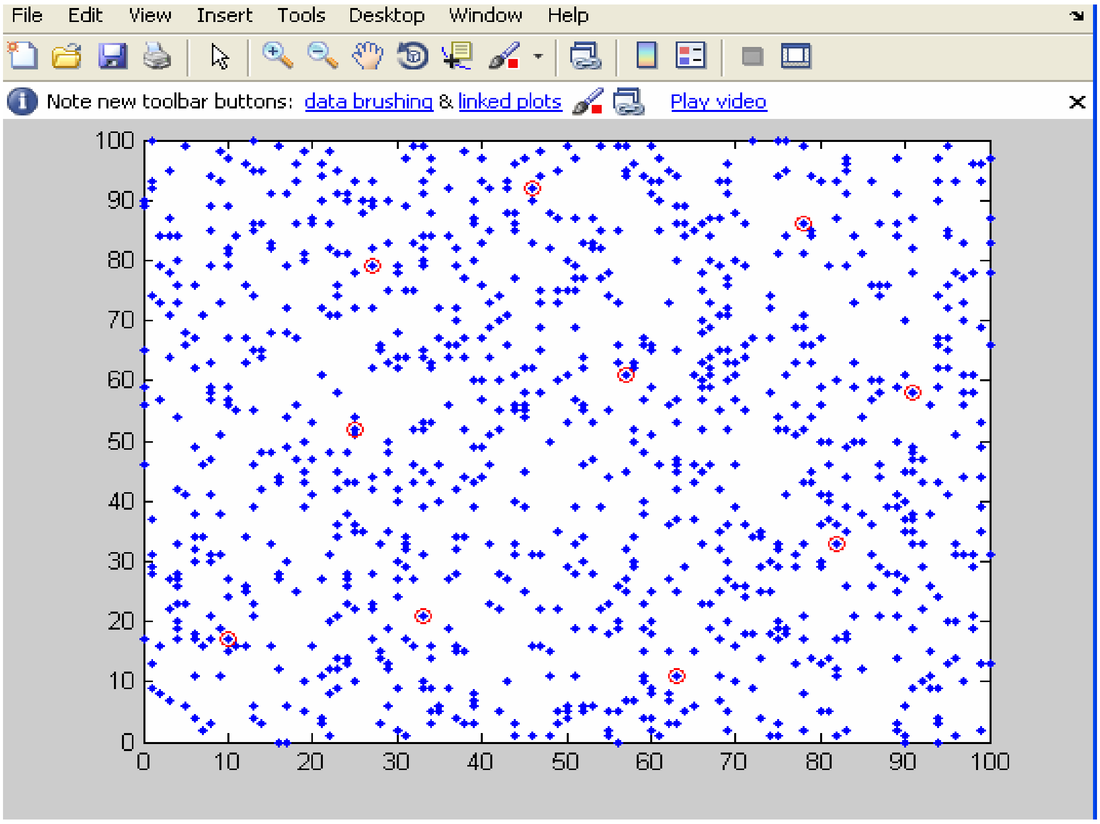


Figure 25 : Specially designed data, *U,* solution with the hybrid heuristic

The tabulated results of the specially designed data sets are shown in Table 23.

Table 23 : The results for the specially designed data sets *n*=1000

n =1000

		Objective		Run time		Preparedness	
problem	#of ambulance	ADD	Hybrid	ADD	Hybrid	ADD	Hybrid
D1	10	3279	3312	20.3	1108	1569	1596
D1	10	3297	3319	20.7	1041	1585	1623
D1	10	3285	3317	20.5	1037	1570	1576
D1	10	3291	3314	20.8	1005	1578	1571
D2	10	3247	3268	20.2	1003	1474	1455
D2	10	3240	3251	20.3	1003	1456	1506
D2	10	3260	3275	20.6	1006	1460	1462
D2	10	3252	3269	20.9	1009	1465	1468
D3	10	3138	3170	20.9	1002	1388	1379
D3	10	3120	3120	20.4	1000	1307	1307
D3	10	3119	3142	20.8	1004	1350	1344
D3	10	3148	3165	20.5	1017	1342	1335
U	10	3158	3158	20.7	1103	1338	1338
U	10	3137	3184	20.5	1033	1351	1331
U	10	3110	3157	20.9	1020	1329	1332
U	10	3125	3162	20.8	1041	1345	1348

Table 24 : Average objective function values of the specially designed data

average objective value results n=100, p=10	ADD	HYBRID	Percentage improvement by the HYBRID
D1	3,288.00	3,315.50	0.84%
D2	3,249.75	3,265.75	0.49%
D3	3,131.25	3,149.25	0.57%
U	3,132.50	3,165.25	1.05%

Both SOLOMON's data and specially designed problem instances reveal valuable insights for our study. First of all, when the demand points are clustered, the percentage improvement in the objective function values provided by the hybrid heuristic worsen. When the demand pattern is more uniformly distributed, the performance of the hybrid heuristic is much more outstanding.

Also, when the preparedness measure is defined differently, the obvious dominance of centralized solutions becomes to disappear; this might direct us towards the conclusion that dispersed solutions might produce even better preparedness values. However, for the previously studied problem instance where n=400 and p=10 (Figures 19 and 20), the preparedness measures still become 475 and 372, respectively, meaning that multiple coverage are still capable of producing higher preparedness index values. On the other hand, in further analysis, different preparedness definitions could be utilized, since the definitions for the concept are heavily subjective as mentioned in Andersson (2007).

CHAPTER 6

CONCLUSIONS AND DIRECTIONS FOR FUTURE STUDY

Briefly in this study, a multi-level coverage problem, which is an NP-hard problem, is addressed both by a mathematical model and some heuristic methods. The study contributes to the literature in terms of providing the comparison of fast solution techniques such as ADD-DROP and genetic heuristics for these multi-level coverage and partial coverage types of problems. The approach proposed is applicable to the ambulance location and relocation problems and also for any other strategic location issues such as recycle baskets and ATM machines. The hybrid heuristic that utilizes a greedy ADD heuristic at start and then advanced genetic algorithm with non-classical crossover technique is the most suitable solution technique for the large size problems, whereas, for small sized problems, exact algorithm could be used. The heuristic technique produces results nearly 5% close to the optimal value, and the performance of the heuristic could be increased by fine-tuning the genetic algorithm parameters. The future studies could focus on the performance improvement for the fast heuristic techniques.

