# Ambulance Allocation Using GIS

Christian Krog Lindeskov – Master Thesis
Technical University of Denmark – 30th of August 2002

# Abstract

This report deals with the topic of combining Operations Research (OR) and Geographic Information Systems (GIS) to improve the ambulance service of Falck A/S, the leading provider of ambulance services in Denmark.

Using the GIS software ArcView 3.2a (ESRI) the geographic occurrence of accidents over time and three different location allocation models are investigated. Two models on minimizing the average response time and one model on minimizing the maximum response time.

The location allocation models used are the Multi Facility Location Allocation problem (MFLA), the p-center problem and the p-median problem. The method used for solving the MFLA is the Multi Restart Cooper. The p-center and p-median problems are solved using a metaheuristic called the Noising Method, very similar to Simulated Annealing.

The project is carried out under the assumption that the traditional stochastic models used for ambulance allocation do not match the way Falck operates very well. Hence the approach is to place the resources at hand as well as possible in a given situation, without consideration of what happens when the number of available ambulances changes.


Keywords: Location, Allocation, GIS, Ambulances

# Preface

This thesis has been established with the help from Informi GIS A/S and Falck A/S.

My background is a vide variety of engineering fields: Geographic Information Systems (GIS), Operations Research (OR), Building construction, Surveying, Traffic modeling and Basic engineering disciplines as physics, math and statistics.

The last two years of my education as a Civilingeniør ("Master of Engineering") at the Technical University of Denmark (DTU) has primarily dealt with GIS and OR, hence it seemed logical to combine the two fields in my thesis.

Through a trainee project at Informi GIS I got acquainted with Heino Sørensen who inspired me to do this project. He also did his thesis on ambulance allocation using GIS and also in cooperation with Informi GIS and Falck. I found the project inspiring, not only because it would let me combine GIS and OR, but also because it would involve a practical problem with data from "real life".

The project has been carried out under the supervision of Henrik Juel, Informatics and Mathematical Modeling (IMM) at the Technical University of Denmark (DTU).

When writing this report I saw two options, either do a more theoretical study with theory and formulas or do a more easy to read report with a much broader audience. I chose the last since it could help more people benefit from the project, especially Informi GIS and Falck.

There has been done a large amount of programming in this project, traditionally source code and often also data is printed out and handed in with the project. However the amount of both source code and data is so large that it is without justification to print it out, especially considering that it is seldom read. A good guess is that the source code is about 5000 to 7000 lines, and the data both available to this project and produced by this project would fill more than one CD (680 MB). A CD with the source code and the most important data is handed in for examination purposes only (data may not be used without the permission of Falck A/S and Kraks Forlag A/S).

References are presented as [AA p. 1]. "AA" refers to the full reference at the end of the report, "p." to the page number of the reference, in this case page one. Definitions of a few terms, which might trouble some readers is included in the Definition at the end of the report.

The computing power used (when real computing time is mentioned) is:

Amd Athlon Palomino 1800+ (1.53GHZ) Processor
256 MB DDR PC2100 RAM
Running Microsoft Windows XP Professional

In some of the figures, in this report, continuous curves are used to represent discrete data, some might argue that it in principle is wrong to do so, though it has been done to make the figures easier to understand, since using bars would not have been a good alternative. In some cases bars are used, especially when the differences between the different “series” presented on the figure are so small that they could not have been revealed by continuous curves.

I apologize for any figures which may look a little odd, MS Excel and MS Word, does not cooperate all that well. What I see on the computer screen is not always what comes out of the printer, sorry. Though there should be only cosmetic errors left.

The thesis is writing in English, not to annoy any Danes, but help me get a job outside Denmark if things get too dull around here.

Christian Krog Lindeskov, c971771. 30th of August 2002

## Thanks to

Heino Sørensen at Informi GIS for doing his thesis and for help along the way.

Jepser K. Petersen and Søren Andersen at Informi GIS for allowing this project and Informi GIS as well for letting me use their software.

Torben Ruber, IT-manager at Falck, for having so many great ideas and allowing this project.

Søren Dejgaard Thomsen, for preparing data, and Steen Gylling Nielsen for helping me understand how Falck operate, both at Falck.

Kraks Forlag A/S for letting me use their data for free.

Henrik Juel  at IMM, DTU, for being a great supervisor.

Jens Clausen at IMM, DTU, and Peter Falster at ELTEK, DTU, for starting this project and for helping along the way.

Nina Lindeskov, my aunt, who has assisted me with the English.

# 1 Introduction

*This chapter gives a short introduction to the topics dealt with in this report.*

Solving location-allocation problems is not a new discipline. Since the beginning of mankind it has been a high priority to find a good solution to location-allocation problems, from the early years of trying to find a good place to settle near resources such as wood, food and water till today's problem of placing transistors in CPU chips while maximizing speed and minimizing heat. Despite the long history and rewarding benefits of solving these problems to optimality, many are still difficult to solve to optimality if not impossible.

This project deals with the location allocation of mobile units. The focus of this project will be on placing ambulances on the Danish islands of Funen, Tåsinge, Thurø, Siø and Langeland, which can all be seen on Figure 1.

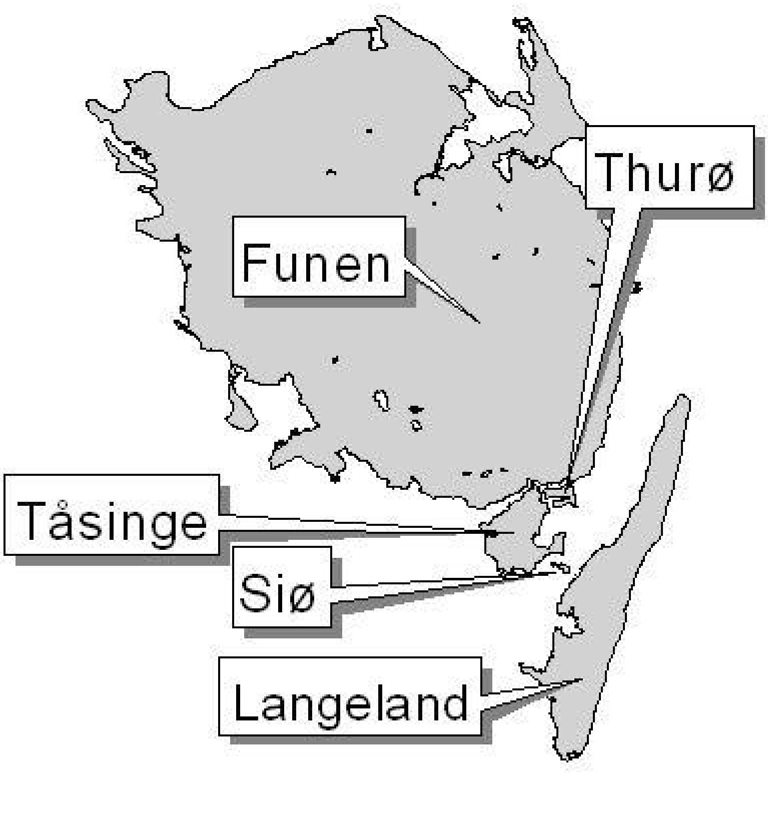



**Figure 1. The island investigated in this project.**

The islands are connected by bridges, making access by ambulance easy. Together these islands will be referred to as Funen. The reason why the rest of Denmark does not need to be taken into consideration is that ambulances seldom move off Funen to assist in other regions.

Data has been supplied by the leading provider of ambulance services in Denmark, Falck Danmark A/S (from now on Falck), and by one of the major distributors of digitized maps in Denmark: Kraks Forlag A/S. Often when ambulances respond to an emergency it is a matter of life and death with time being the crucial factor. Poor performance by Falck or other providers is very likely to reach the front page of leading national newspapers. In order both to increase performance and profit, Falck is always seeking technology and knowledge that can help them improve their routines. The latest major technology step is using Geographic Information System (GIS) and the Global Positioning System (GPS) to identify the positions of current available ambulances and to dispatch the ambulances so that they will reach the locations of accidents as quickly as possible.

The aim of this project is to investigate the possibility of introducing methods from Operations Research (OR) to station ambulances as optimal as possible prior to the received emergency calls. However, there are many problems involved in this process, such as:

How is optimality defined?
How are future accidents predicted?

Before attempting to answer these questions, a short introduction of Falck ambulance service will follow. GIS and OR will be explained later in its own context.

# 2 Project Formulation

*This chapter is a precise definition of what is being investigated in this project.*

The aim of this project is to investigate the possibilities of using Geographic Information Systems (GIS) and Operations Research (OR) to allocate ambulances. The data consists of elementary GIS data such as a road network, demographic data, area use etc. and data from the Danish company Falck, such as ambulance duties/accidents and Falck-resources (number of vehicles, garages etc.).

The investigation will be carried out in three steps.

1) Analysis and preparation of data for optimization algorithms

   The main purpose of this step is to understand the data, using it to create data structures/models that can be used later in the project. This will be done using relatively simple statistics and the features in the GIS software ArcView 3.x and its extensions Spatial Analyst and Network Analyst. The aim is to process the data in such a way that it can be used in the optimization methods.

2) Static optimization

   This step will present and evaluate different models for allocating ambulances. Using the data from step 1 with the Multi Facility Location Allocation model with Euclidean distance (MFLA), the p-median problem and p-center problem. This is done considering the problem as a static problem with a fixed number of accidents and a fixed number of ambulances to be placed. The possibility of expanding the models with constraints, such as time and capacity requirements, and multi-criteria objectives to satisfy the more complex aspects of the real life problem will also be discussed.

3) Future possibilities

   A discussion of how the methods used in the project can be improved to handle more complex aspects, such as dynamic allocation of ambulances.

This project will hopefully help Falck and their partners to develop systems that can improve the allocation of ambulances, hence provide a better service to their customers and possibly even save lives

# 3 Falck Danmark A/S

*This chapter explains who Falck is and what it is that they do.*

Falck is part of the worldwide company Group 4 Falck A/S. Falck is active in many different areas most of which deal with helping people in some sort of an emergency, some of these being:

- Fire fighting
- Auto assistance
- Ambulance service for emergency
- Ambulance service for transportation of patients to and from treatment
- Security facilities for private and businesses

Common for all of these is that time is a critical factor in one way or the other. Fire fighting is about material damage and often life and death, ambulance is also often about life and death, whereas the auto assistance is helping people out of an inconvenient situation as quickly as possible.

Falck is a private company and unlike many other countries Denmark allows commercial interests in such vital areas as fire fighting and ambulance services. This is due more to a historical development than politically based, but Falck has proved to be quite competitive and is expanding to other countries. Falck is by far the largest provider on the Danish market in both areas, but they are not the only one. The largest competitors are "public companies" owned by municipalities or counties and organizations based on volunteers (fire fighting only).

Though this project is about ambulances, many parallels can be drawn to the other areas of Falck's business, primarily fire fighting, auto assistance and the security section, which may also be able to benefit from the investigations of this project.

## *3.1 The Ambulance Service In Depth*

Falck is hired by municipalities to operate in a specific area according to some requirements agreed upon, such as not more than 10% should wait for more than 20 minutes for an ambulance and the average ***response time*** should not be more than 10 minutes.

Response time is the time between an emergency call is received and the first unit arrives, this also applies if several ambulances are required, i.e. major traffic accidents, then it is still only the first unit that counts. Not all units are ambulances, Falck also have agreements with doctors and nurses in remote areas, such as small islands. Motorcycles with medical personal being more mobile than ambulances, and even helicopters are used once in a while, however ambulances are by far the most common response unit.

No legal limits have been stipulated for the response time, other than those in the individual contracts. However, with regard to fire fighting there is legislation stipulating a maximum limit for how long it must take before the fire fighting crew arrives, i.e.10 minutes in cities and 15 minutes for other areas. Like other companies Falck will probably raise their prices if they are to give such guarantees. Hence contracts are often focused on the average time rather than worst case. National politicians have been discussing introducing maximum requirements on the ambulance services. However, as I see it there is one important aspect to be considered, there are less than 14.000 fires a year and around 300.000 accidents (in all of Denmark), therefore covering extreme situations with many fires/accidents at the time are probably a lot simpler to handle for fire fighting than for the ambulance service. Since extreme situations are a lot more extreme when it comes to ambulance service compared to fire fighting.

It would be an easy task for Falck if they knew in advance where to pick up the next patient, however if accidents could be foreseen they probably never would happen. Hence Falck has to cover the entire area, but still take into account that the risk of accidents is greater in some parts of the area than others. The tradeoff between low average response times and few high-end response times is a political matter rather than a commercial matter.

There are other complicated matters than predicting accidents:

In their ambulance service Falck is limited by:

- The number and sort of resources that they can afford, and still make a profit.
- The technical limits of the equipment. Ambulances do have both physical and practical speed limits.
- The requirements of the ambulance crew’s union, such as the number of breaks during the day and the salaries.

Still the Falck situation is even more complicated than this. Ambulances are also used for other assignments that emergency. As mentioned earlier, some people have to be transported using ambulances, a job that Falck also takes care off. In some cases, when e.g. an ambulance is on its way to pick up a patient from a hospital and move him to another hospital, the ambulance may be redirected to an emergency. The reason being that a man with e.g. a complicated leg fracture will survive waiting 15 minutes more whereas the man with a possible heart attack might not. Besides this Falck often chooses to place ambulances out in the field rather than in their garages, both to meet the requirements of the municipality and to cover a temporary increased risk of accidents, such as large sports events, concerts and highway rush hour traffic. Furthermore the ambulance crew is a not a crew of specialists, they are trained to handle many types of assignments, ambulance, auto assistance etc. Falck’s flexibility within its organization might be a commercial advantage, but it also tends to complicate matters.

It is clear that the problem as a whole is rather difficult to handle, and cannot be dealt with in its entirety by one person within six months. The focus of this project will be to “optimize” the response time and then only for the emergency part: ambulance service. Allocation of resources to or form other areas will not be considered, nor which type of unit is used, all unites are considered to be ambulances.

# 4 What is GIS?

*This chapter is a very short and not very detailed introduction to GIS, it is meant to help readers with no knowledge of GIS whatsoever to understand what GIS is and how it is used in this thesis. For more precise information about GIS, www.gis.com would be a good place to start.*

GIS is an abbreviation of Geographic Information System, and is normally used about systems that handle geographically related data in one way or another. The shortest and most precise definition I have found is:

*"...GIS is a computer system capable of assembling, storing, manipulating, and displaying geographically referenced information, i.e. data identified according to their locations. "* (http://www.usgs.gov/research/gis/title.html, 27 March 2002).

The type of GIS explained in this project is the "ArcView 3.x way", of which only the most important features have been used. ArcView is the GIS software used in this project. In Appendix III: How it works – Software Engineering"

## *4.1 Representing Data*

Data is stored as either vector data (points, lines, polygons) or as raster data (grids, images etc.).

### 4.1.1 Vector Data

Vector data is basically represented using points (vertices), i.e. lines are composed of two or more points, and polygons consist of three or more points. Both lines and polygons have a direction, i.e. the order of their points, hence the left and right side of a line can be defined.

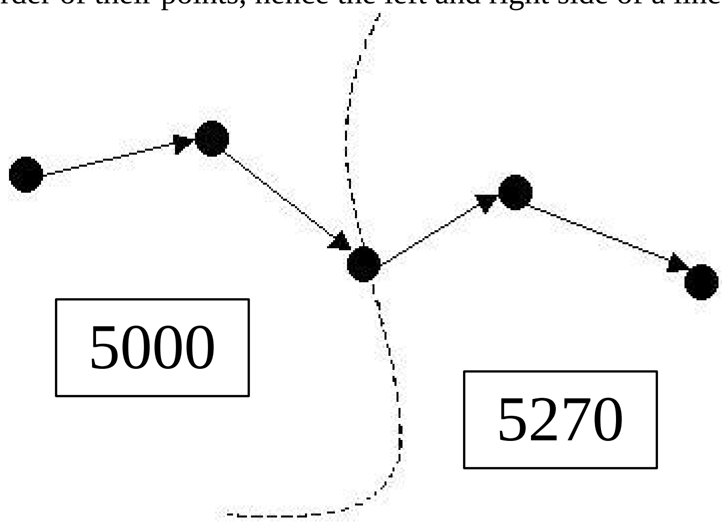



**Figure 2. Example of vector data: Polyline**

The vector data is referred to as either shapes or features. Features have attributes like records in a database. A feature can only have one set of attributes. As a result of this a feature, e.g. a road with a postal code attributed must be spilt into two features if it lies within two postal code areas. An example of this is shown in Table 1.

| Shape information | ID | Length | Postal code |
|---|---|---|---|
| Shape information | 1 | 4,2 | 5000 |
| Shape information | 2 | 4,4 | 5000 |
| Shape information | 3 | 3,9 | 5270 |
| Shape information | 4 | 4,1 | 5270 |

**Table 1. Attribute data.**

Topology is an important matter when working with GIS. Topology is the description of how features relate to each other, or you could say what they know about each other. Topology may be explained as two lines that are connected or two polygons that are adjacent.