Also, the preparedness index provided us another insight such that the multiple coverage approaches create more system preparedness compared to the multi-level coverage approach. However, it is observed that multiple coverage idea might fail in critical distance coverage. For this reason, it is suitable to use both concepts together in a sequential manner. At first, multi-level coverage problem can be solved; then in each responsible coverage field, multiple coverage concepts can be adopted.

REFERENCES

Altay N., Green W., OR/MS research in disaster operations management. European Journal of Operational Research, Vol.175. pp. 475-493, 2006.

Altman Z., Ben Jamaa S., Picard J.M., Fourestié B., Mourlon J. New challenges in automatic cell planning of UMTS networks. Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th, Vol. 2 pp. 951-954.

Andersson T., Varbrand P., Decision support tools for ambulance dispatch and relocation. Journal of Operational Research Society, 58. pp. 195-201, 2007.

Balçık B., Beamon, B.M., Facility Location in Humaniterian Relief. International Journal of Logistics, Vol 11, No:2, pp. 101-121, 2008.

Ball, M.O., Lin, L.F., A reliability model applied to emergency service vehicle location. Operations Research 41, 18-36, 1993.

Batta R., Dolan M. J., Krishnamurthy N. N. The Maximum Expected Covering Location Problem: Revisited. Transportation Science, Vol. 23. No.4, November 1989.

Brotcorne L., Laporte G., Semet F., Ambulance location and relocation models. European Journal of Operational Research 147, 451-463, 2003.

Chardaire, P., and Lutton, J. L. 1993, Using Simulated annealing to solve concentrator location problems in telecommunication networks, Applied Simulated Annealing, Springer, Berlin, pp. 175 – 199.

Church R., ReVelle C., The Maximal Covering Location Problem. Papers of the Regional Science Association, 32:101-120, 1974.

Church R., Gerrard R., The multi-level location set covering model. Geographical Analysis, 35, 2003.

Current J., Morton O., Locating Emergency Warning Sirens. Decision Sciences. Vol :3, pp.221-234, 1992.

Daskin, M.S., Stern, E.H., A hierarchical objective set covering model for emergency medical service vehicle deployment. Transportation Science 15, 137-152, 1981.

Daskin M.S., Maximum Expected Covering Location Model: Formulation, Properties and Heuristic Solution. Transportation Science, Vol. 17. No.1, pp. 48-70, February 1983.

Daskin M.S., Owen S.H., Strategic Facility Location: A review. European Journal of Operational Research 111, 423-447, 1998.

Denizel M., Usdiken B., Tuncalp D., Drift or Shift? Countinuity, change and international variation in knowledge production in OR/MS. Operations Reseacrh 51, pp. 711-720, 2003.

Dessouky M., Hongzhong J., Ordonez F., A modeling framework for facility location of medical services for large-scale emergencies. IEE transactions, Vol.39, pp.41-55, 2007.

Drezner Zvi, Erkut Erhan, Alp Osman., An Efficient Genetic Algorithm for the p-median Problem. Annals of Operations Research 122, 21-42, 2003.

Eaton, D.J., Daskin, M.S., Simmons,D. Bulloch, B., Jansma, G., Determining Emergency Medical Deployment in Austin, Texas. Interfaces 15, 1985

Erkut E., Ingolfsson A., Erdoğan Güneş. Ambulance Location for Maximum Survival. Naval Research Logistics, Vol. 55, 2008.

Gendreau M., Laporte G., Semet F., Solving an ambulance Location Problem by Tabu Search. Location Science, Vol.5, No.2 pp.75-88, 1997.

Goldberg B. J., Operations Research Models for the Deployment of Emergency Services Vehicles. EMS Management Journal, Vol.1 No.1, Jan-March 2004.

Gülden, B., A geographical information system application for ambulance routing services: A Prototype. Thesis submitted to Graduate School of Informatics of the Middle East Technical University. 2004.

Haghani A. and Oh S.C., Formulation and solution of a multi-commodity, multi-modal network flow model for disaster relief operations, Transportation Research Part A—Policy and Practice 30, 1996.

Hogan, K., ReVelle, C.S., Concepts and Applications of Backup Coverage. Management Science, 34, 1986.

Iannoni P. A., Morabito R., Saydam C., An optimization approach for ambulance location and the districting of the response segments on highways. European Journal of Operational Research, article in press, 2008

Ingolfsson A., Budge S., Erkut E., Optimal ambulance location with random delays and travel times. Health Care Mangement Science, 2007.

Jamaa, S.B. Altman, Z. Picard, J.M. Fourestie, B. Combined coverage and capacity optimisation for UMTS networks. Telecommunications Network

Strategy and Planning Symposium. NETWORKS 2004, 11th International 13-16 June 2004. pp. 175-178.

Karasakal, E., Karasakal O., A maximal covering location model in the presence of partial coverage, Computers and OR, 31, 2004.

Kuehn, A., and Hamburger, M. J. 1963, A heuristic Program for Locating Warehouses, Management Science, Vol. 9, pp. 643 – 666.

Kolesar P., Walker E., Warren. An algorithm for the dynamic relocation of fire companies. Operations Research, Vol.22, No.2, pp.249-274, 1974.

Laporte G., Gendreau M., Semet F., A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. Paralell Computing, Vol. 27, pp.1641-1653, May, 2001.

Larson C. R., A Hybercube Queuing Model for Facility Location and Redistricting in Urban Emergency Services. Comput. & Ops. Res. Vol.1, pp.67-95, 1974.

Larson C. R., Approximating the Performance of Urban Emergency Service Systems. Operations Research, Vol. 23. No.5, September-October 1975.

Lightner C., Tavakoli A., Implementing a mathematical model for locating EMS vehicles in Fayetteville, NC. Computers and OR, 31, 2004.

Mandell, M.B., Covering models for two-tiered emergency medical systems. Location Science 6, 355-368, 1998.

Marinov, V. ReVelle, C.S., Siting Emergency Services. In : Drezner, Z. (Ed.), Facility Location. Springer, New York. 1995

Marinov, V. ReVelle, C.S., The capacitated standard response fire protection siting problem: deterministic and probabilistic models. Annals of Operations Research 40, 1992.