ArcView 3.x does provide some topology facilities that allow features to relate to each other. It is for example possible to define two roads crossing each other as either connected or unconnected. For a road network the connected case could be an intersection and the unconnected case could be a bridge crossing another road.

The topology allows the network representation to be converted into a graph (directed graph/digraph), which can be used for shortest path algorithms such as Dijkstra.

### 4.1.2 Raster Data

Grids and images are characterized by presenting some areas as number of squares, with a value assigned to each square. Each value or interval of value then has a color assigned to it.

## 4.2 Working with Data

Features are collected in themes (layers), such as “road network”, ”accidents”, “outline of Funen” etc. Themes are gathered in a stack so that the features in the top theme will cover the themes beneath. An example of Falck garages on top, then the road network and at the bottom the theme representing Funen, can be seen in Figure 3.

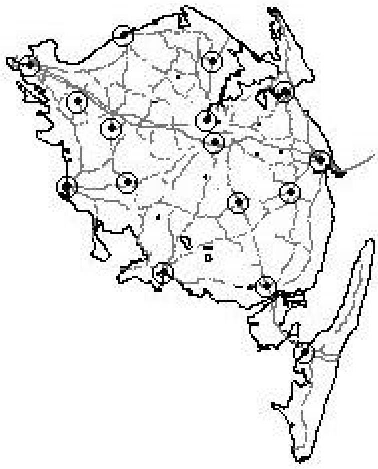


**Figure 3. Geograhic data from Funen. On top is Flack Stations (there are less today), then roads with speed limits above 70 km/h and beneath that the outline of Funen.**

The themes can be switched on and off. Spatial queries such as determining all accidents within ten kilometers of a Falck garage or locating all the highway approaches close-by can be performed. A feature that has been applied much in this project is the ability to snap one feature to another feature. In order to calculate the distance between two points on a road, the points actually have to be on the road. An example of snapping is shown in Figure 4.

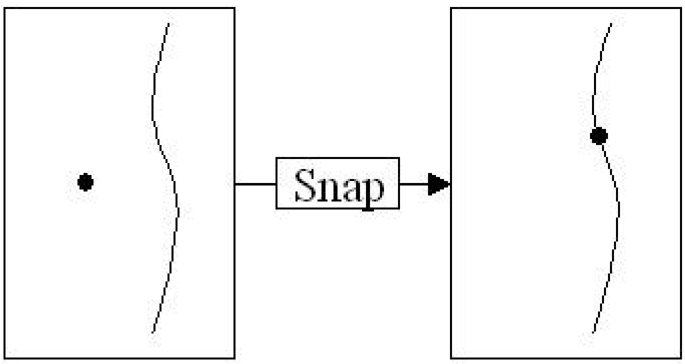



**Figure 4. Snapping a point to a line using "snap to boundary"-rule.**

Snapping can be used to move a point onto a line using a rule such as snapping to the nearest end of the line or closest “point” on the line (“snapping to boundary”), as shown in Figure 4.

### 4.3 Using GIS

Basically GIS is an interactive atlas on a computer, with all the possibilities that it entails. Moreover GIS can communicate with other programs such as web-browsers, spreadsheet database systems etc. making it a very powerful tool in any environment that deals with data that has a geographic dimension.

# 5 What is Operations Research?

*This chapter introduces the concepts of Operations Research (OR) used in this project. There is much more to OR than presented in this chapter.*

> *Operations research can be defined as "a scientific approach to decision making, which seeks to determine how best to design and operate a system, usually under conditions requiring the allocation of scarce resources".*
>
> *Professor Wayne L. Winston*
> *Indiana University, USA*
>
> *The short version is "quantitative planning".*
>
> *Professor Jens Clausen,*
> *Informatics and Mathematical Modeling,*
> *Technical University of Denmark*

Operations research (OR) is used in many areas, crew and fleet scheduling, location and distribution, production planning etc.

A typical way of applying OR to a real life problem could be:

1. Create a model of the problem.
2. Solve the model.
3. Validate the solution.

## 5.1 The Model

A model in OR is a tool used to define what a feasible solution to a problem is and what a good solution is. A model often has an objective such as minimizing the cost, maximizing profit, minimizing the maximum travel time etc. however sometimes the objective is nothing more than finding a feasible solution.

There are different ways of defining models. The most common one is probably the mathematical formulation, a method that uses equations to define the model. Another common approach is a graph-definition often used for problems such as finding shortest distance in a network and other network problems.

One of the most obvious reasons for using a mathematical model is that it often can be solved as is, especially if the equations are linear. Another good reason is that a mathematical model is a very clear statement of what is actually being solved. The drawback is that people who are not familiar with this type of language, may find it difficult in the beginning to understand the concept. Therefore it is often a good idea to combine the mathematical description with a more common explanation.

Models also serve the purpose of simplifying the problem. Real life often has so many stochastic elements and other complex elements such as physical laws and labor union rules etc. that it is impossible to model them all. By simplifying the problem it becomes easier to solve it, the drawback however is that there is no guarantee that the optimal solution to the simplified problem will also be the optimal solution to the real problem. Still, a well-defined simplification will nearly always prove also to be a good solution to the real life problem.

## *5.2 Solving the Model*

Once the model is clearly defined, it is time to solve it. The first reaction may be, : "How difficult can it be?"..."Computers today are very fast, they must be able to solve anything within a few seconds." However, this is far from so, in order to explain this a new term needs to be introduced: Computational complexity.

### 5.2.1 Computational Complexity

There are basically two things to be understood about computational complexity. There is the number of possible solutions to a problem and the amount of time it takes for a solution procedure to solve the problem. These two factors are of course often correlated, i.e. it takes more time to solve complicated problems, since these very same problems often have more possible solutions. The most important of the two is of course how much time it takes to solve a problem to optimality, rather than how many "non-optimal" solutions can be found to a problem.

The reason for introducing a concept like computational complexity is that when a problem doubles in size, more roads, customers, facilities etc. it does not necessarily double in running time / number of possible solutions. Actually it might not even be polynomial; many problems are exponential if not worse. Figure 5 illustrates five functions: three polynomial (linear, quadratic and cubic) and two non-polynomial (exponential and faculty) on a logarithmic scale.

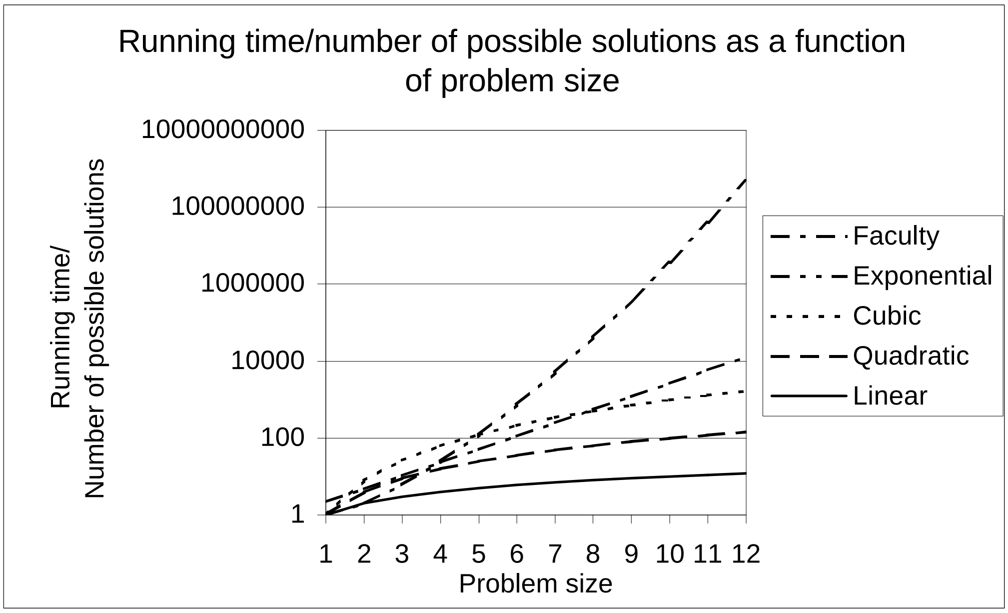



**Figure 5 shows the relation between solution time (logarithmic scale) and problem size.**

There is a special category of problems which no one has found a polynomial time solution methods for: NP-hard (NP stands for nondeterministic polynomial) [DSA p. 446].

It is clear that if the problem can be reduced in size, a lot is to be gained. Regarding the problems in this project there might be something to gain by reducing the number of accidents from 60,000 to perhaps 5,000 or even 1,000, how this can be done will be discussed in Chapter 9: Representing Demand.

Figure 5 also illustrates why trying all possible solutions might not be a good idea. The single facility location allocation problem is a classical example of this. The objective is to place a facility so that the average weighted distance between facility and the customers is minimized. Since the facility can be placed anywhere in space (often only two dimensional), the number of possible solutions is infinite. Another problem is the *p*-meidan, which may have the same objective, but has a finite number of solutions, since the facility can only be placed at a customer. If there are *n* customers and only one facility to place, there are only *n* possible solutions however if there are 10 facilities and 1000 customers the number of possible solutions are the number of different ways 10 facilities can be placed out of a 1000 possible locations which is:

$$\binom{1000}{10} = \frac{1000!}{10! \cdot (1000-10)!} \approx 2.6341 \cdot 10^{23}$$

Even if it would be possible to test one billion solutions every second, it would still take more than 8 million years to have tried all possible solutions. And it must be remembered there are a finite number of solutions. More on that in section 11.2.3: “Complexity of the p-median and p-center problems”.

#### 5.2.1.1 Big O notation

A special notation for computational complexity is the Big O notation. The Big O notation is represented with the symbol O(·), where the · is the input size of the problem being solved, hence O($n$) is linear complexity, O($2^n$) is exponential and so forth. O(·) is an upper bound on the running time. There also exists other types of notation, a recommended reading is “Data Structures, Algorithms and Applications in C++” by S. Sahni [DSA p. 83]. The O(·) might be far form a practical running time, it should be considered a “worst case scenario”. However the running time also depends on how an algorithm is implement, hence the skill of the programmer and the programming language used can have an influence on the O(·).
O(·) will be used in this project with the intention that those who have an interest in O(·). can use it, nevertheless it not necessary to understand O(·) to get a thorough understanding of the solution methods used.

### 5.2.2 Solution Methods

There are many different ways to solve optimization problems, the most obvious one is to try all solutions, however if it were that easy, why would there be a science called OR? Many problems, as mentioned above, have so many feasible solutions that it is impossible to try them all.

In order to solve such problems an intelligent search for solutions is needed. Many problems may be solved by exact solution methods, which find the best, hence optimal solution. The single facility location problem e.g. is a good example of this.

Problems that can be formulated relatively simple i.e. using linear equations; can be solved by using a method called simplex. Software such as CPLEX and GAMS can solve problems that are a bit more difficult, but they still rely on the mathematical formulation, and generally do it quite well. Nevertheless, as the problems increase in size so does the number of equations in the formulation and for quite big problems even CPLEX and GAMS will continue calculating “forever”.

It is a classical problem in OR that when problems grow in size, i.e. more customers, more facilities etc., the exact solution procedures often run into the same problem as with trying all solutions, it simply takes too long time. In other cases there exist no exact solution procedure for the problem at hand. For those purposes approximation algorithms, also called heuristics, have been developed. Some of them have been designed for solving a specific problem, such as the solution procedure for the single facility location problem. Other heuristics called metaheuristics are designed to solve optimization problems in general, they are frameworks needing a little customization to solve the problem at hand. The most famous of these are Simulated Annealing (SA), Tabu Search and Genetic Algorithms.

They all try to achieve the same thing, to find local minima in the solution space (set of all feasible solutions), and then again escape these local minima to find other local minima, hoping that one of these minima will be the optimal solution, and if not that at least the best found solution is acceptable. The major drawback of these solution methods is that there will be no guarantee of the quality of the found solution. In theory there exists (at least for SA) an investigation that concludes that given the right parameters SA will find the optimal solution [HA], however it is seldom possible to carry out in practice. There also exist descent algorithms, which go straight for the local minima, but have no utilities for escaping these local minima, they always yield the same solution if given the same staring parameters.

## *5.3 Validating the Solution*

Once a solution has been found it is often necessary to test it in reality. The model solved is seldom a perfect copy of the real life problem that was actually to be solved. However the real world is very complex and often the only way is to start using the new results and see how they perform compared with perhaps older methods. Often a few simulations or tests can be run on historic data, but the underlying model is often just a more precise “simplification”, hence not the real thing.

## *5.4 OR is More than Finding Solutions*

Some solution methods, such as simplex, also provide other information than the optimal solution and its solution value. Marginal information (marginal cost) can be retrieved from the calculation to find out where it the pays-off to improve the production line etc. The advantage is that it is not necessary to calculate a wide range of solutions to find out where to improve, thereby saving much time and work.

# 6 Previous Work

*This chapter will provide a very short overview of how others have dealt with ambulance allocation.*

The most common approach to stochastic demand, like accidents, is to define a set of server locations, where *p* ambulances can be placed and then minimize the probability that some “accident ” cannot be serviced within a given service time *S* (response time). This is typically done using extensions of the Maximum Covering Location Problem (MCPL) a model, which maximizes the number of accidents covered with in a given *S*. The most common extensions are the Maximum Expected Covering Location Problem (MEXCLP) by Daskin [SAP p. 2] and the Maximum Availability Location Problem (MALP) by Revelle and Hogan [SAP p. 2]. The main downside for these models is the use of coverage within the service time, *S*. If an accident cannot be covered, or if an ambulance is not available, they must be serviced from “elsewhere”. The models accept that some accidents may not get attended to within the service time. Furthermore the models do not consider the average response time as a part of their objective. Finally the models make a lot of assumptions on the “stochastic behavior” which is not necessarily satisfied.

There are many other ways to approach the problem of ambulance allocation using a stochastic demand. S. H. Owen and M. S. Daskin made a detailed overview in: “Strategic Facility Location: A Review” (1998), European Journal of Operations Research [SAP p. 19].

In 2000 H. Sørensen did a master thesis at Department of Electric Power Engineering (ELTEK) at the Technical University of Denmark (DTU), studying the use of the MALP on Falck’s data from the northern part of Jutland, Denmark [HS]. Sørensen showed how ambulances should be placed to ensure the most reliable coverage. He also found several problems in using the MALP model. First of all the assumptions for the model were not fulfilled entirely, secondly the lack of consideration to the average response time is a problem and third: only 50% of the accidents were attended to from the Falck garages, which H. Sørensen used as server points [HS].

# 7 Reducing the Problem

The model investigated by H. Sørensen (chapter 6: “Previous Work”) relies on two things, which to the findings of this project do not comply with the real life problem at hand. First of all the assumption that there is always at least one ambulance in each of the garages, and that the ambulances are expected to respond to emergency calls from the garages do not correspond with the real life situation very well. Secondly, the stochastic assumptions made about distributions are not likely to be fulfilled by real life data, hence there is no “guarantee” that the model provides a very accurate picture of the real life situation, despite its very fine mathematical properties.

The approach in this project will be different from most other work done on ambulance allocation (according to the knowledge of this author). The main idea in previous projects has been to assume a kind of static placement of ambulances at Falck stations, compensating for the static placement by using allocation ambulances so that a certain response time can be met for 95% of the accidents.

In this project the ambulances will be placed according to the following philosophy: Given a certain demand (expected accidents) and ten ambulances, what is the most optimal placement of the ambulances?

The placement of the ambulances will be carried out without a consideration of the coverage when an ambulance is called to an accident. The reason for allowing such a model is the dynamic use of Falck’s resources, especially the use of ambulances which are assigned to other services, such as transportation assignments, which may be canceled, to ensure coverage for the emergency service.