Marinov, V. ReVelle, C.S., The queueing probabilistic location set covering problem and some extentions. Socio Economic Planning Sciences 28, 1994.

Morabito R., Galvao D. R., Chiyoshi Y.F., A note on solutions to the maximal expected covering location problem. Computers and Operations Research 30, 87-96, 2002.

Morabito R., Galvao D. R., Chiyoshi Y.F., Towards unified formulations and extensions of the two classical probabilistic location models. Computers and Operations Research 32, 15-33, 2005.

Morabito R., Galvao D. R., Chiyoshi Y.F., Non-homogeneous servers in emergency medical systems: Practical applications using the hypercuve queuing model. Socio-Economic Planning Sciences, 2007.

Nayeem Abu SK. MD. Pal Madhumangal. Genetic algorithm to solve the p-centre and p-radius problem on a network. International Journal of Computer Mathematics, Vol.82, No.5, pp.541-550, 2005.

Özdamar L., Ekinci E., Kücükyazıcı B. Emergency logistics planning in natural disasters, Annals of Operations Research 129, 2004.

Özgönenç H. A genetic algorithm for the location-routing problem with time windows. Thesis submitted to Graduate School of Natural and Applied Sciences of the Middle East Technical University. 2006.

Peleg K., Pliskin S. J. A geographic information system simulation model of EMS: Reducing ambulance response time. American Journal of Emergency Medicine, Vol.22 No.3, May, 2004

ReVelle C., Toregas C., Swain R., Bergman L. The Location of Emergency Service Facilities. Operations Research 19(6), 1363-1373, 1971.

ReVelle C., Hogan, K., The maximum availability location problem. Transportation Science 23 192-200, 1989.

Richards, D., Optimised Ambulance Redeployment Strategies, Masters Thesis, University of Auckland, 2007.

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

Rajagopalan K. H., Saydam C., Xiao J., A multiperiod set covering location model for dynamic redeployment of ambulances. Computers & Operations Research 35, 814-826, 2008.

Repede, J.F., Bernardo, J.J., Developing and validating a decision support system for locating emergency medical vehicles in Louisville, Kentucky. European Journal of Operations Research 75, 567-581, 1994.

Saydam C., Aytuğ H., Solving large-scale maximum expected covering location problems by genetic algorithms: A comparative study. European Journal of Operational Research 141, 2002.

Saydam C., Aytuğ H., Accurate estimation of expected coverage: revisited. Socio Economic Planning Sciences 37, 69-80, 2003.

Schilling, D.A., Elzinga, D.J., Cohon, J., Church, R.L., ReVelle, C.S., The TEAM/FLEET models for simultaneousfacility and equipment sitting. Transportation Science 13, 1979.

Singer M., Donoso P., Assesing an ambulance service with queing theory. Computers & Operations Research 35, 2549-2560, 2008.

Viswanath K., Peeta S., The multicommodity maximal network design problem. The IEEE 5th International Conference on Intelligent Transportation Systems. Singapore, 3-6 September 2002.

Yamada T., A network flow approach to a city emergency evacuation planning, International Journal of Systems Science 27, 1996.

APPENDIX A

GAMS CODING FOR THE EXACT ALGORITHM

```
scalars
p total number of ambulances /5/
;
sets
i set of demand points /1*400/
k duplicate of set of demand points /1*400/
j set of potential ambulance sites  /1*400/
dummy dummy scalar for distanceXY table /1,2/
s number of sectors /1*3/
;

parameters
lambda(i) demand at point i /set.i 1/
w(s) weight for each sector /1 2, 2 1, 3 0.5/
;
parameter DistanceXYi (j,dummy)
        DistanceXYk (k,dummy);

$libinclude xlimport DistanceXYi Coordinates200.xls Koordinatlar!a1:c201
$libinclude xlimport DistanceXYk Coordinates200.xls Koordinatlar!a1:c201

parameter d;
d(j,k) = sqrt((abs(DistanceXYi(j,"1")-DistanceXYk(k,"1")))**2 +
(abs(DistanceXYi(j,"2")-DistanceXYk(k,"2")))**2);
parameter maxim;
maxim = smax((i,k),d(i,k));
parameters r(s);
r("3") = maxim/2;
r("2")= r("3")/2;
r("1")=r("2")/2;

binary variables x, y ;
variable Z;


equations Objective
        Equation_2
        Equation_3(i,j,s)
```

```
        Equation_6(i,j)
        Equation_7(i,j)
        Equation_8_1(i)
        Equation_8_2(i)
        Equation_8_3(i)
        Equation_9(i,j,s)

;

Objective..                       Z=e=sum(s,sum(i,sum(j$(d(i,j)le
r(s)),w(s)*lambda(i)*x(i,j,s))));
Equation_2..                       sum(j,y(j)) =e= p;
Equation_3(i,j,s) $(d(i,j)le r(s))..     x(i,j,s) =l= y(j);
Equation_6(i,j)..                    x(i,j,"1")=l=x(i,j,"2");
Equation_7(i,j)..                    x(i,j,"2")=l=x(i,j,"3");
Equation_8_1(i)..                     sum(j$(d(i,j)le r("1")),x(i,j,"1")) =l= 1;
Equation_8_2(i)..                     sum(j$(d(i,j)le r("2")),x(i,j,"2")) =l= 1;
Equation_8_3(i)..                     sum(j$(d(i,j)le r("3")),x(i,j,"3")) =l= 1;
Equation_9(i,j,s)..                  x(i,j,"3")$(d(i,j)gt r("3"))=e=0;

model mlc /all/;
file results_200 /mlc_results/
option iterlim = 10000000;
*option limrow=100000000;
*option limcol=100000000;
option mip=cplex;
*option lp=cplex;
solve mlc maximizing Z using mip;

*$libinclude xlexport y.m myspread.xls output5!a1..a1
scalar preparedness /0/;
loop((j)$( y.l(j) gt 0), loop(k $((d(j,k) gt 0) AND (d(j,k) lt r("3") )), preparedness =
preparedness + 1/d(j,k)));

put results_200 ;
put "---------mlc Results---------------"/
put "--------- X Values---------------"/
loop((i,j,s)$(x.l(i,j,s) gt 0), put i.tl, @4, j.tl, @7, s.tl, @14, x.l(i,j,s):0:19 /)  ;
put "--------- Y Values---------------"/
loop((j)$(y.l(j) gt 0), put j.tl, @14, y.l(j):0:19 /)  ;
put "---------Z Value------------"//;
put "Z   ---> " Z.l:0:19 /  ;
put "---------preparedness Value------------"//;
put "Z   ---> " preparedness:0:100 /  ;
put "-----------------------------------------------------"//;
display d, maxim, r, y.l, x.l, preparedness;
```