In order to build and use such a model the following four elements must be investigated and decided upon:

- The demand that the ambulances have to meet.
- A measure of response time such as travel time / distance, i.e. what is the price of moving one ambulance from A to B.
- An objective for what is optimality.
- Possible requirements of various sorts, such as capacity, limits on maximum response time, etc.

## *7.1 Demand*

What is to be the determining factor with regard to demand: population density, type of area i.e. residential or industrial, the geography (cover the entire Funen), and accidents in the past or something else?

GIS provides a lot of options, any information about people and their whereabouts could be useful. The most obvious thing is to make use of the accidents in the past. It might be conservative to think that this year will probably be like last year, however the overall geographic pattern of accidents as a whole is not likely to have changed all that much from one year to the next year. It could be interesting to compare the data on accidents with information about people's whereabouts, land use etc. However, this is not within the scope of this project.

The demand used in this project will be a composition of geography and accidents in the past, more on that in the section 12.3.4.1: "The Cost dij".

## 7.2 A Measure of Response Time

The best measure would be the real travel time from one place to another, however such a measure is not a fixed value and hence difficult if not impossible to work with. The most obvious alternative is one of the following:

Network travel time:
: The digitized road network can be transformed into a graph, with the cost of each edge being an estimated/expected travel time based on the speed limit on the road, which the edge represents. This is the traditional way to handle road networks in GIS. This model, however, lacks both the dynamic and stochastic elements of a real road network. Dynamic aspects such as rush hour and the general variation of traveling from A to B are not considered.

Euclidean distance:
: An even simpler model is to eliminate the roads and use a distance measure such as Euclidian or Rectangular distance, or another p-norm distance. However, this model lacks a time reference, and since time is a crucial parameter, this might be a fatal flaw.

For both types of travel cost it is possible to establish models, which can be solved, however for large instances there is no guarantee that an optimal solution can be found. Though a good solution should be possible to find.

The time it takes for the ambulance crew to receive an assignment from the emergency operator, or to get into the ambulance is not considered at all.

## 7.3 Objective

What characterizes an optimal solution?

1. The "best" possible service to the people whom Falck serve?
2. The maximum time it takes to reach an accident should be as short as possible?
3. The average time it takes to reach an accident should be as short as possible?
4. It is done as cheaply as possible?
5. It is done with as little risk as possible?

There are many choices, the optimal solution would be to include them all, however when it comes to comparing them there is no clear answer to what is best: One accident less with ambulances involved or an average response time 20 seconds faster? The good thing is that all of the subjects can be treated individually. It is not that they do not affect each other, they do, but they are not prerequisites for each other.

The objectives dealt with in this report will be the average response time and the maximum response time, since these are expected to be the most important to Falck.

## 7.4 Requirements

Requirements can be many things, labor union rules, a limit on the number of expected accidents an ambulance is allowed to cover (capacity constraints), a limit on the maximum response time etc.

The most obvious to introduce are the capacity constraints and the limit on the maximum response time. Both requirements will be dealt with several times in this report.

## 7.5 Conclusion on Reducing the Problem

It should be quite clear that it is both necessary and possible to reduce the problem to something that can be modeled and solved, while providing Falck with valuable information.

The next step is to obtain an understanding of which data is available to this project and what it consists of. Secondly, it will be necessary to create a model of what is being solved and find/develop methods that can solve the model given the available data.

# 8 Data

*This chapter is a discussion and a presentation of the data available to this project and how it used.*

The main elements of the data are the accidents and the road network. Data or rather the positions of the Falck garages are also available, as well as cosmetic data like the extent of Funen. A more technical description can be found in Appendix I. The data was partly delivered by Falck (the accidents and Falck garages) and by Kraks Forlag A/S (road network, cosmetic data etc.).

### 8.1.1 Accidents

Accidents only include responses to emergency calls, however data for transportation and other jobs carried out by ambulances are available, though not used in this project..

The data about accidents consists mainly of addresses, time etc. Data is from the some time early in 1999 till 18$^{th}$ of March 2002 22:43 (10:43 pm) (more on this in section "8.1.3: Geocoding."). There are no coordinates to the accidents. The only geographic reference is the address (or assumed address). That an accident has happened in "Jernbanegade 4, 5000 Odense C " is not a very precise position; it could still be several meters, even hundreds, from the position of the address to the correct location of the accident. It is not even certain that the accident actually happened there it might as well be in number "Jernbanegade 6".

Accidents are placed on the network using a process called geocoding, more about that in section 8.1.3: Geocoding.

The time information available is the time of the call, the time that the ambulance reaches the accident and the time when the ambulance is available again.

There are many other types of information, such as which ambulance serviced the call, to which station does the ambulance belong, and much more. Some of this is described in Appendix I, which also includes some comments on possible errors in the data.

### 8.1.2 Road Network

Besides the geography, the road network contains information about one-way streets (the network can be considered to be a digraph), speed limits, travel time, and information about which addresses belong to which road. The travel times are based on the length of the road and its speed limit; hence it does not take traffic congestion, traffic lights and other variation in the traffic pattern into account. The travel time is a rough estimate of how much time it takes to travel a certain path, but is much more precise than distance models, such as rectangular and Euclidean distance. The travel times used in this project are not the same as those of Falck, nor can the travel times be used to evaluate the current performance of Falck. The travel times in this project should only be used to compare solutions and methods used in this project.

The road network has been altered a bit. The geographic data remains the same, however the one-way information and the travel time / speed limits has been altered.

#### 8.1.2.1 One-way

All roads with the “one-way” attribute set to “n”, the symbol for no car traffic, have been altered to “” (null string), the symbol for free traffic. Hence ambulances are free to travel on the entire network, this however is not always true. Sometimes the road is physically blocked, and cannot be passed even by an ambulance, however many of the “n”-roads are pedestrian streets”. The main reason for altering this was that many accidents occur on addresses on “n”-roads, and hence would have required some special placement rule, if they were to be accessed by an ambulance. Since it could give wrong results both to assume that ambulances could and could not use an “n”-road, the simple solution was to choose: no “n”-roads. True one-way streets however have not been changed. Ambulances are generally not allowed to drive the wrong-way, one reason could be that having ambulances driving the wrong way on e.g. highways is not a good idea, however it does happen once in a while.

The speed limits have also been altered. All speed limits lower that 40 km/h have been raised to 40 km/h. This has been done to avoid the dramatic impact that a low speed limit has on the travel time. Some streets (many of the “n”-roads) had a speed limit of 2 km/h which is slower than walking. Why an “n”-road has a speed limit is not quite clear, however not all transportation has to be by car.

#### 8.1.2.2 Travel Time

The travel time was recalculated to:

$$\frac{\textit{length of road}}{\textit{speed limit} * 0.85} = \textit{travel time}$$

The parameter 0.85 was primarily based on the fact that it was a very common factor for the travel time data supplied with the road network data. It is important to keep in mind that the model is still a very primitive model, since this model does not take traffic loads and turn cost into consideration (turning in traffic lights actually takes much time when traveling in town areas).

The road network also has errors, i.e. parts of data do not reflect real life the way they should. However the number of errors is so low that it is expected to have a minimal impact on the solutions.

### 8.1.3 Other

Other data used in this project is the positions of the Falck garages. Since most ambulances respond to emergency calls from Falck garages this can be used as a reference “comparing” how Falck is expected to perform in a given situation. Such a comparison is necessary since comparing with the actual response time of ambulances is a bad idea, as they use a “different” measure of travel time (the real one) from the one available to this project. Data such as the outline of Funen and other “cosmetic data” will also be used.

## 8.2 Geocoding

The traditional way of handling address-based data in GIS is to geocode. The idea is quite simple. Most road segments have a street name, and information about zone/postal code (zip code, postnr. etc.), and the house numbers on the road (including literary numbers such as 4c) and the numbers on each side of the road.

As an example the following two addresses could be geocoded:

Højland 8, 5000
Lavland 22, 5000

The process is illustrated in Figure 6 and Figure 7:

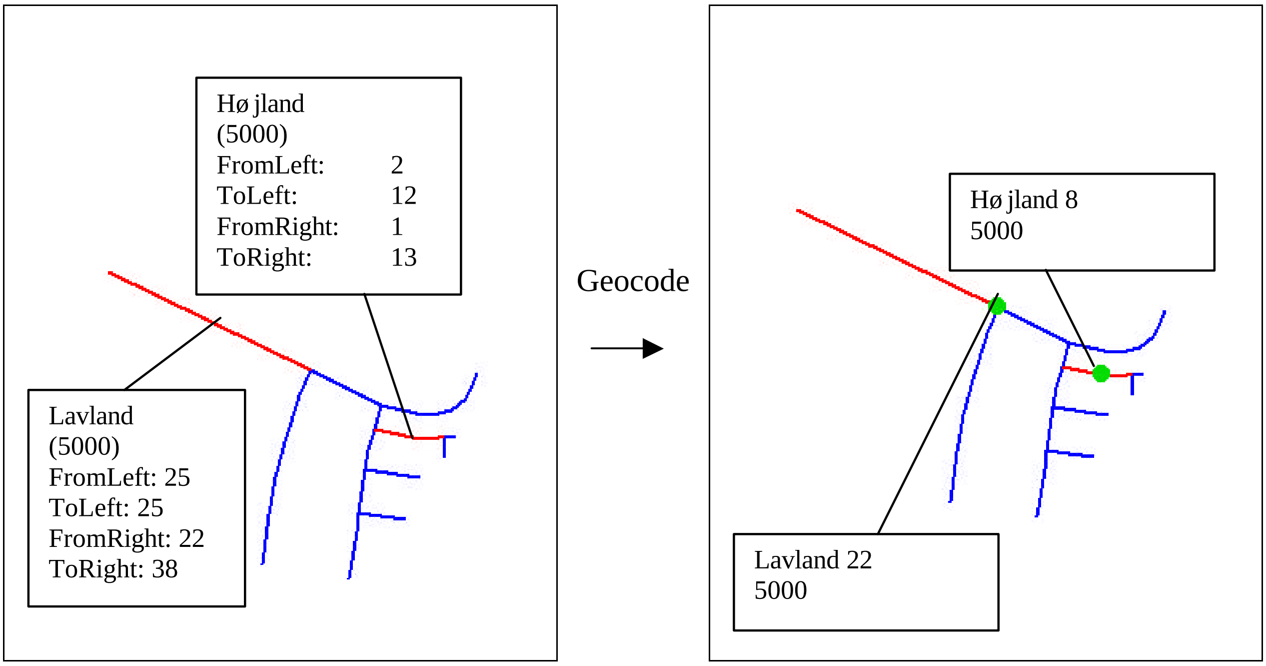



**Figure 6. Road network with address information.** **Figure 7. Addresses geocoded.**

Lavland 22 is placed at the beginning of its arc and Højland 8 is placed 60% along its arc. Geocoding is a bit more complex than shown here, among the more advanced features is the scoring system used in ArcView 3.x that takes spelling errors and the like into account.

### 8.2.1 Geocoding the Accidents

Prior to geocoding all data originating from before 11th of January 2000 22:43 (10:43 pm) all data with obvious errors, such as missing data and invalid data, refer to Appendix I for a full explanation. Data was reduced from 76144 accidents to 65619 (due to errors in data). After working a little with the address information, see Appendix I for a full explanation, data was geocoded, and 7,5 % (4905) of the accidents could not be matched to a known address. 1804 (2,7 %) of these were accidents on the islands located close to Funen, but not included in this project because the only connection is by boat, and therefore not part of the road network used. All in all about 95% (60714) of the accidents could be geocoded. No further steps have been taken determine errors in data, or to geocode the unmatched accidents. The reliability of the data is expected to be relatively high (due to the high match percentage), but further investigations might be necessary before real life decisions are made using geocoded accidents. Errors are expected to exist [HS p. 65].

# 9 Representing Demand

*This chapter deals different ways of representing the demand (accidents).*

Accidents are represented as points on the road network, as shown in section: 8.2 Geocoding. But is this a good representation? It is the most precise one presently available, but it seems a bit difficult to handle as much as 60,000 points. As mentioned in chapter 5: “What is Operations Research?”. It will be much easier to solve a problem if the number of accidents is reduced. Two methods will be used to reduce the amount of data:

1. Separate data in time.
2. Aggregate data.

1). Data does not necessarily happened to follow the same pattern all year, week, day etc. it might be wise to separate data into different categories in which accidents happen individually, geographically speaking, more about that in chapter 10: “Accident Analysis”.

2). There is no need to have five points representing that five accidents happened on the same address or in the same neighborhood, instead one point, with the weight of five could be used. Perhaps something other than a discrete representation might be used. Often when dealing with data in GIS a more continuous data representation is preferable, like iso-curves (i.e. isobar, isotherm), an example is shown in Figure 8.

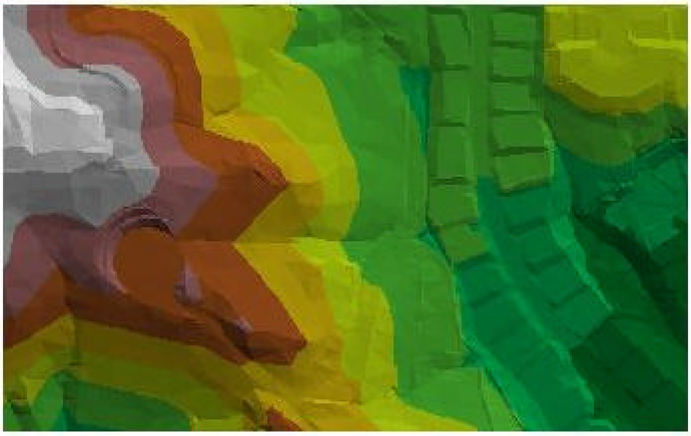


**Figure 8. Surface plot from the ArcView 3.x 3D Analyst tutorial, with 3D effects.**

Such a representation could also be relevant in this project. Accidents do not happen at the same address again and again, at least not in most cases. Representing data as areas with a probability of an accident occurring within a given amount of time would be natural. However, all the optimization models used in this project are based on discrete accidents data. The reason why the models only use discrete data is obvious; it is not possible to calculate the time it takes to drive from a given point to a certain “area”. The time depends on where in the area. Calculating the distance to the center of the area could of course solve the problem, but that would still be discrete. However, a visualization of the area-based idea will be useful, since it is difficult to look at 60,000 points on a 10 cm by 10 cm figure.

The rest of this chapter will deal with the various aspects of aggregation, in order to determine how to aggregate. The basis will be an aggregation from “points” to “ weighted points”. The weighed points will be referred to as ***demand points***.

## 9.1 Guidelines for Aggregation

The guidelines for aggregation have been adapted from Francis, Lowe and Tamir (Chapter 7 in “Facility Location: Applications and Theory” [FL]) and consist of the following six issues:

1) Aggregation error.
2) Computational cost to
    a. get demand point data.
    b. implement and run Aggregation Algorithm (AA).
    c. solve the approximating location model.
3) Ease of explanation.
4) Problem structure exploitation.
5) Robustness.
6) GIS implementable.

All issues and their relation to this project will be discussed in the following.

### 9.1.1 Aggregation Error

An aggregation error is the error introduced when an accident is assigned to a demand point. The error is the difference in response time/distance between the ambulance and the accident and between the ambulance and the demand point. Hence the error can be both positive and negative.

It seems reasonable to expect that the sum of all errors is zero, based on the assumption that the error is random [FL p. 211]. As such is should have no greater impact when minimizing the average response time. When minimizing the maximum response time on the other hand the error can have an effect in two different ways. The solution value, when minimizing the maximum response time, only consists of one response time.

The solution value, when minimizing the maximum response time, is the highest response time between any demand point and the ambulance to which it is allocated. Since most travel times between demand points and accidents are different from zero, there is no guarantee that the one response time which make up the solution value is actually the highest response time, given that most response times has an error. Hence the ideal thing to do with a model that minimizes the maximum response time / distance is to construct a set of demand points which can only have negative errors. Constructing such a demand point set might be difficult, since the position of the ambulances cannot be known in advance.

### 9.1.2 Computational Cost

The computational cost is only critical if the aggregation has to be done “on the fly”, which is not the case for this project. It does not really matter whether it takes one minute or one hour to calculate, as it will only have to be done “once”.

### 9.1.3 Ease of Explanation

Being able to explain how the aggregation algorithm (AA) works is very important to this project. Since the goal of this project is to develop methods, which can help Falck make decisions. It is vital to understand when the methods delivers unusable solutions and why, so that decisions are not compromised.