APPENDIX B

MATLAB CODING FOR GREEDY ADD HEURISTIC

B.1 : Main Body

```
tic
global cities;
global distances;
cities = 200;
locations = zeros(cities,2);
dummy_locations = zeros(cities,1);  n = 1;

while (n <= cities)
   xp = int8(rand*100);
   yp = int8(rand*100);
    locations(n,1) = xp;
    locations(n,2) = yp;
    dummy_locations(n,1)=n;
    n = n+1;
end

dummy_locations=horzcat(dummy_locations,locations);
dummy_locations=vertcat([str2double(' ') 1
2],dummy_locations);

xls_file = strcat ('C:\Documents and Settings\mesut\My
Documents\gamsdir\projdir\Coordinates',num2str(cities),'.
xls');
xlswrite(xls_file ,dummy_locations,'Koordinatlar');
%% mesut

distances = zeros(cities);
for count1=1:cities,
    for count2=1:count1,
        x1 = locations(count1,1);
        y1 = locations(count1,2);
        x2 = locations(count2,1);
        y2 = locations(count2,2);
        distances(count1,count2)=sqrt((x1-x2)^2+(y1-
y2)^2);

distances(count2,count1)=distances(count1,count2);
```

```
    end;
end;

max_distance = max(max(distances))
R3 = max_distance/2
R2 = R3/2
R1 = R2/2
ambulans_sayisi = 10;
eklenen_indeksler = zeros(1,cities-ambulans_sayisi);
eklenen_indeks_sayisi = 0;
ambulansli_indeksler = [];
kalan_indeksler = 1:cities;

for j = 1:ambulans_sayisi
    scores = zeros(1,cities-j+1);
    for m = 1 : cities-j+1
        p = [ambulansli_indeksler kalan_indeksler(m)];
        scores(m) = add_fitness(R3, R2, R1, p, j);
    end

    [min_value eklenen_indeks]=min(scores);
    eklenen_indeksler(j) = eklenen_indeks;
    eklenen_indeks_sayisi = eklenen_indeks_sayisi+1;
    yeni_indeksler = zeros(1, j);
    yeni_indeksler(1:j-1)=ambulansli_indeksler;
    yeni_indeksler(j) = eklenen_indeks;
    ambulansli_indeksler = yeni_indeksler
    tmp_kalan_indeksler =
zeros(1,length(kalan_indeksler)-1);
    t=1;
    for r=1:length(kalan_indeksler)
        if (r ~= eklenen_indeks)
            tmp_kalan_indeksler(t) = kalan_indeksler(r);
            t = t+1;
        end
    end
    kalan_indeksler = tmp_kalan_indeksler;

    length(ambulansli_indeksler)
end

ambulansli_indeksler
ambulans_sayisi = length(ambulansli_indeksler);
f=0;
p=ambulansli_indeksler;
min_score = add_fitness(R3, R2, R1, p, ambulans_sayisi)

figure;
plot(locations(:,1),locations(:,2),'b.');
```

```
for i=1:ambulans_sayisi
    hold on;
    plot(locations(ambulansli_indeksler(i),1),
locations(ambulansli_indeksler(i),2), 'ro');
end

cozum = ambulansli_indeksler


[n3 n2 n1] = coverage_numbers(R3, R2, R1, cozum,
ambulans_sayisi)
R3_COVERAGE = n3
R2_COVERAGE = n2
R1_COVERAGE = n1


coverage_matrix = find_coverage_matrix(R3, R2, R1, cozum,
ambulans_sayisi);

katsayi = find_katsayi(R3, cozum, ambulans_sayisi);
save_filename='mesut.mat';
save (save_filename,'locations');
toc
```

B.2 : Objective Evaluation

```
function [score] = add_fitness(R3, R2, R1, p, j)

global cities;
global distances;

f=0;
for i = 1:cities
    tmp = zeros(1,3);
    n1 = 0;
    n2 = 0;
    n3 = 0;
    for k = 1 : j
        if (distances(i, p(k))<R3)
            tmp(1) = 1;
            n1 = 0.5;
        end
        if (distances(i, p(k))<R2)
            tmp(2)=1;
            n2 = 1;
```

```
        end
        if (distances(i, p(k))<R1)
            tmp(3)=1;
            n3 = 2;
        end
    end

    f = f + ( -1 * (n1*tmp(1)+n2*tmp(2)+n3*tmp(3))) ;
end

 score = f;
```

B.3 : Preparedness index calculation

```
function    [katsayi]    =    find_katsayi(R3,        p,
ambulans_sayisi)

global cities;
global distances;

katsayi = 0;
for i = 1:cities
        for k = 1 : ambulans_sayisi
            if (distances(i, p(k))<R3 && distances(i,
p(k)) ~= 0)
                katsayi = katsayi + 1 / distances(i,
p(k));
            end
        end

end

katsayi= katsayi / 1e4;
```