### 9.1.4 Problem Structure Exploration

This is a rather complicated matter, but the basic idea is to use the structure of the demand data, and the customers to create demand points that give a good representation. This of course is a bit contrary to 9.1.3 Ease of Explanation, since exploiting the problem structure often makes the aggregation algorithm more complicated. An important issue for this project is that even if an area does not contain any accidents it cannot be left out, since accidents may happen everywhere. Generation of demand points with virtual demand could be necessary. Aggregation methods that can generate a overview such as “all” of Funen could with advantage be taken into consideration.

### 9.1.5 Robustness

The AA must be trustworthy. It should not deliver completely different results if:

1) a few customers are removed or added.
2) problem independent parameters are changed.

and it should of course always

3) be a reliable representation of the original data,

such that a good placement of ambulances among the demand points would also be a good placement among the accidents.

### 9.1.6 GIS Implementable

This is very important since the project is based on GIS data and is focused to use GIS to solve ambulance-allocation problem. That GIS should have limitations as to how aggregation can be done compared with programming in general is not likely, since the GIS used in this project can interface with other software through Dynamic Link Libraries, which again can be created by programming languages such as Visual Basic and C++.

## *9.2 Possible Aggregation Algorithms*

The position of the demand points can be created in several ways. One very simple way would be to use the center or corner point of the polygon of a tessellation. H. Sørensen (the project prior to this) [HS p. 24] used a quadratic tessellation, but suggested that others such as triangular or hexagonal tessellation could prove useful as well. It also might be worth trying simple location allocation algorithms.

### 9.2.1 Within Polygon Algorithm (WPA)

The idea is to assign all the accidents within the polygon to the demand point (center of the polygon) belonging to the polygon (allocation to nearest demand points in terms of Euclidean distance). The demand point would then be the center point of the polygon, snapped to nearest road. Snapping is necessary to calculate the travel time on the network. This is the algorithm (quadratic) that Sørensen used.

The advantage is that it is very simple and fast, however there is no guarantee that accidents are assigned to the closest demand point in terms of travel time. An accident could be assigned to demand points on the other side of a barrier such as a lake, river or a highway, making the travel time very long. However, only under very special geographic circumstances is this likely to play an important role. According to Sørensen the position of the first polygon was critical for the value assigned to the demand point. The reason seemed to be that the accidents are congested around the cities. If a polygon covered an entire city it would get a value, which was roughly four times higher than if the city was split by four polygons, which might be a problem for some solution methods.

### 9.2.2 Closest Center Polygon Algorithm (CCPA)

Like the WPA polygons are made and their centers are snapped to the road network. The difference is that only the center point is used. All accidents are assigned to the nearest demand point in terms of travel time, though the position of the first polygon will probably still be critical. The computational cost will be much heavier than for the WPA, since a “closest facility” algorithm has to be run for every accident. The advantage is that the maximum error is easier to control, and is most likely to be very small.

### 9.2.3 Location-allocation Algorithms (LAA)

An idea could be to solve the aggregation as a location-allocation problem either as an average travel time minimization or as a minimization of the maximum travel time. However it may be impossible to find the optimal or near optimal solution, since the number of accidents and the number of demand points necessary to give a reliable representation is quite large.

### 9.2.4 Other Algorithms

There also exist special algorithms for specific problems, usually with the goal to reduce the amount of errors introduced with the aggregation [FL p. 215]. However I have not found any that seem to fit into this project.

### 9.2.5 Conclusion on aggregation algorithms

It seems most beneficial to try the polygon algorithms WPA and CCPA. These are very easy to implement, easy to understand and it should be possible to control the maximum error, though under special circumstances WPA might have rather large errors. It will also be possible to introduce virtual demand using the WPA and CCPA, since demand points with no weight (weight equal to zero) can have a given weight allocated.

## 9.3 Evaluating WPA and CCPA

A minor test is carried out using 5575 accidents, to get an idea of how large errors each method will generate. The first test concerns WPA and CCPA using quadratic cells and a distance between centers of 1000 meters. The WPA only took a few seconds to run, whereas the CCPA worked for about 25 minutes to reach the result. The results are shown in Figure 9.

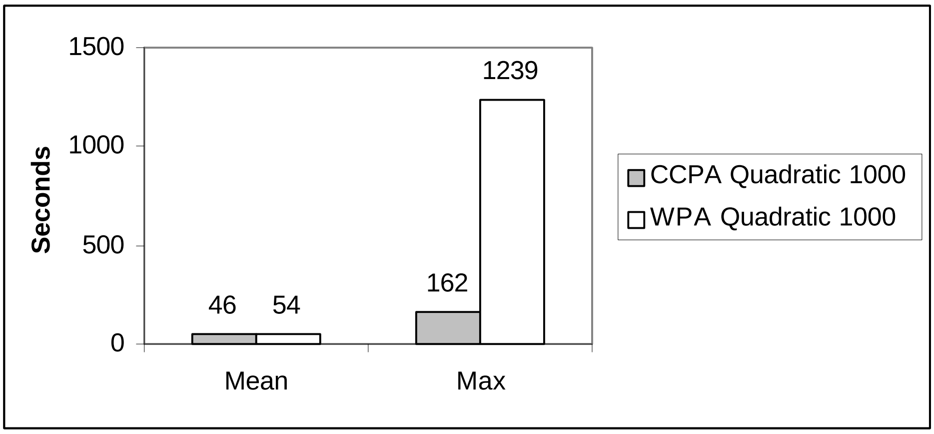



**Figure 9. Mean and maximum travel times for WPA and CCPA using a quadratic 1000 meter tessellation.**

The test was carried out using the two algorithms on the 5575 accident dataset, using the same starting point, hence the only difference in the algorithms is that in WPA an accident is allocated to closest demand point in terms of Euclidean distance, and in CCPA it is the closest using the travel time on the network. In order to compare the results of the WPA with the CCPA the travel time on the network between the accidents and the demand points to which they were allocated is calculated.

As can be seen in Figure 9 mean values are not that different, though as could be expected the WPA is the highest. The maximum values on the other hand clearly show the problem with the WPA with the maximum value being 1239 seconds, over 20 minutes, opposed to the CCPA, which is only 162 seconds, less than 3 minutes. It only seems to be a minor portion of the accidents that have been allocated “very poorly”, even at the 75% quartile there is only a difference of 7 seconds, as can be seen in Figure 10.

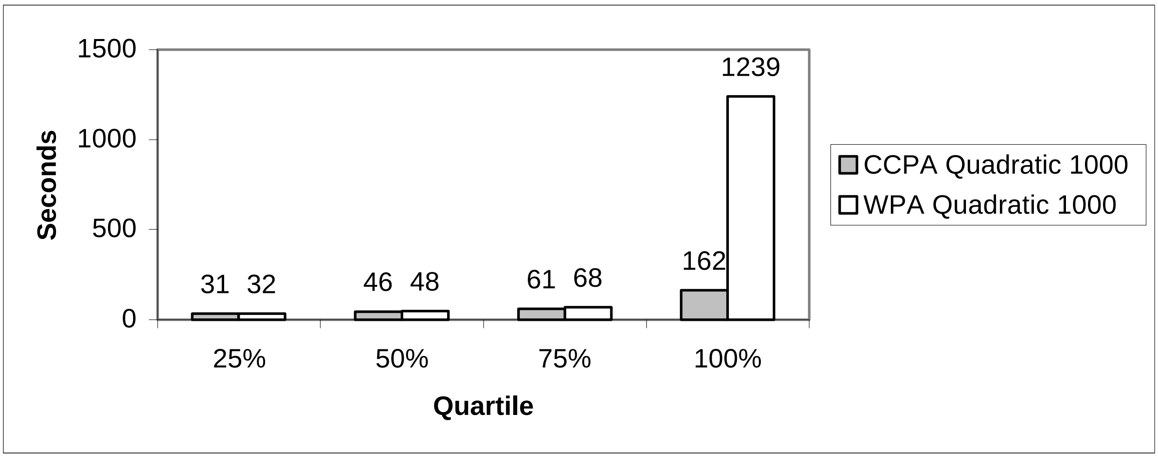



**Figure 10. Quartile travel times for WPA and CCPA using a quadratic 1000 meter tessellation.**

In fact there are only 16 accidents of more than 5 minutes and only 3 above 10 minutes for the WPA. However for other areas with a different geography and road network results are likely to differ considerably. An example of this type of error is presented (using ambulances and demand points) in Figure 22 on page 55, which in general is due to the extent of (or lacks of extent of) the road network.

The second test is a comparison of the CCPA using hexagons and quadrants. The distances of the to quadratic tessellations are 1000 and 2000 meters, and for the hexagonal tessellation it is 1070 and 2140 metes. The difference in meters is necessary to operate approximately with the same number of demand points, so that a fairly unbiased comparison can be made.

| Dataset | # Demand points |
|---|---|
| 1000 Quadratic | 3812 |
| 1070 Hexagon | 3851 |
| 2000 Quadratic | 1041 |
| 2140 Hexagon | 1057 |

**Table 2. Number of demand points in each tessellation.**

In both cases the quadratic tessellation have a little less demand points. Hence the hexagonal tessellation is expected to be the best. The average (mean) and maximum values from the test can be seen in Figure 11.

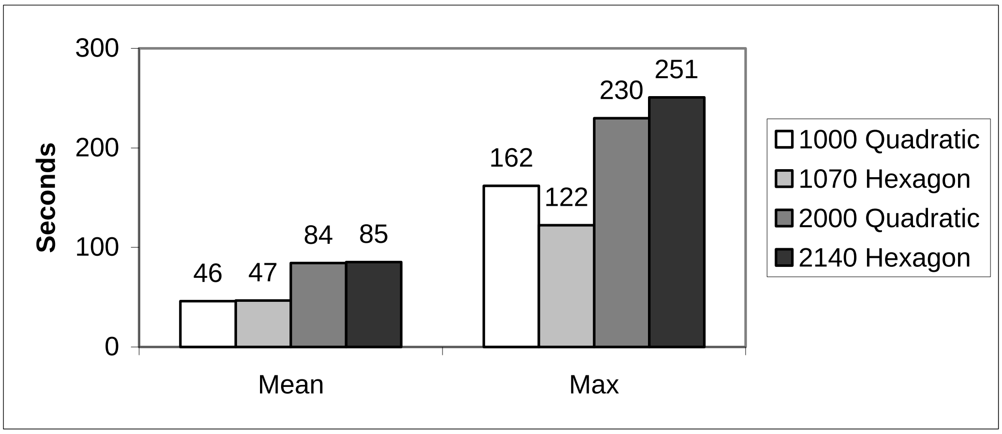



**Figure 11. Mean and maximum travel times for CCPA using four different tessellations.**

Neither seems to stand out significantly. The mean values are almost the same, and the maximum value is lowest for the Quadratic tessellation for the short distance and lowest for hexagonal tessellation for the long distance. It is a bit interesting that the mean and maximum values do not double when the distance is doubled. Why this is so is a bit unclear to me.

## *9.4 Conclusion on demand points*

CCPA seems to be the algorithm to use when there is time enough, whereas the WPA can be used “on the fly” for instant answers. It does not seem to be very important which type of tessellation is used they both perform quite well.  The rest of this project, except for chapter 10: “Accident Analysis”, will use the 2140 CCPA hexagon tessellation.

# 10 Accident Analysis

*The purpose of this chapter is to provide an understanding of the data.*

It is necessary to understand the data for several reasons: First of all it is very time consuming if not impossible to handle 60,000 accidents at a time, so some kind of summation or grouping of the accidents is necessary. Secondly, it will be difficult to comment on solutions to the ambulance allocation, and to get ideas on how to improve the allocation if the accidents were just numbers in a table. Furthermore, a good breakdown of the accidents can help provide better solutions in the end. The accidents will be dealt with as follows:

1) Accidents in general.
    a. Where do accidents happen and when.
2) Dividing data into time periods.
    a. Which time periods to compare.
    b. How to compare.
    c. Do the accidents differ as to where they happen and when?

In order to get a general feeling of data, the first step is a geographic plot of accidents and their distribution over time, since time and place are the main attributes of the accidents. Data is from the period 11th January 2000 22:42 to 18th March 2002 22:43.

## 10.1 Geography

A quick glance at Figure 12 and Figure 13 leaves no doubt: Accidents mostly happen in towns, where people live and work. Figure 12 is probably as good as any satellite photo to identify the positions of towns on Funen. Grid cells are one by one kilometer. There is no correction for cells that contain lakes, sea and other areas that cannot contain accidents since there are no roads.

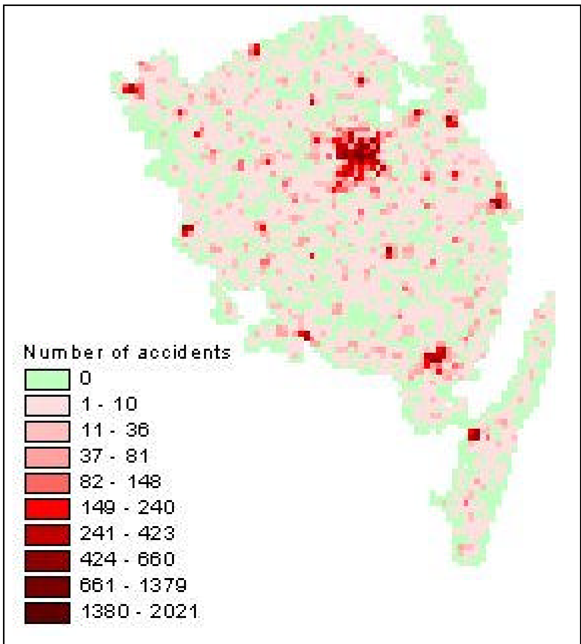



**Figure 12. Accidents per km$^2$ over the period.**

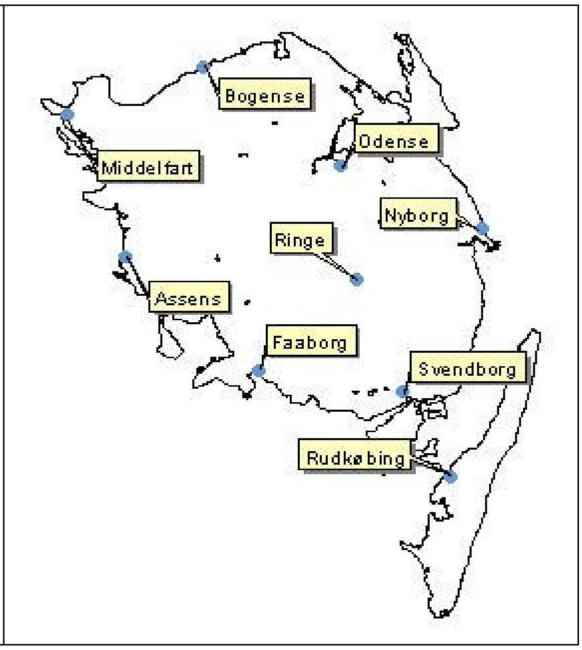



**Figure 13. Positions of major towns on Funen.**

There are of course other towns on Funen than those shown in Figure 13, most of which can be found as read spots in Figure 12. The classification used for Figure 12 is called “Natural Breaks”, it is based on a rather complicated statistical formula (Jenk’s optimization) [AM p. 104]. By classifications is meant the choice of limits on each group/class. The first group though has been changed from the original classification by being divided into two groups, one including all cells with the value zero and one including all cells with at least one and no more than 10 accidents. Using the same classification it is clear that most accidents happen in a relatively small number of cells (Figure 14).