APPENDIX C

MATLAB CODING FOR GREEDY DROP HEURISTIC

C.1 : Main Body

```
tic
global cities;
global distances;
cities = 400;
locations = zeros(cities,2);

n = 1;
load mesut.mat

distances = zeros(cities);
for count1=1:cities,
    for count2=1:count1,
        x1 = locations(count1,1);
        y1 = locations(count1,2);
        x2 = locations(count2,1);
        y2 = locations(count2,2);
        distances(count1,count2)=sqrt((x1-x2)^2+(y1-
y2)^2);

distances(count2,count1)=distances(count1,count2);
    end;
end;

max_distance = max(max(distances))
R3 = max_distance/2
R2 = R3/2
R1 = R2/2
ambulans_sayisi = 10;


cikartilan_indeksler = zeros(1,cities-ambulans_sayisi);
cikartilan_indeks_sayisi = 0;
mevcut_indeksler = 1:cities;
for j = 1:(cities-ambulans_sayisi)
    scores = zeros(1,length(mevcut_indeksler));
    for m = 1 : length(mevcut_indeksler)
```

```
        tmp_ambulans_sayisi = length(mevcut_indeksler)-1;

        p = [];

        if (m == 1)
            p =
mevcut_indeksler(2:length(mevcut_indeksler));
        elseif (m == length(mevcut_indeksler))
            p=
mevcut_indeksler(1:length(mevcut_indeksler)-1);

        else
            p = [mevcut_indeksler(1:m-1)
mevcut_indeksler(m+1:length(mevcut_indeksler))];
        end

        scores(m) =
drop_fitness(R3,R2,R1,p,tmp_ambulans_sayisi);
    end

    [max_val cikartilan_indeks]=min(scores);
    cikartilan_indeksler(j) = cikartilan_indeks;
    cikartilan_indeks_sayisi =
cikartilan_indeks_sayisi+1;
    yeni_indeksler = zeros(1, cities-
cikartilan_indeks_sayisi);
    t=1;
    for r=1:length(mevcut_indeksler)
        if (r ~= cikartilan_indeks)
            yeni_indeksler(t) = mevcut_indeksler(r);
            t = t+1;
        end
    end
    mevcut_indeksler = yeni_indeksler;
    length(mevcut_indeksler)
end

mevcut_indeksler
tmp_ambulans_sayisi = length(mevcut_indeksler);
p = mevcut_indeksler;
min_score = drop_fitness(R3, R2, R1, p,
tmp_ambulans_sayisi)

figure;
plot(locations(:,1),locations(:,2),'b.');


for i=1:ambulans_sayisi
```

```
    hold on;
    plot(locations(mevcut_indeksler(i),1),
locations(mevcut_indeksler(i),2), 'ro');
end

cozum = mevcut_indeksler

[n3 n2 n1] = coverage_numbers(R3, R2, R1, cozum,
ambulans_sayisi);
R3_COVERAGE = n3
R2_COVERAGE = n2
R1_COVERAGE = n1
coverage_matrix = find_coverage_matrix(R3, R2, R1, cozum,
ambulans_sayisi);

katsayi = find_katsayi(R3, cozum, ambulans_sayisi)
toc
```

C.2 : Objective Evaluation

```
function [score] = drop_fitness(R3, R2, R1, p, j)

global cities;
global distances;

f=0;
for i = 1:cities
    tmp = zeros(1,3);
    n1 = 0;
    n2 = 0;
    n3 = 0;
    for k = 1 : j
        if (distances(i, p(k))<R3)
            tmp(1) = 1;
            n1 = 0.5;
        end
        if (distances(i, p(k))<R2)
            tmp(2)=1;
            n2 = 1;
        end
        if (distances(i, p(k))<R1)
            tmp(3)=1;
            n3 = 2;
        end
    end

    f = f + ( -1 * (n1*tmp(1)+n2*tmp(2)+n3*tmp(3))) ;
```

```
end


 score = f;
```

C.3 : Preparedness index calculation

```
function [katsayi] = find_katsayi(R3,  p,
ambulans_sayisi)

global cities;
global distances;

katsayi = 0;
for i = 1:cities
        for k = 1 : ambulans_sayisi
            if (distances(i, p(k))<R3 && distances(i,
p(k)) ~= 0)
                katsayi = katsayi + 1 / distances(i,
p(k));
            end
        end

end

katsayi= katsayi / 1e4;
```

APPENDIX D

MATLAB CODING FOR GA1 HEURISTIC

D.1 : Main Body

```
tic

global cities;
global distances;

cities = 400;
locations = zeros(cities,2);
dummy_locations = zeros(cities,1);

n = 1;
load mesut.mat

dummy_locations=horzcat(dummy_locations,locations);
dummy_locations=vertcat([str2double(' ') 1
2],dummy_locations);

xls_file = strcat ('C:\Documents and Settings\mesut\My
Documents\gamsdir\projdir\Coordinates',num2str(cities),'.
xls');
xlswrite(xls_file ,dummy_locations,'Koordinatlar');

distances = zeros(cities);
for count1=1:cities,
    for count2=1:count1,
        x1 = locations(count1,1);
        y1 = locations(count1,2);
        x2 = locations(count2,1);
        y2 = locations(count2,2);
        distances(count1,count2)=sqrt((x1-x2)^2+(y1-
y2)^2);

distances(count2,count1)=distances(count1,count2);
    end;
end;


max_distance = max(max(distances))
```

```
R3 = max_distance/2
R2 = R3/2
R1 = R2/2
ambulans_sayisi = 8;
FitnessFcn = @(x) ambulans_fitness(x,distances, R3, R2,
R1, ...
    ambulans_sayisi,cities);


my_plot = @(options,state,flag) ambulans_plot(options,
...
    state,flag,locations);


options = gaoptimset('PopulationType',
'custom','PopInitRange', ...
    [1;cities]);

options =
gaoptimset(options,'CreationFcn',@create_ambulans_permuta
tions, ...
    'CrossoverFcn',@crossover_ambulans_permutation, ...
    'MutationFcn',@mutate_ambulans_permutation, ...
    'PlotFcn', my_plot, ...
    'Generations',300,'PopulationSize',75, ...
    'StallGenLimit',50,'Vectorized','on');



numberOfVariables = ambulans_sayisi;
[x,fval,reason,output] =
ga(FitnessFcn,numberOfVariables,options)
figure;
plot(locations(:,1),locations(:,2),'b.');


for i=1:ambulans_sayisi
    hold on;
    plot(locations(x{1}(i),1), locations(x{1}(i),2),
'ro');
end

cozum = x{1}
[n3 n2 n1] = coverage_numbers(R3, R2, R1, cozum,
ambulans_sayisi);
R3_COVERAGE = n3
R2_COVERAGE = n2
R1_COVERAGE = n1
```

```
coverage_matrix = find_coverage_matrix(R3, R2, R1, cozum,
ambulans_sayisi);


katsayi = find_katsayi(R3, cozum, ambulans_sayisi)
toc
```

D.2 : Fitness Evaluation

```
function scores = ambulans_fitness(x,distances, R3, R2,
R1, ambulans_sayisi, cities)

scores = zeros(size(x,1),1);
for j = 1:size(x,1)

    p = x{j};

    f = 0;
    for i = 1:cities
        tmp = zeros(1,ambulans_sayisi);
        n1 = 0;
        n2 = 0;
        n3 = 0;
        for k = 1 : ambulans_sayisi
            if (distances(i, p(k))<R3)
                tmp(1) =1;
                n1 = 0.5;
            end
            if (distances(i, p(k))<R2)
                tmp(2)=1;
                n2 = 1;
            end
            if (distances(i, p(k))<R1)
                tmp(3)=1;
                n3 = 2;
            end
        end

         f = f + ( -1 * (n1*tmp(1)+n2*tmp(2)+n3*tmp(3)))
;


    end
    scores(j) = f;
end
[min_score,index] = min(scores)
```

```
noktalar = x{index}
max_score = max(scores)
```

D.3 : Initial Population Generation

```
function    pop    =    create_ambulans_permutations(NVARS,
FitnessFcn,options)
global cities;
totalPopulationSize = sum(options.PopulationSize);
n = NVARS;
pop = cell(totalPopulationSize,1);
for i = 1:totalPopulationSize
    a = randperm(cities);
    pop{i} = a(1:n);
end
```

D.4 : Crossover Technique

```
function xoverKids  =
crossover_ambulans_permutation(parents,options,NVARS, ...
    FitnessFcn,thisScore,thisPopulation)
global cities;
nKids = length(parents)/2;
xoverKids = cell(nKids,1);
index = 1;

for i=1:nKids

    parent = thisPopulation{parents(index)};
    parent2 = thisPopulation{parents(index+1)};
    index = index + 2;

    p1 = ceil((length(parent))/2)+1;

    tmp = parent2(p1:length(parent2));


    for j=p1:length(parent)
        for k=1:(p1-1)
            if (parent2(j)==parent(k))
                tmp(j-p1+1) = tmp(j-p1+1)+1;
                if (tmp(j-p1+1)== cities+1)
```

```
                    tmp(j-p1+1)=1;
                end
            end
        end
    end


    parent(p1:length(parent)) = tmp;

    child = parent;

    xoverKids{i} = child;

end
```

D.5 : Preparedness index calculation

```
function [katsayi] = find_katsayi(R3,  p,
ambulans_sayisi)

global cities;
global distances;

katsayi = 0;
for i = 1:cities
        for k = 1 : ambulans_sayisi
            if (distances(i, p(k))<R3 && distances(i,
p(k)) ~= 0)
                katsayi = katsayi + 1 / distances(i,
p(k));
            end
        end

end

katsayi= katsayi / 1e4;
```

APPENDIX E

MATLAB CODING FOR GA2 HEURISTIC

E.1 : Main Body

```
tic

global cities;
global distances;
global R3;
global R2;
global R1;

cities = 400;
locations = zeros(cities,2);

n = 1;
load mesut.mat

distances = zeros(cities);
for count1=1:cities,
    for count2=1:count1,
        x1 = locations(count1,1);
        y1 = locations(count1,2);
        x2 = locations(count2,1);
        y2 = locations(count2,2);
        distances(count1,count2)=sqrt((x1-x2)^2+(y1-
y2)^2);

distances(count2,count1)=distances(count1,count2);
    end;
end;


max_distance = max(max(distances))

R3 = max_distance/2
R2 = R3/2
R1 = R2/2
ambulans_sayisi = 8;
```

```
FitnessFcn = @(x) ambulans_fitness(x,distances, R3, R2,
R1, ...
    ambulans_sayisi,cities);


my_plot = @(options,state,flag) ambulans_plot(options,
...
    state,flag,locations);


options = gaoptimset('PopulationType',
'custom','PopInitRange', ...
    [1;cities]);

options =
gaoptimset(options,'CreationFcn',@create_ambulans_permuta
tions, ...
    'CrossoverFcn',@crossover_ambulans_permutation, ...
    'MutationFcn',@mutate_ambulans_permutation, ...
    'PlotFcn', my_plot, ...
    'Elitecount',6, ...
    'Crossoverfraction',0.9, ...
    'SelectionFcn', @selectionuniform, ...
    'Generations',500,'PopulationSize',150, ...
    'StallGenLimit',100,'Vectorized','on');



numberOfVariables = ambulans_sayisi;
[x,fval,reason,output] =
ga(FitnessFcn,numberOfVariables,options)
figure;
plot(locations(:,1),locations(:,2),'b.');


for i=1:ambulans_sayisi
    hold on;
    plot(locations(x{1}(i),1), locations(x{1}(i),2),
'ro');
end

cozum = x{1}
[n3 n2 n1] = coverage_numbers(R3, R2, R1, cozum,
ambulans_sayisi);
R3_COVERAGE = n3
R2_COVERAGE = n2
R1_COVERAGE = n1
```

```
coverage_matrix = find_coverage_matrix(R3, R2, R1, cozum,
ambulans_sayisi);


katsayi = find_katsayi(R3, cozum, ambulans_sayisi)
toc
```

E.2 : Fitness Evaluation

```
function scores = ambulans_fitness(x,distances, R3, R2,
R1, ambulans_sayisi, cities)

scores = zeros(size(x,1),1);
for j = 1:size(x,1)
    p = x{j};
        f = 0;
    for i = 1:cities
        tmp = zeros(1,ambulans_sayisi);
        n1 = 0;
        n2 = 0;
        n3 = 0;
        for k = 1 : ambulans_sayisi
            if (distances(i, p(k))<R3)
                tmp(1) =1;
                n1 = 0.5;
            end
            if (distances(i, p(k))<R2)
                tmp(2)=1;
                n2 = 1;
            end
            if (distances(i, p(k))<R1)
                tmp(3)=1;
                n3 = 2;
            end
        end

         f = f + ( -1 * (n1*tmp(1)+n2*tmp(2)+n3*tmp(3)))
;


    end
    scores(j) = f;
end
[min_score,index] = min(scores)
noktalar = x{index}
max_score = max(scores)
```

E.3 : Initial Population Generation

```
function    pop    =    create_ambulans_permutations(NVARS,
FitnessFcn,options)
global cities;
totalPopulationSize = sum(options.PopulationSize);
n = NVARS;
pop = cell(totalPopulationSize,1);
for i = 1:totalPopulationSize
    a = randperm(cities);
    pop{i} = a(1:n);
end
```