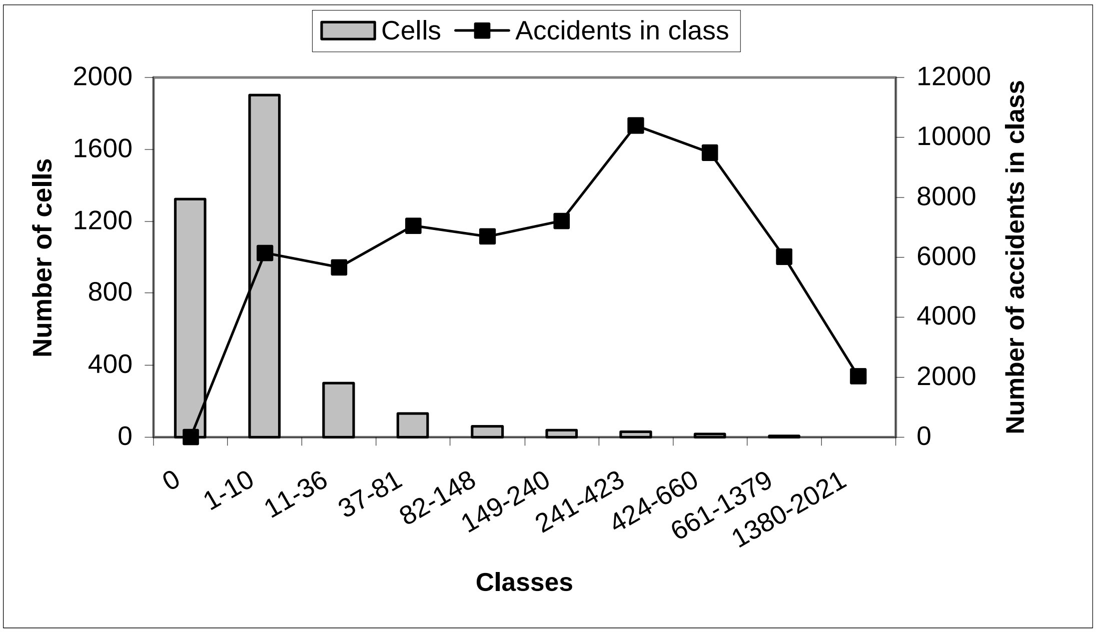



**Figure 14. Number of cells and number of accidents in each class.**

About 50 percent of the accidents happen in 2% of the cells, in other words there is a great concentration of accidents in smaller areas as could also be seen in Figure 12 as dark areas and in Figure 13 as towns and cities.

## 10.2 Time

The separation of accidents in time over the year, month , week and day will be dealt with in the following. Figure 15 presents the number of accidents over year divided into month.

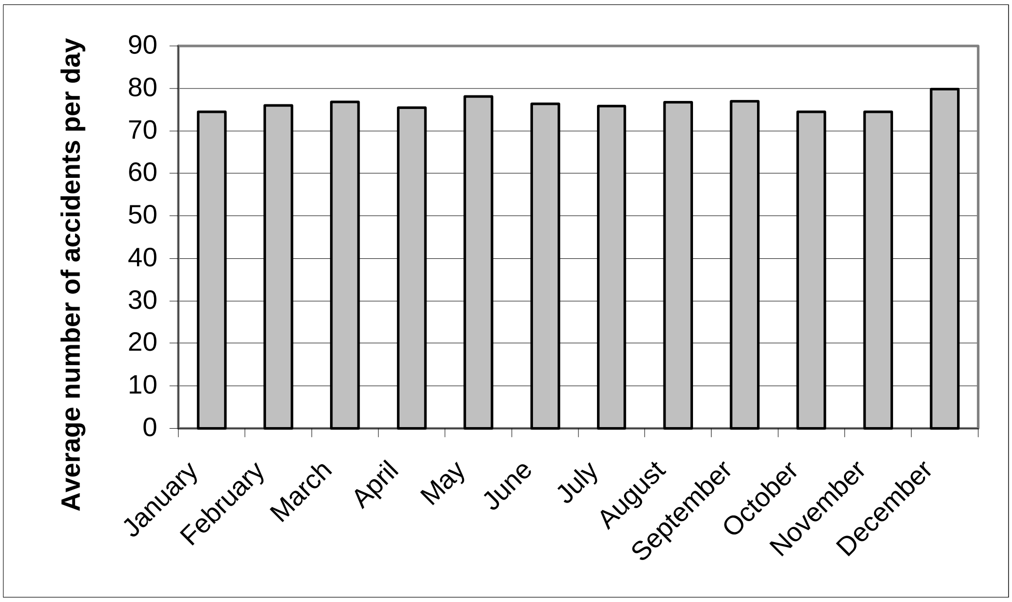



**Figure 15. Average number of accidents per day in a standard month (equal number of Mondays, Tuesdays, Wednesdays, Thursdays, Fridays, Saturdays and Sundays).**

As can be seen in Figure 15, there is no particular difference in the number of accidents over the year. The largest deviation from the average month (76,3 accidents per day) is December with an extra 5%, whereas the lowest month is January with 2% less.

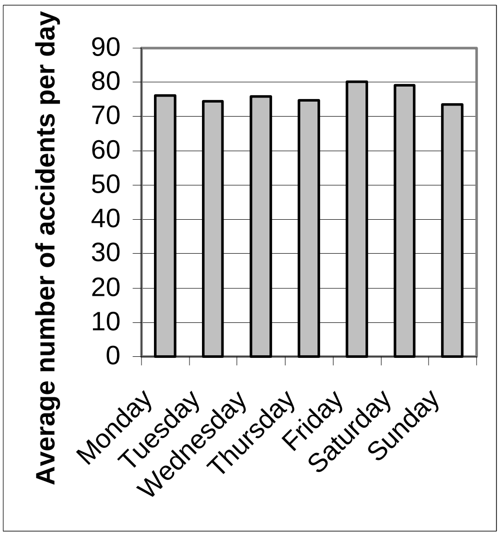



**Figure 16. Average number of accidents per day.**

The deviations are somewhat larger when looking at days, Figure 16. Friday has 5% more accidents than the average day and Sunday has the lowest number of accidents, 4% less than the average day. As for the months there does not seem to be any significant differences between some days or a smaller group of days. Friday and Saturday do have the two highest number of accidents, but not by much.

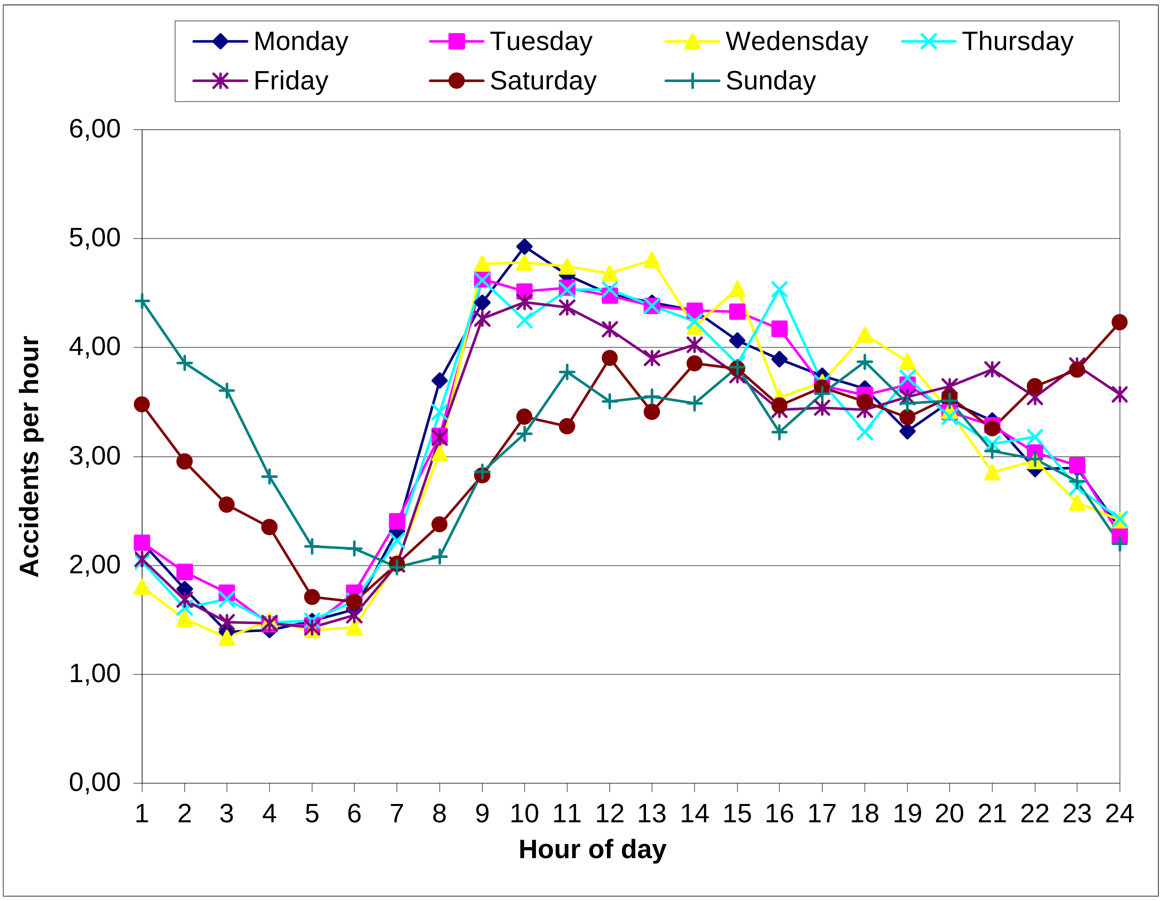



**Figure 17. Average number of accidents per hour over a mean day. Hour of day 1 if from 0 am to 1 am.**

The variation over the week on an hourly basis, rather than on a daily basis, shows something more interesting (Figure 17). It does not come as a surprise that the lowest number of accidents happen when people are at sleep, but that the peak is between 9 am and 2 pm in the weekdays when people are at work, is a bit unexpected. Though it might indicate that a quite a few accidents happen at work. There also seem to be two different patterns: Weekend and weekdays, or more precisely Friday to Sunday afternoon shows one pattern, whereas Sunday to Friday afternoon has another pattern.

## *10.3 Dividing Data Into Time Periods*

Why not use data as it is: One big distribution/population? First of all, there is so much data that it would be quite time consuming to work with it all at one time. Secondly, placing ambulances as “optimal” as possible at a given time will be improved anyhow since those areas with the best service will also be those with the potentially highest risk at the given time.

It is reasonable to assume that accidents do not happen at a completely random basis. Some supernatural power does not pick the time and place of the accidents. As discussed in sections: “10.1 Geography” and “10.2 Time”, accidents only happen where people are. It therefore seems logical to divide data into groups according to people’s dynamic geographic whereabouts. Since is no standard pattern in terms of geography and time exists, I have chosen to divide the data into the following periods:

A. Time of the day/week:
   i. At work, Weekdays from 8 am to 5 pm
   ii. Off work, Weekdays from 5pm to 8 am
   iii. Weekend

B. Time of the year:
   i. Winter (December, January and February)
   ii. Spring (March, April and May)
   iii. Summer (June, July and August)
   iv. Fall (September, October and November)

Of course many other periods could have been used. Taking the late hours, when people sleep into, consideration could be an option. The choice of the periods is based on the assumption that if there is a significant difference in geographic occurrence of accidents, it would prove most significant when comparing the periods above.

## *10.4 Comparing Spatial Distributions*

This section will discuss the problems of comparing geographically distributed data, i.e. spatial distributions.

### 10.4.1 Unbiased Comparison

The first problem is to introduce an unbiased measure that can be used to compare two different time periods with a different amount of accidents. Table 3 shows that there are big differences in the number of accidents, hours and accidents per hour in the data available. Which complies well with the results from Figure 17.

| Time period | # Days | # Hours | # Accidents | # Accidents per hour |
|---|---|---|---|---|
| Winter at Work | 162 | 1458 | 6307 | 4.33 |
| Winter Off Work | 162 | 2430 | 6051 | 2.49 |
| Winter Weekend | 65 | 1560 | 5064 | 3.25 |
| Spring at Work | 143 | 1287 | 5576 | 4.33 |
| Spring Off Work | 143 | 2145 | 5577 | 2.60 |
| Spring Weekend | 57 | 1368 | 4369 | 3.19 |
| Summer at Work | 131 | 1179 | 4870 | 4.13 |
| Summer Off Work | 131 | 1965 | 5114 | 2.60 |
| Summer Weekend | 51 | 1224 | 4078 | 3.33 |
| Fall at Work | 129 | 1161 | 5005 | 4.31 |
| Fall Off Work | 129 | 1935 | 4797 | 2.48 |
| Fall Weekend | 51 | 1224 | 3906 | 3.19 |

**Table 3. Data from time periods.**

The number of accidents in a given period as such is not interesting to project, since it is not the objective to determine an overall resource allocation strategy, but to deal with the resources at hand as well as possible. Hence a reasonable measure seems to be the number of accidents assigned to a cell point as a fraction of the total number of accidents for the entire period or simply as an index. An index could be:

$$\frac{\textit{number of accidents within cell}}{\textit{number of accidents in period}} \;*\; \textit{number of accidents per year} \;=\; \textit{new cell value}$$

This would also render the index more meaningful. Each cell value would then represent the number of accidents per year in that cell just as if accidents happened through an entire year as they do in the actual period. The number of accidents per year is set to 30.000 (on Funen).

### 10.4.2 How to Compare

The second problem is how to compare the various time periods, or rather how to determine when there is a significant geographic difference: One way of comparing the data is simply by visualizing it using the GIS software and see if there appears to be a difference and preferably finding an explanation of why. Another way could be to use statistics to verify the results. However, it has not been possible to develop or find a model that suits this problem.

The statistical models dealing with spatial problems are, to this author’s knowledge, mostly used in geology and fields dealing with sample data and then trying to interpolate what have not been measured. Models such as Inverse Distance Weighted (IDW) and Kriging [SAM p. 92] are not designed to deal with discrete events like accidents, or entire populations as in this project, but to interpolate a surface from a minor set of measurements. An alternative could be to use a classical homogeni/chi^2 test, but these tests are normally used to compare small samples from a large section of a population, and not an entire population like in this project. Since all data, or at least what could be geocoded, is used in this project we know the distribution and we also know that there is a difference, the question therefore is: “Does the difference matter when placing ambulances?” or in other words “Is the difference so big that it should be taken into consideration?” Of course, it could be a case of an underlying distribution of accidents, and then the more traditional statistical methods could be used. But what defines the underlying distribution, does such a distribution make sense?

Alternatively, two time periods can be compared by comparing the optimal solutions for placing ambulances, or rather calculating the difference i.e. using the optimal solution found to the one problem on the other problem, and then calculating the new solution value. Nevertheless since the number of ambulances vary, i.e. for each number of ambulances there is a new comparison, and since finding optimal solutions for large number of ambulances is more or less impossible (chapter 11: “Models”), this method does not seem to be applicable.

I have chosen to rely mostly on the visual inspection of data, for the pure reason that if the difference is so small that it cannot be visualized, it is probably also so small that it will be of no significance, and it will be difficult to convince anyone that a difference is actually present.

### 10.4.3 Which to Compare

With 12 different time periods the number of possible comparisons are 66, and using visual inspection to compare the periods, it may be an advantage to establish which periods are expected to be different and then compare those first to test if the method works at all.

The major problem is that what we are looking for are “large” geographical changes. Such changes could of course be calculated when comparing the difference between various divisions of Funen, such as residential, industrial and commercial divisions or divided into municipalities. The residential, industrial and commercial divisions seem to be the most reasonable choice, however such divisions are seldom so unique that they comprise only one type of activity, especially commercial and residential areas tend to mix in the cities. That there should be a difference between municipalities in the given periods does not seem obvious, a difference would probably be difficult to explain as anything else than random variation.

A completely different way is to divide Funen into a grid and then sum up the absolute difference over all the cells between each of the time periods. The ones with the highest difference might be worth investigating. An alternative is to use the same grid division, but simply pick the periods, which are expected to have the biggest difference.

The result of the “sum of absolute difference” method is shown in “Appendix II” The one with the highest score was “summer at weekends” compared with “fall at work”. The subjective guess is that there is a big difference between “summer at weekends” and “winter at weekends”, based on the assumption that people will be attracted to different areas, such as beaches and nature in general during summer, whereas during winter they are more prone to stay indoors.

## 10.5 Visual Inspection of Accidents

The visual inspection is carried out using the grid features of spatial analysis. Each grid is a square of one by one km, i.e. one km$^2$. There is no correction for cells that contain lakes, sea and other areas that cannot contain accidents since there are no roads.

Figure 18 shows the difference between accidents in "summer weekends" and "winter weekends" and Figure 19 shows the difference between accidents in "summer weekends" and "fall at work".

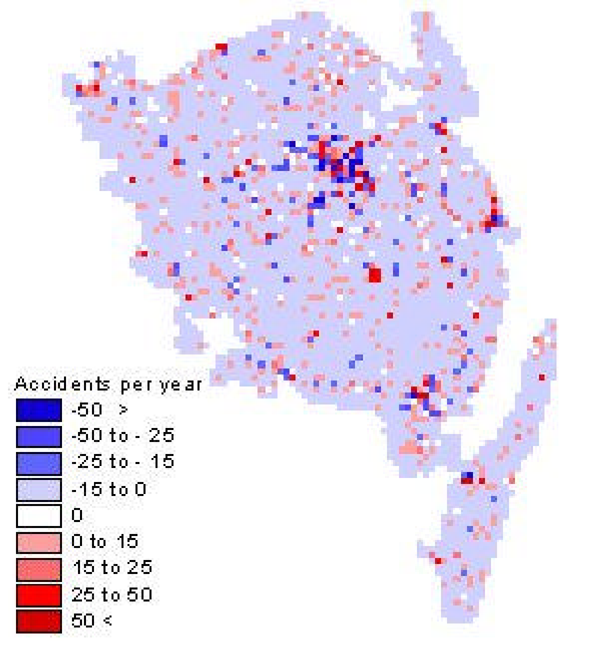



**Figure 18. Accidents per km$^2$: "summer weekends" minus "winter weekends".**

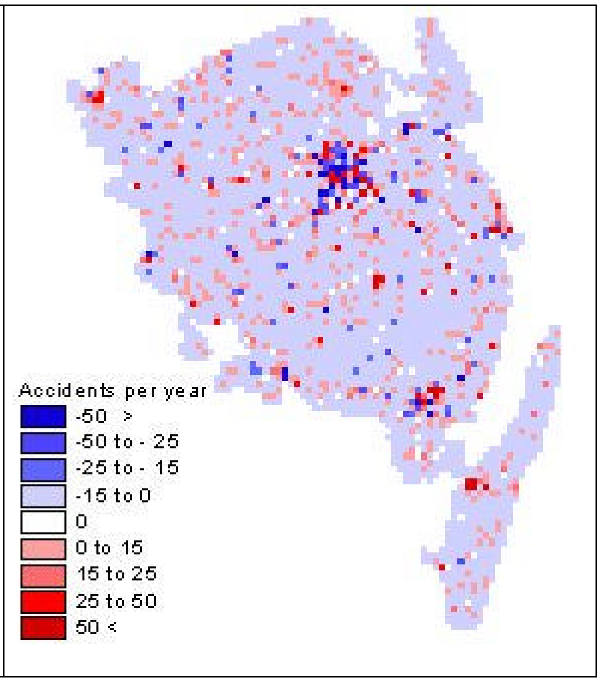
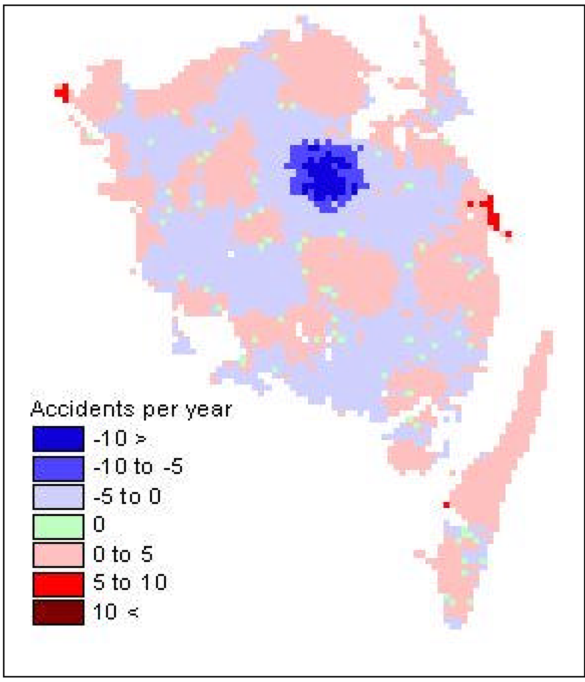



**Figure 19. Accidents per km$^2$: "summer weekends" minus "fall at work".**

It is very difficult to see whether there are any significant differences, however for both plots there is one thing of interest, there is a lot of light blue, hence there seem to be slightly less accidents in areas where very few people live or work. The small group of cells (three to four cells) in the middle of the plot is Ringe, where a festival is held every summer with many people attending, therefore it makes sense that there are more accidents. The figures most of all seem to represent some general variation, with one cell showing a negative difference and the cell next to it showing a positive difference. To overcome that problem a new method is introduced. By using this each cell value now becomes the mean of any cell within five kilometers (it could of course be any distance), in this way the general variation should be removed. The results are shown in Figure 20 and Figure 21.



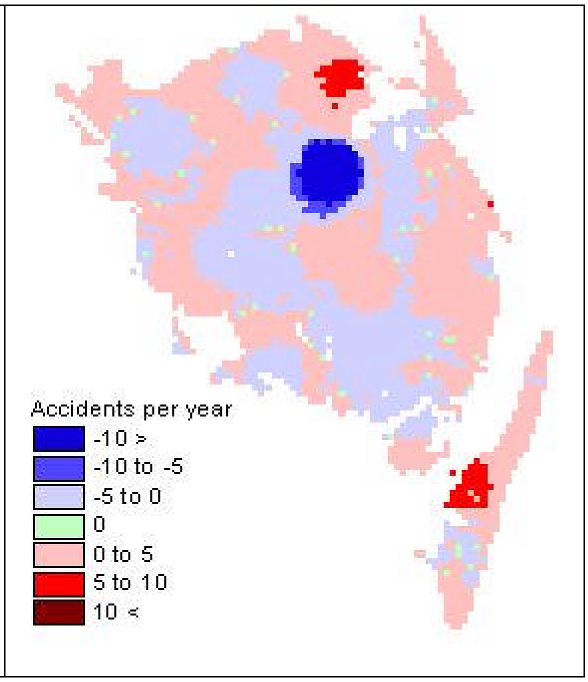




**Figure 20. Accidents per km$^2$ (mean within five km radius): "summer weekends" minus "winter weekends".**

**Figure 21. Accidents per km$^2$ (mean within five km radius): "summer weekends" minus "fall at work".**

The new plots are easier to interpret. For both plots there seems to be a decrease in Odense and an increase on some of the costal areas, especially Lange Land (the long island in the lover right corner) seems to have an increased its number of accidents. Evidently there is a difference, and it even seems to exceed the general variation. However, it is difficult to see how this will affect the allocation of the ambulances, other than perhaps reduce the number of ambulances in Odense. There will be no further investigation into the geographic occurrence of accidents in time.

### *10.6 Other Methods*

During this project other methods have been suggest by the group following the project, for example:

- Investigating a smaller area than the entire Funen
- "Simulate" the accidents (let them show up on the computer screen one by one) to see if a pattern over time could be recognized.

These methods have not been tried due to the difficulties of the visual inspection, such as proving whether or not a difference in a geographic pattern of accidents makes a difference when placing ambulances. This is also likely to apply to the alternative methods.

## 10.7 Conclusion on Accident Analysis

There seems to be a significant difference in the geographic occurrence of accidents over time, nevertheless it has not been possible to establish whether or not the difference is significant enough to be important for the allocation of ambulances.

In the rest of this project, data from the period “Spring at Work” will be used to place ambulances. There is no particular reason for using this specific period. The dataset used is the snapped hexagonal tessellation with a distance of 2140 meters between the demand points before snapping. Four demand points has been removed from the dataset since they were inaccessible. One was at the end of one of the bridges (“The New Small Belt Bridge”), which was cut of when creating the road network of Funen, making it impossible to reach due to a one-way restriction. Only one accident was allocated to the demand points, three of them had a weight of zero. More on the dataset in chapter 13: “Test of Solution Methods”.

# 11 Models

*This chapter deals with defining the mathematical models used in this project. The models are the multi facility location allocation problem (MFLA), the p-median and p-center problems.*

All models have the objective to place $p$ ambulances in the most optimal way. As discussed earlier one of the key problems is to define optimality. Three different models with three different objectives will be presented. In addition it will be discussed how these models can be combined with a variety of limitations, which may be of relevance to this and future projects.

The models can be divided into two categories, continuous and discrete models. The difference between the continuous model and the discrete model is that in the continuous model it is feasible to place an ambulance anywhere, it could be on a lake, in a forest, anywhere on the map so to speak, whereas the discrete model is given a set of possible locations, this could be the demand points or any other set of points.

The continuous model will have the objective to minimize the average weighted Euclidean distance. The discrete model will have the objectives of minimizing the average weighted travel time on the network and minimizing the maximum travel time on the network.

## 11.1 Continuous Model

In order to state the model in terms of math, a few definitions are needed:

Ambulances are considered to be facilities, represented with the letter $i$, customers are referred to as demand points (accidents) and are represented by the symbol $j$. The number of ambulances is $p$.

The following notation is used:
$I$: The set of all ambulances, with $|I| = p$.
$i$: A specific ambulance
$\boldsymbol{f}_i$: The coordinates for the $i$'th ambulance
$J$: The set of all demand points
$j$: A specific demand point
$\boldsymbol{a}_j$: The coordinates for the j'th demand point.
$J_i$: The set of demand points allocated to the $i$'th ambulance
$w_j$: The weight associated with demand point $j$, i.e. the number of accidents assigned to that demand point.
$d_2(\boldsymbol{f}_i,\boldsymbol{a}_j)$: The Euclidean distance between the $i$'th ambulance and the j'th demand point

The single continuous model used in this project has the objective of locating the ambulances anywhere in continuous space, i.e. not necessarily on a road or even on land, such that the average weighted Euclidian distance to the demand points is minimized. The objective function

$$\text{Minimize} \sum_{i \in I} \sum_{j \in J_i} w_j \cdot d_2(\boldsymbol{f_i}, \boldsymbol{a_j})$$

is actually a minimization of the total weighted Euclidean distance, but since the number of demand points (and weight, i.e. accidents) is a constant it corresponds to minimizing the average weighted Euclidean distance.

### 11.1.1 The Multi Facility Location Allocation Problem - MFLA

The formulation of the Multi Facility Location Allocation problem (MFLA) is partly adapted from lecture notes by Henrik Juel [HJ].

Minimize
$$\sum_{i \in I} \sum_{j \in J_i} w_j \cdot d_2(\boldsymbol{f}_i, \boldsymbol{a_j}) \qquad (11.1.1)$$

Subject to
$$(J_1, \ldots, J_p) \text{ is a partition of } J \qquad (11.1.2)$$

*or*

$$(J_1 \cup J_2 \cup \ldots \cup J_{p-1} \cup J_p) = J \qquad (11.1.3)$$

$$J_{i1} \cap J_{i2} = \emptyset \qquad \forall i1 \neq i2, \{i1, i2\} \subset \{1, 2, \ldots, p\} \qquad (11.1.4)$$

Equation (11.1.1) states that the sum of the weighted Euclidean distances between each ambulance and the demand points allocated to it, and then summed over all ambulances, should be minimized. Which in essence is the same as minimizing the average weighted Euclidean distance, since the number of demand points is fixed. The allocation of the demand points is defined in (11.1.2) or (11.1.3) and (11.1.4) and ensures that all demand points are included in the solution. (11.1.3) and (11.1.4) are actually just the definition of (11.1.2), but are included to emphasize what a partition is.

The decisions to be made are the location of the ambulances, $\boldsymbol{f_i}$, for all $i \in I$. Each demand point is allocated to the nearest ambulance.

A practical experiment by Eilon, Watson-Gandy and Christofides showed that a problem with 50 customers (demand points) and 5 facilities (ambulances) had 61 local minima, the worst of which had a deviation of 41 % percent from the best solution. Megiddon and Supowit proved in 1984 that the problem is NP-hard. [FL p. 17].

The MFLA model lacks many vital elements. The most obvious one is the fact that an ambulance can be located at any point, also at sea, in lakes, in the middle of a building, or in a forest. In most cases though, it would be expected still to be a good solution even if each ambulance is moved a little, i.e. to the nearest road. Furthermore, the Euclidean distance is a rather rough estimate of the response time of an ambulance. The Euclidean distance and the response time are without doubt correlated since it takes more time to travel a longer distance at the same speed. However the model might allocate a demand point to the ambulance on one side of the river only 1 km away from the relevant place on the other side of the river, regardless that the nearest crossing is 30 km away. So despite a short Euclidean distance, the response time becomes tremendous. An example of this is shown in Figure 22.

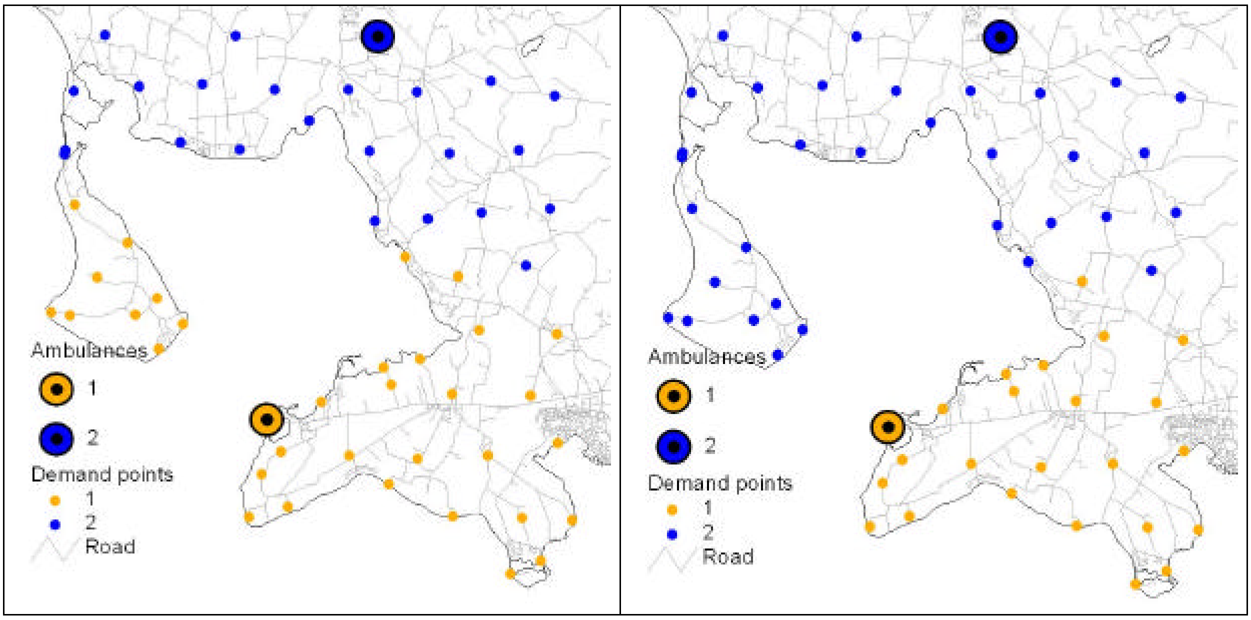



**Figure 22. Illustration of possible risk when using the Euclidean distance as a measure for response time The left picture shows allocation when mimimizing the average weighted Euclidean distance, while the right picture shows the result when minmizing the average weighted response time on the road network. This is a contructed example.**

The model could be expanded with elements such as barrier definitions, i.e. forbidden areas where the ambulances are prohibited, such as lakes and forests. This is an area in which the GIS features might prove useful, since it could easily be determined using GIS whether a solution is feasible or not.

Other likely additions to the model could be capacity constraints, i.e. limitation on the number of potential accidents that an ambulance is allowed to serve. Leaving capacity constraints out allows a single ambulance to cover a large amount of demand, like in Odense. On the other hand, when using capacity constraints demand points might not be allocated to the nearest ambulance, even though it is the closest ambulance, which will respond. A problem which makes the use of capacity constraints a little odd, although it might not prove all that important in a practical situation, since the number of demand points not allocated to the nearest ambulance is expected to be relatively small.

The mathematical expressions of the barrier definitions and the capacity constraints are not included, since it would require a complete redefinition of the entire model.

## *11.2 Discrete Models*

The discrete models only consider a certain fixed number of potential sites as possible locations for ambulances, ***ambulance points***. This could be any set of points, i.e. the demand points, the Falck stations or perhaps a set of predefined dispatcher points which are expected to be good strategic positions such as in the cities, near intersections on fast roads etc.

Two different objectives will be considered. The first is to minimize the average response time (i.e also total) from the ambulances to the demand points, the p-median problem. The second is to minimize the maximum response time, p-center problem.

The response time from ambulance point $i$ and demand point $j$ is defined as $d_{ij}$. The response time, $d_{ij}$, could be any number, any sort of cost or distance. In this project the travel time on the road network is used for $d_{ij}$.