E.4 : Crossover Technique

```
function xoverKids  =
crossover_ambulans_permutation(parents,options,NVARS, ...
    FitnessFcn,thisScore,thisPopulation)
global cities;
global R3;
global R2;
global R1;

nKids = length(parents)/2;
xoverKids = cell(nKids,1);
index = 1;


for i=1:nKids
    parent = thisPopulation{parents(index)};
    parent2 = thisPopulation{parents(index+1)};
    index = index + 2;

    p1 = ceil((length(parent))/2)+1;

    tmp = parent2(p1:length(parent2));

    yeni_uzunluk = (length(parent2)-
p1+1)+length(parent2);


    for j=p1:length(parent)
```

```
            for k=1:length(parent)
                if (parent2(j)==parent(k))
                    tmp(j-p1+1) = tmp(j-p1+1)+1;
                    if (tmp(j-p1+1)== cities+1)
                        tmp(j-p1+1)=1;
                    end
                end
            end
        end

        tmp2 = [parent tmp];
        tmp3 = [parent(p1:length(parent)) tmp];
        cikartilan_indeks_sayisi = 0;
        for k = 1:(length(tmp2)-p1+1)/2
            scores = zeros(1,length(tmp3));
            for  m = 1:length(tmp3)

                p = [];

                if (m == 1)
                    p = tmp3(2:length(tmp3));
                elseif (m == length(tmp3))
                    p= tmp3(1:length(tmp3)-1);
                else
                    p = [tmp3(1:m-1) tmp3(m+1:length(tmp3))];
                end
                p2 = [parent(1:p1-1) p];
                scores(m) =
    drop_fitness(R3,R2,R1,p2,length(p2));
            end

            [max_val cikartilan_indeks]=min(scores);
            cikartilan_indeks_sayisi =
    cikartilan_indeks_sayisi+1;
            yeni_indeksler = zeros(1, length(tmp3)-1);
            t=1;
            for r=1:length(tmp3)
                if (r ~= cikartilan_indeks)
                    yeni_indeksler(t) = tmp3(r);
                    t = t+1;
                end
            end
            tmp3 = yeni_indeksler;
            length(tmp3)
        end


        parent = [parent(1:p1-1) tmp3];
        child = parent;
```

```
    xoverKids{i} = child;

end
```