### 11.2.1 p-median Problem

The symbols $w_j$, $p$, $J$, and $j$ denote the same meaning as they did for the continuous model. $I$ and $i$ have changed their meaning slightly and two new definitions are added for $x_i$ and $y_{ij}$:

$I$: Is the set of all ambulance points $|I| \geq p$.
$i$: Is a specific ambulance point.
$x_i$: Is 1 if an ambulance is placed at the $i$'th ambulance point and 0 if not.
$y_{ij}$: Is 1 if the demand point $j$ is allocated to the ambulance at ambulance point $i$, and 0 if not.

Formulation partly taken from “Facility Location: Applications and Theory” [FL p. 90-91]

$$\text{Minimize} \quad \sum_{i \in I} \sum_{j \in J} w_j \cdot d_{ij} \cdot y_{ij} \qquad (11.2.1)$$

$$\text{Subject to} \quad \sum_{i \in I} x_i = p \qquad (11.2.2)$$

$$\sum_{i \in I} y_{ij} = 1 \qquad \forall j \in J \qquad (11.2.3)$$

$$y_{ij} - x_i \leq 0 \qquad \forall i \in I, j \in J \qquad (11.2.4)$$

$$x_i \in \{0,1\} \qquad \forall i \in I \qquad (11.2.5)$$

$$y_{ij} \in \{0,1\} \qquad \forall i \in I, j \in J \qquad (11.2.6)$$

Equation (11.2.2) makes sure that exactly $p$ ambulances are used, and (11.2.3) ensures that each customer is allocated to one ambulance. (11.2.4) Prohibits demand points from being assigned to ambulance points with no ambulance. The integer requirement on $x_j$ (11.2.5) defines that an ambulance can only be placed at one demand point, since it is not possible to place the one half of an ambulance at one demand point and the other half at a different demand point. (11.2.6) Ensures that the demand from one demand point is allocated to one and only one ambulance. If necessary (11.2.6) can be relaxed (removed), however many times when solving the problem it will be easier to just allocate one demand point to one ambulance, and in an optimal solution a demand point will always allocate its entire weight to the nearest ambulance. If (11.2.6) is not meet it might also be difficult to visualize the result.

The decision to be made is at which ambulance points are the *p* ambulances placed. Each demand point allocated to the nearest ambulance (ambulance point with an ambulance on it).

It is also possible to expand this model. Adding constraints on the maximum allowed travel time, i.e. an ambulance cannot cover a demand point which is more than $d_{max}$ minutes away etc. is fairly easy. Either all the $y_{ij}$ with a corresponding $d_{ij} > d_{max}$ are banned from the solution or all the $d_{ij} > d_{max}$ are substituted with a number so large that they will not be part of the optimal solution.

It is also possible to add capacity constraints to the problem, i.e. setting an upper limit on the potential number of accidents covered by the ambulance. If the capacity for an ambulance is set to *k* then the constraints can be formulated as,

$$\sum_{j\in J} y_{ij} \cdot w_j \le k \qquad \forall i \in I \qquad (11.2.7)$$

it is of course required that

$$\sum_{j\in J} w_j \le p \cdot k \qquad (11.2.8)$$

to ensure that the total demand to be met does not exceed the total capacity of the ambulances, in order to ensure a feasible solution. The capacity constraint for the p-median problem will have the same disadvantages as those for the MFLA.

### 11.2.2 p-center Problem

The p-center problem is a minimax covering problem, i.e. its objective is to minimize the maximum response time represented by the letter *W*. It is important to notice that the weights play no role for the basic formulation of this problem, since we are minimizing the maximum response time, not the maximum cost. However if capacity constraints were to be applied, the weight would play a role.

$$\text{Minimize} \qquad W \qquad (11.2.9)$$

$$\begin{aligned}
\text{Subject to} \quad & \sum_{i\in I} x_i = p & & (11.2.10)\\
& \sum_{i\in I} y_{ij} = 1 & \forall j \in J \quad & (11.2.11)\\
& y_{ij} - x_i \le 0 & \forall i \in I, j \in J \quad & (11.2.12)\\
& W - \sum_{i\in I} d_{ij} \cdot y_{ij} \ge 0 & \forall j \in J \quad & (11.2.13)\\
& x_i \in \{0,1\} & \forall i \in I \quad & (11.2.14)\\
& y_{ij} \in \{0,1\} & \forall i \in I, j \in J \quad & (11.2.15)
\end{aligned}$$

The model is very similar to the one from the *p*-median problem. It only differs in the objective function and the new constraint (11.2.13), which defines *W* as the maximum travel time between any of the demand points and the ambulance to which they are allocated. That *W* defines the maximum travel time can be seen by examining (11.2.13) more closely. If *W* is not the maximum travel time it means that there exists a $\Sigma d_{ij} \cdot y_{ij}$ which is greater than *W*, hence there exists a travel time $d_{ij} > W$, since $y_{ij}$ is either one or zero. Given that (11.2.13) must be met there cannot exist a $d_{ij} > W$, hence $d_{ij} \leq W$. Because *W* is minimized (11.2.9) it will be equal to the largest $d_{ij}$, hence a minimization of the maximum distance.

Response time limits like (11.2.7) are obviously not relevant for this problem, since the maximum requirement will be met if *W* is low enough. Capacity constraints are the same as for the p-median problem.

### 11.2.3 Complexity of the p-median and p-center problems

The number of possible solutions can easily be found. If there are *n* demand points, $n = |J|$, and *p* ambulances is corresponds to the number of different ways which p ambulance can be placed on *n* locations. Which is the same as the number of possible combinations to pick *p* elements out of n elements:

$$\binom{n}{p} = \frac{n!}{p! \cdot (n-p)!}$$

with 1000 demand points and 10 ambulances it would be

$$\binom{1000}{10} = \frac{1000!}{10! \cdot (1000-10)!} \approx 2.6341 \cdot 10^{23}$$

and as mentioned in section 5.2.1: “Computational Complexity” it would take 8 million years, even if one billion solutions were tested every second, to check them all. Though solutions with fewer ambulance has less possible solutions, the number of solution increases almost by a 1000 (*n*) every time an ambulance is added to the problem, number for shown in Table 4, the “Solution time” is the amount of time it takes to check all solutions with the given number of solutions checked per second.

| *p* | *Possible solutions* | *“Solution time“ One billion a sec.* | *“Solution time“ One million a sec.* | *“Solution time“ One thousand a sec* |
|---|---|---|---|---|
| 1 | $10^3$ | Less than a second | Less than a second | One second |
| 2 | $5.00 \cdot 10^5$ | Less than a second | Less than a second | 500 seconds |
| 4 | $4.14 \cdot 10^{10}$ | 41 seconds | More than 11 hours | More than a year |
| 6 | $1.37 \cdot 10^{15}$ | 16 days | More than 43 years | 43442 years |
| 8 | $2.41 \cdot 10^{19}$ | 764 years | 764 thousand years | 764 million years |
| 10 | $2.63 \cdot 10^{23}$ | 8 million years | 8 billion years | 8 trillion years |

**Table 4. Number of possible solutions and the time it takes to check them all.**

Despite the many number of possible solution, the p-center and p-median problems can be solved in polynomial time for fixed values of *p* which is relative easy to see since

$$\binom{n}{p} \leq n^p$$

though as seen in Table 4, even for small number of *p* it is practical impossible, even with extreme computing power. For varying number of *p* the problems are NP-hard (Garey and Johnson, 1979) [FL. p. 91].

# 12 Solution Methods

*This chapter deals with the methods used for solving the models in Chapter 11: "Models".*

As mentioned in chapter on  (OR) there are many different ways of solving optimization problems. For the three problems at hand:

- Multi facility location allocation problem with Euclidean distance
- p-center problem
- p-median problem

Heuristic methods will be applied to the problems in this project, because to the amount of demand points and ambulances in the problem. The reason for not applying exact solution methods is that the available exact methods are very time consuming, for large problems like those dealt with in this project.

## *12.1 Solving the MFLA*

The Multi facility location allocation problem with Euclidean distance (MFLA) will be solved using a well known heuristic the: Multi Restart Cooper algorithm (MRC). Despite that the model for the MFLA is a very primitive model of the real life problem, it is very difficult to solve to optimality for larger instances. Cooper proved in 1967 that the objective function is neither convex nor concave, it may have a large number of local minima [FLMM p. 14].

The idea is quite simple. A procedure known as the Weiszfeld procedure renders proven optimal solutions for single facility location problems with Euclidean distance. The Cooper algorithm uses the Weiszfeld procedure, by randomly selecting *p* different points in continuous space as ambulances, and allocating each demand point to the closest ambulance. Now the problem can be divided into *p* single facility location problems, which can each be solved to optimality. Then the demand points are reallocated to the closest ambulance. This is done until no further improvement is found, hence the solution is a local minimum. A diagram of the algorithm is presented in Figure 23.

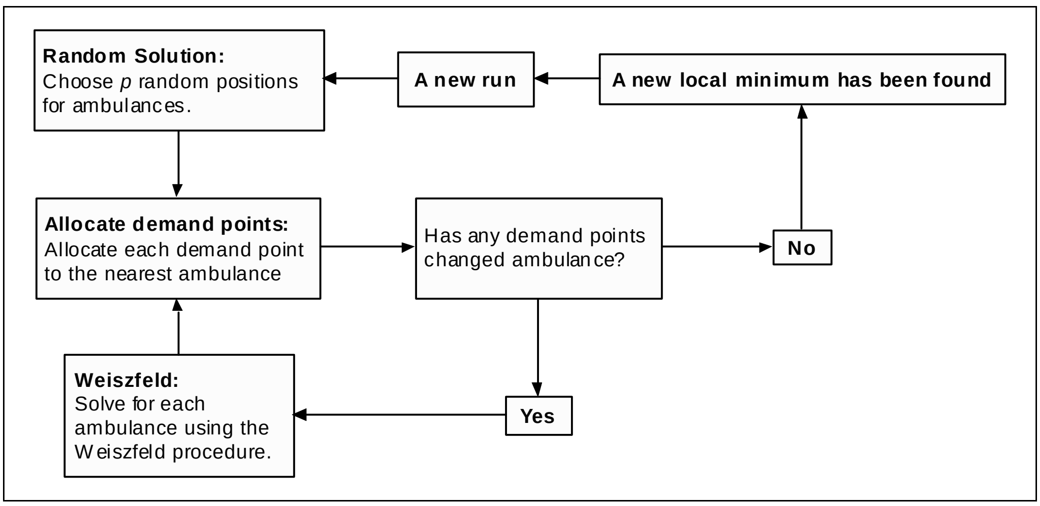



**Figure 23. The Multi Restart Cooper algorithm.**

The MRC without the restart is just called the Cooper algorithm. When Cooper is given the same starting solution it always yields the same local minimum. In order to avoid the “worst of the best” solutions the Cooper algorithm is then restarted, so that many different random starting solutions are tested. And hopefully a good solution is found. The reason for not trying all possible combinations is a bit complicated, in terms of math though quite evident, more on that in section 12.1.3: “Stirlings Number of the Second Kind”.

### 12.1.1 The Weiszfeld Procedure

This section is partly adapted from “Facility Locations: Models & Methods”, chapter 2, by Love, Morris and Wesolowsky [FLMM].

The idea behind the Weiszfeld procedure is very simple, since the objective function for a single facility location problem with Euclidean distance is convex [FLMM p. 14], it only has one local minimum, hence a global one. The Weiszfeld procedure finds the optimal solution for the single facility location problem with Euclidean distance.

The objective is to minimize the total/average weighted Euclidean distance:

$$W(\boldsymbol{f}) = \sum_j w_j \cdot d_2(\boldsymbol{f}, \boldsymbol{a}_j)$$

where $d_2(\boldsymbol{f},\boldsymbol{a}_j)$ is the Euclidean distance between the position of the ambulance, $\boldsymbol{f}$, and the position of the $j$’th demand point $\boldsymbol{a}_j$. By solving $W'(\boldsymbol{f}) = 0$ it is possible to find the optimal solution, as known from basic calculus. Figure 24 shows an example.

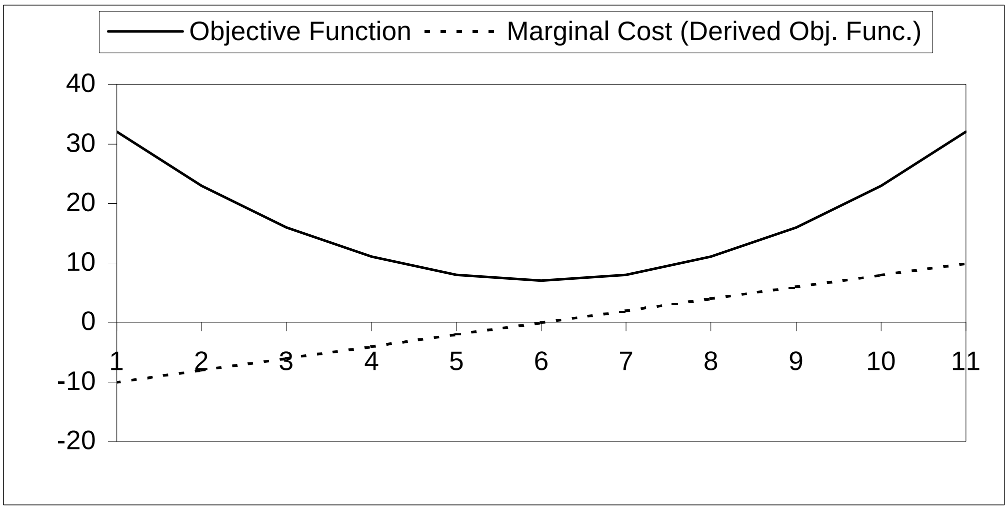



**Figure 24. A second-degree polynomial function and the derived function.**

However there are two problems when finding the local minimum: First of all it is not possible to isolate the coordinates of the ambulance, $\boldsymbol{f}$, on the one side of the equation, due to calculation of the Euclidean distance, $d_2(\boldsymbol{f},\boldsymbol{a_j})$, instead the following iterative procedure, known as the Weiszfeld procedure, is applied:

$$\boldsymbol{f}^{(k+1)} = \frac{\sum_j \frac{w_j \cdot \boldsymbol{a_j}}{d_2(\boldsymbol{f}^k, \boldsymbol{a_j})}}{\sum_j \frac{w_j}{d_2(\boldsymbol{f}^k, \boldsymbol{a_j})}}$$

The second problem is that the Euclidean distance, $d_2(\boldsymbol{f},\boldsymbol{a_j})$, can be zero if the position of an ambulance in some iteration coincides with one of the demand points the procedure will fail at some point (division by zero is not possible). The problem is illustrated in Figure 25.

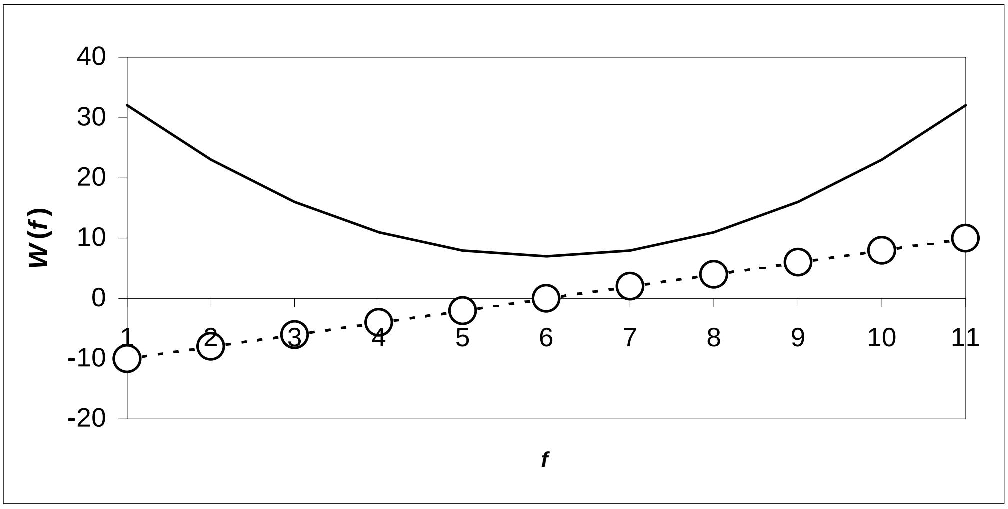



**Figure 25. The W(*f*) as a continuous function. O marks the values for which W(*f*) has no derivative.**

There are two ways of avoiding this, either by checking that the facility does not coincide with a facility or by adding a small value, ε, to the actual Euclidean distance. The problem then is that the procedure does not get to be as exact as the correct one, but with a small enough ε, the solution will come close. However the solution never coincides with the location of a demand point unless: (|| <some vector> || defines the length of <some vector>)

$$w_r \geq \left\| \sum_{j, j \neq r} \frac{w_j (\boldsymbol{a_r} - \boldsymbol{a_j})}{d_2(\boldsymbol{a_r}, \boldsymbol{a_j})} \right\|$$

$d(\boldsymbol{a_r},\boldsymbol{a_j})$ is the distance between the position of the $r$'th and $j$'th demand point. The easiest way to explain why that is so, is to see the right side of the equation as the resulting weight (or force) from the other points pulling on the demand point $r$. If the weight of the demand point $r$

is greater than the resulting weight from all of the other demand points, the current demand point *r* is the optimal solution for the single facility location.