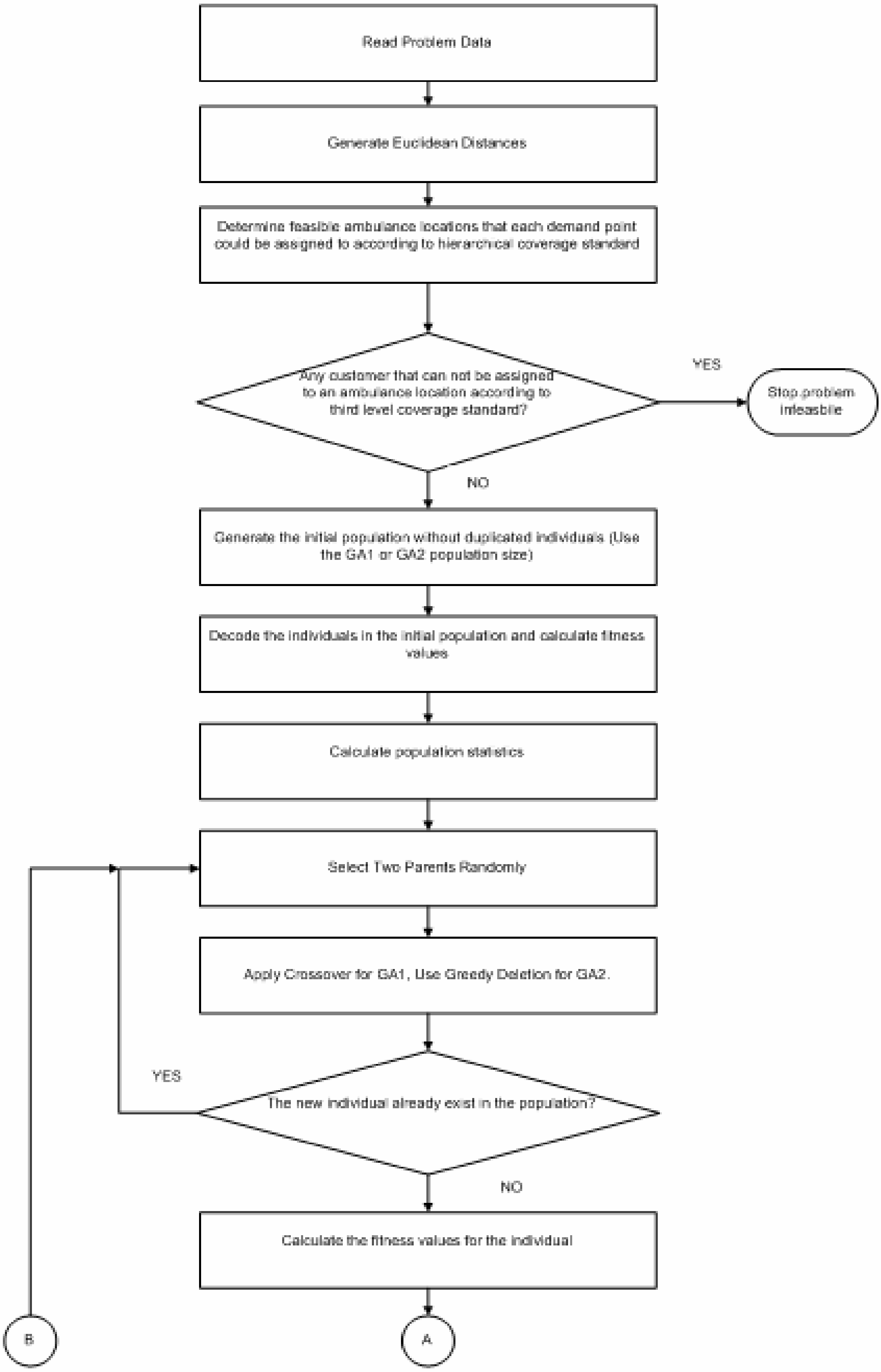
E.5 : Preparedness index calculation

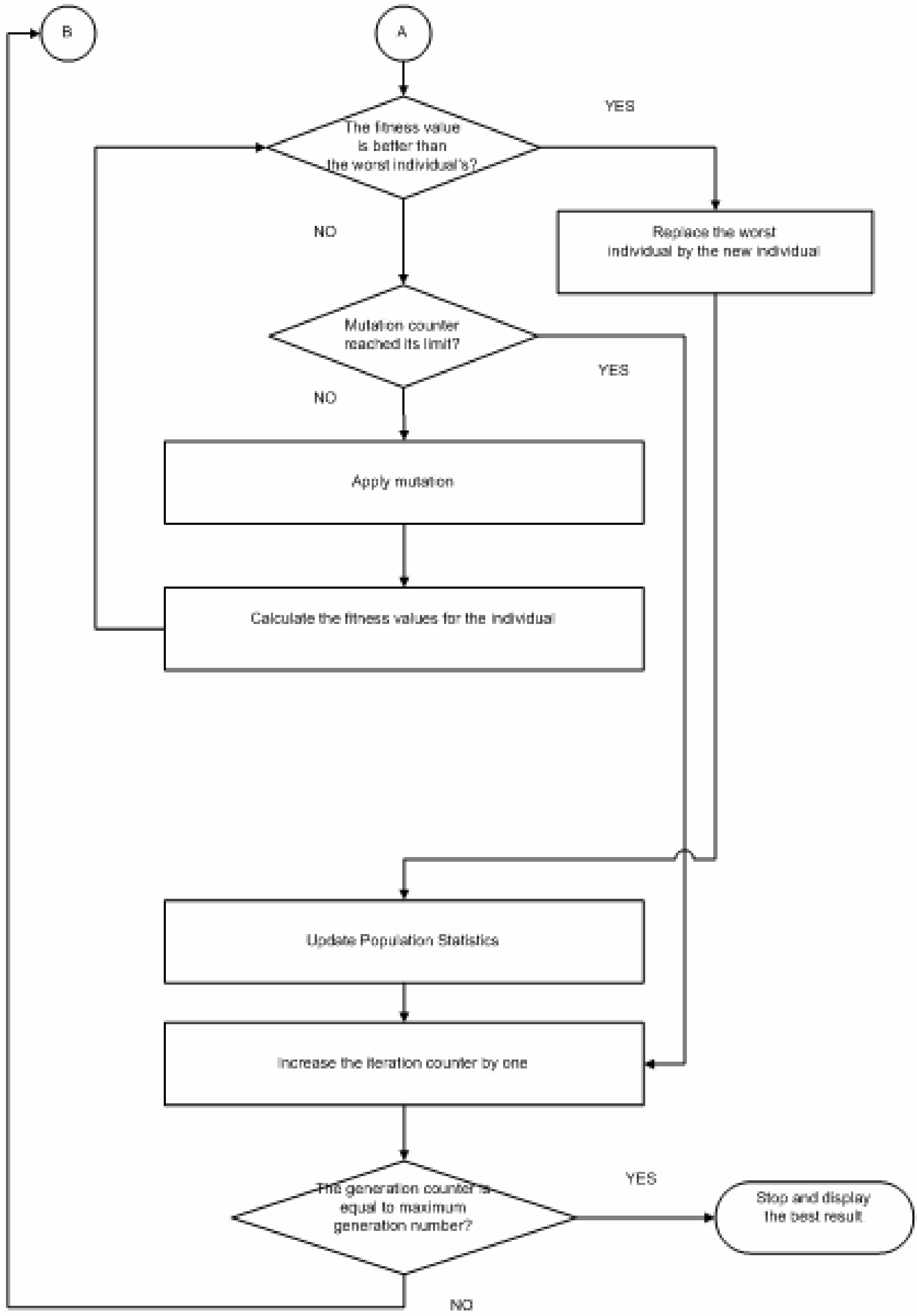
```
function [katsayi] = find_katsayi(R3,  p,
ambulans_sayisi)

global cities;
global distances;

katsayi = 0;
for i = 1:cities
        for k = 1 : ambulans_sayisi
            if (distances(i, p(k))<R3 && distances(i,
p(k)) ~= 0)
                katsayi = katsayi + 1 / distances(i,
p(k));
            end
        end

end

katsayi= katsayi / 1e4;
```

APPENDIX F

FLOWCHART of THE GAs

Starting from the next page.


Read Problem Data
Generate Euclidean Distances
Determine feasible ambulance locations that each demand point could be assigned to according to hierarchical coverage standard
Any customer that can not be assigned to an ambulance location according to third level coverage standard?
YES
Stop.problem infeasbile
NO
Generate the initial population without duplicated individuals (Use the GA1 or GA2 population size)
Decode the individuals in the initial population and calculate fitness values
Calculate population statistics
Select Two Parents Randomly
Apply Crossover for GA1, Use Greedy Deletion for GA2.
YES
The new individual already exist in the population?
NO
Calculate the fitness values for the individual
B
A


B
A
The fitness value
is better than
the worst individual's?
YES
NO
Replace the worst
individual by the new individual
Mutation counter
reached its limit?
YES
NO
Apply mutation
Calculate the fitness values for the individual
Update Population Statistics
Increase the iteration counter by one
The generation counter is
equal to maximum
generation number?
YES
Stop and display
the best result
NO

APPENDIX G

THE MINITAB OUTPUT

Estimated Effects and Coefficients for obj. (coded units)

Term	Effect	Coef	SE Coef	T	P
Constant		1170,33	5,526	211,77	0,000
Beg. Population size	32,00	16,00	5,526	2,90	0,020
termination condition	4,33	2,17	5,526	0,39	0,705
Beg. Population size* termination condition	6,33	3,17	5,526	0,57	0,582

S = 19,1442   PRESS = 6597
R-Sq = 52,56%   R-Sq(pred) = 0,00%   R-Sq(adj) = 34,77%

Analysis of Variance for obj. (coded units)

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Main Effects	2	3128,3	3128,3	1564,2	4,27	0,055
2-Way Interactions	1	120,3	120,3	120,3	0,33	0,582
Residual Error	8	2932,0	2932,0	366,5		
Pure Error	8	2932,0	2932,0	366,5		
Total	11	6180,7				

Estimated Coefficients for obj. using data in uncoded units

Term	Coef
Constant	1150,83
Beg. Population size	0,130000
termination condition	-8,3333
Beg. Population size* termination condition	0,126667