The only major problem with the method is that it may have problems with convergence when the location of the ambulance coincides with the location of a demand point, though it seldom materializes [FLMM p. 15]. A lower bound is used as stopping criterion.

### 12.1.2 Lower Bound

The idea behind a lower bound is simple. If a value, which is known to be lower than the optimal solution value, can be found, the maximum possible gain between a known solution value and the optimal solution value can be assessed. If a solution value equals the lower bound found, it is the optimal solution. Figure 26 shows an example of the relationship between a lower bound and an objective function.

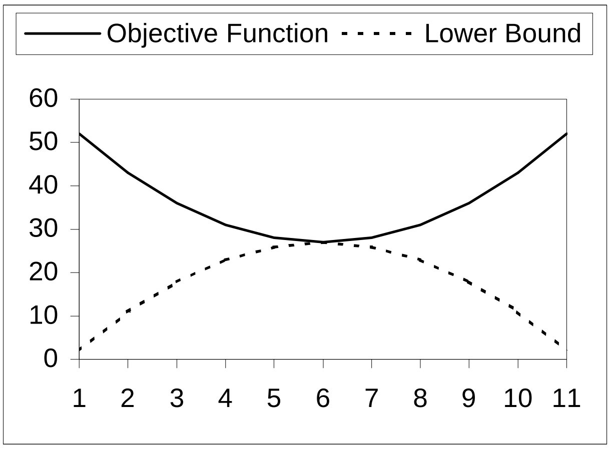



**Figure 26. Relationship between an objective function (minimization) and a lower bound.**

It is very important that lower bounds are tight, meaning that the gab between the objective function and the lower bound is as small as possible. The advantage of lower bounds is that they can be used to terminate an algorithm when a solution comes sufficiently close to the lower bound, hence the maximum loss can be determined.

There exist quite a few lower bounds for the Single Facility Location Problems (with Euclidean distance). The best one known to this project is one found by Drezner in 1984 [DZ].

$$W(\boldsymbol{f}^*) \geq \min\left\{\sum_j \frac{w_j \cdot \left|f_1^{(k)} - a_{j,1}\right|}{d_2(\boldsymbol{f}^{(k)}, \boldsymbol{a_j})} \cdot \left|f_1 - a_{j,1}\right| : f_1 \in R\right\}$$
$$+ \min\left\{\sum_j \frac{w_j \cdot \left|f_2^{(k)} - a_{j,2}\right|}{d_2(\boldsymbol{f}^{(k)}, \boldsymbol{a_j})} \cdot \left|f_2 - a_{j,2}\right| : f_2 \in R\right\}$$

$W(\boldsymbol{f}^*)$ is the optimal solution, $f_1^{(k)}$ and $f_2^{(k)}$ are the coordinates for the ambulance of the current iteration. The lower bound is the sum of the optimal solution to two one-dimensional single facility location problems, one for each dimension, which is easier to see if the equation is rewritten:

$$y_{j,1} \equiv \frac{w_j \cdot \left| f_1^{(k)} - a_{j,1} \right|}{d_2(\boldsymbol{f}^{(k)}, \boldsymbol{a_j})} \wedge y_{j,2} \equiv \frac{w_j \cdot \left| f_2^{(k)} - a_{j,2} \right|}{d_2(\boldsymbol{f}^{(k)}, \boldsymbol{a_j})} \Rightarrow$$

$$W(\boldsymbol{f}^*) \geq \min\left\{ \sum_j y_{j,1} \cdot \left| f_1 - a_{j,1} \right| : f_1 \in R \right\}$$

$$+ \min\left\{ \sum_j y_{j,2} \cdot \left| f_2 - a_{j,2} \right| : f_2 \in R \right\}$$

$Y_{j,1}$ and $Y_{j,2}$ can be defined as a new weights, one for each dimension. The problem is very easy to solve to optimality. The solution method is to sort the coordinates of the demand points and sum up the new weight from one end until half of the total weight is reached (or just above half), the demand point coordinate reached is the optimal solution.

### 12.1.3 Stirlings Number of the Second Kind

One way to solve the MFLA to optimality is to try all possible combinations of allocating the demand points to the ambulances. This will work because Weiszfeld provides optimal solutions to the single facility location problem (Euclidian distance). This corresponds to the number of ways of partitioning a set of *n* elements (demand points) into *p* nonempty sets (i.e. *p* blocks/ambulances), also called a Stirling number of the second kind. This can be found using the following formula [MATH]:

$$S(n, p) = \frac{1}{p!} \cdot \sum_{i=0}^{p-1} (-1)^i \cdot \binom{p}{i} \cdot (p - i)^n$$

It seems reasonable to assume from the formula that the Stirling number of the second kind is exponential in the number of demand points. However, it is somewhat more difficult to see the effect on the number of ambulances. To confirm the assumption and to investigate the effect of varying the number of ambulances, various combinations of the Stirling number of second kind have been plotted in Figure 27, logarithmic scale ($10^x$).

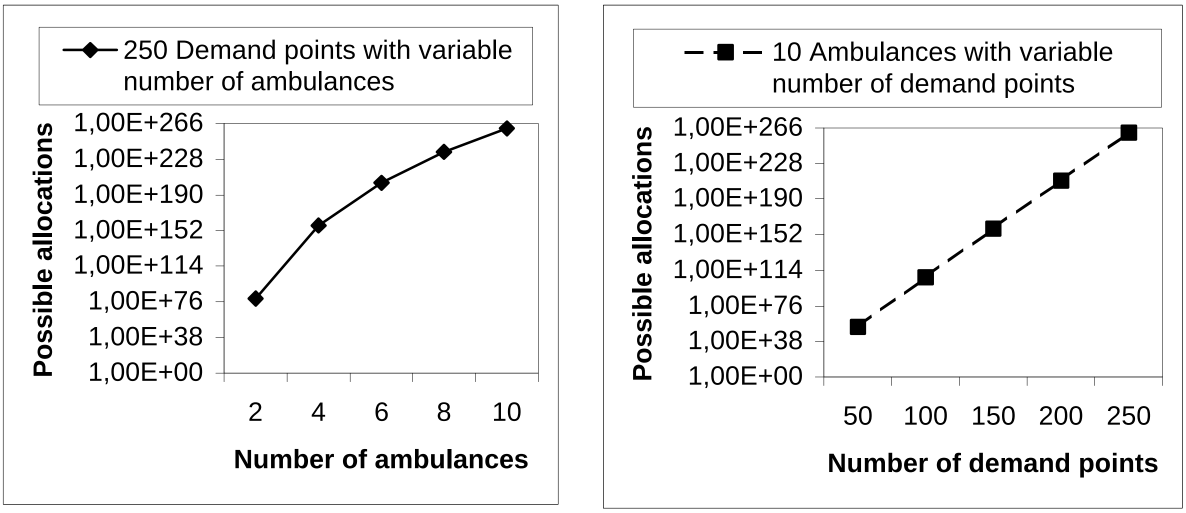



**Figure 27. Logarithmic scale. Line represents S(250,2), S(250,4), S(250,6), S(250,8) and S(250,10). Dotted line represents S(50,10), S(100,10), S(150,10), S(200,10) and S(250,10) .**

*(Assumptions refer to situations where p is significantly smaller than n)*
The exponential growth of *n* seems to be confirmed, and though the effect of increasing the ambulances is not as extreme as increasing the number of demand points it is still severe. This is in no way proof of the properties of S(*n*,*p*), only an indication. Actually it is a problem in itself to calculate the number with ten ambulances, MS Excel failed at 291 demand points and a ti-92 calculator failed at 325 demand points. This is due to the $(k - i)^n$ factor, $10^{325}$ is a rather large number. It should be clear from Figure 27 that trying all possible allocations in this project is not an option. It might seem impossible to even try solving the problem using the Cooper algorithm, nevertheless there are many solutions which are obvious not to try. The Cooper algorithm, due to its descending nature, does not find many of the poor solutions instead it finds good solutions.

It is clear that even this very primitive model for the relatively large problem at hand is practically impossible to solve to optimality. There do exist exact methods, quite a few are mentioned in "Facility Location: Applications and Theory ", by Drezner and Horst (editors), chapter 1 page 17. Nevertheless MRC is fairly easy to implement and is quite effective compared with the effort, since there are no parameters to be tested. MRC has proven to be quite capable of finding good solutions.

### 12.1.4 Implementation of the MRC

I have chosen to set ε, the parameter introduced to avoid zero Euclidean distance, to 0.01 meter, based on a small experiment of setting it at as small a value as possible, without having the algorithm fail due to the numeric problems, and since 0.01 meter is better than the precision in the data, and since the Euclidean distance is seldom less than a few hundred meters in this project, it does not seem to constitute a problem.

Checking if the solution (optimal) coincides with a demand point due to the resulting weight (force) is left out, since the ε-value eliminates the risk of failure when the position of an ambulance and a demand point coincides. Nonetheless this should be implemented in a commercial distribution, to avoid any possible problems.

The random solution for the MRC is found by picking $p$ different demand points. Half a meter is added to each of the coordinates to avoid coinciding with the demand point, though the ε-value probably would take care of the problem.

The lower bound has been implemented, with a comparison every ten iterations. The difference between the objective function value and the lower bound for a given solution must be less than ten “weighted meters”, in order to terminate the algorithm. The reason for not calculating the lower bound at every iteration is that in terms of computing time it costs about the same to calculate the lower bound as it costs to do another iteration of Weiszfeld, in this way solutions when more than five iterations are required will be found more quickly than when not comparing at every iteration.

### 12.1.5 Running Time

It is not possible to say anything about the running time of the algorithm, given that the termination of the Weiszfeld procedure is based on a lower bound, and the MRC terminates when no improvement is found. There might even be problems with convergence, meaning that it might never terminate (in theory). Nevertheless it is possible to say something about a single iteration of the Weiszfeld procedure, which is linear with respect to the number of demand points (“12.1.1: The Weiszfeld Procedure”), in terms of Big O notation: $O(|J|)$.

## 12.2 Solving p-median and p-center Problems

There exist quite a few different heuristic solution methods for the two problems.

The perhaps most famous heuristic for the p-median problem is the T&B-heuristic developed by Teitz and Bart in 1968 [TB], [FL p. 179]. It is based on a very simple idea of changing the positions of the ambulances while continuously improving the solution; it is a so-called descent heuristic. A descent heuristic is an algorithm that is only concerned with improving the current solution, and when given the same starting solution it always return the same solution, the disadvantage is that the solutions tend to reach local minimas which are "easy" to find, but not offering the ability to investigate those that may be more difficult to find. T&B has proven to be both fast and efficient for minor p-median problems. Other methods have been used such as mathematical programming and Lagrangain relaxation by Narula et. el. (1977). The metaheuristics Simulated Annealing (Murray and Church, 1996), Tabu Search (Rolland et. al.,1996) and Genetic Algorithm (Bozkaya et al. 2001). [FL p. 179].

The most common way of solving p-center problems is using a binary search over a range of coverage distances (must be integer) (Handler and Michandani, 1979; Handler, 1990). The main idea is to cover all accidents (customers) using ambulances (facilities) with a fixed distance covering distance and a variable number of ambulances (facilities). When a solution has *p* ambulances an "optimal" solution has been found [FL p. 89].

As seen in chapter 11: "Models" there is not that much difference, in terms of the mathematical representation, between the p-median and the p-center problems. In order to take advantage of this and to allow a high level flexibility and ensure fast solution times, I have chosen a metaheuristic know as the Noising Method (NM). The noising method is basically a Simulated Annealing (SA), or a generalization of SA as the "inventors" of NM would say [NMG p. 91]. I will not describe the similarities between SA and NM but just present the NM, and to those who know SA the similarities will be clear. The reason for using the NM rather than SA is only a matter of personal preference.

A second solution method will be applied for problems, both p-center and p-median, with one and two ambulances and problem with few ambulance points (Falck garages). It is a solution method that checks all possible solutions, more on that in 12.4: " Checking All Possible Solutions".

The minor difference between the p-center and p-median models might not be that insignificant at all. It is a bit interesting that nowhere during this research has it been possible to find anyone who has tried applying the "same" solution method to the two problems. This could in many cases have been done quite easily by just exchanging the way the solution value is calculated. This simple change will be applied in this project, which is also the reason for including the p-center problem. The p-center problem does have serious drawbacks as described in chapter 11: "Models", but it can be "solved" by the NM with very little work, compared with otherwise having to implement another solution method.

## 12.3 The Noising Method

The main idea is to search the solution space, the set of feasible solutions, in such a way that local minimas can both be found and escaped again. The search is an investigation of the neighborhood (or part of it) for a current solution. The best encountered solution is then noised by adding some value "*noise*" to its solution value. If the neighbors noised solution value is better than the one of the current solution the neighboring solution becomes the new current solution. Which is also the reason for calling the heuristic the Noising Method (NM). The noise is gradually reduced as more and more iterations are carried out, so that the solutions, which are better than the current solution, are more likely to be accepted as a new current solution. Defining the neighborhood and how to search it is the most critical part of the noising method, as with many other metaheuristics.

A solution is defined as $p$ points representing the position of the ambulances, *ambulance points*, hence there cannot be more than one ambulance at an ambulance point. The heuristic deals with three types of solutions: a current solution, $s$, a temporary solution, $s´$ and a best-known solution, *best_solution*. The current solution is the solution, which the heuristic is trying to improve and from which neighbors are picked and examined. The temporary solution is a neighbor to the current solution, which may or many not be accepted as the next current solution. The best-known solution is just a memory of the best solution for the current run of the heuristic. The heuristic is partly adopted from "The Noising Methods: A Generalization of Some Metaheuristics" by Charon and Hudry [NMG], however most of the techniques used are common in the field of Operations Research. The version of NM used in this project for both the p-median and p-center problems is presented in pseudo code in Figure 28. The running times for $O(p \cdot r)$+ and $O(nb)$ (running time for exploration of neighborhood) will be explained in the sections 12.3.4.2: "The Random Solution" and 12.3.4.3: "The Neighborhood Exploration and Solution Evaluation". $O(p)/O(1)$ should be read as $O(p)$ or $O(1)$, for two reasons: Fist of all an "if-statement" might be false and no action is taken, and secondly a solution can either duplicated or references depending on the implementation.

| | **$O(\cdot)$** |
|---|---|
| Draw the initial current solution *s* randomly | $O(p \cdot r)$+ |
| Set *best_solution* ← *s* | $O(1)$ |
| *trial_meter* ← 0 | $O(1)$ |
| While *trial_meter* < *total_number_of_trials* do | |
|     let *s´* be the chosen neighbor | $O(nb)$ |
|     let *noise* be a random number uniformly drawn from [-*rate* , *rate*] | $O(r)$ |
|     if ( *f*(*s´*) + *noise* < *f*(*s*) ) then *s* ← *s´* (new current solution) | $O(p)/O(1)$ |
|     if ( *f*(*s´*) < *f*(*best_solution)* ) then *best_solution* ← *s´* (new best solution) | $O(p)/O(1)$ |
|     if ( *trial_meter* = 0 modulo *nb_trials_at_fixed_rate* ) then | $O(1)$ |
|         $rate \leftarrow rate - rate_{decrease}$ | |
| End of while | |
| Unnoised descent on *best_solution* | |
| *Running Time (per iteration):* | $O(nb)$ |

**Figure 28. Pseudo code of the Noising Method used in this project.**

The "unnoised descent" at the end of the heuristic will be explained in section: 12.3.3 "Other Issues", and is included to ensure a local minimum is found. The running time for each iteration is expected to be dependent on the running time of the exploration of the neighborhood $O(nb)$, though if no or only very little exploration is carried it the other elements will have a greater effect on the overall running time.

There are many choices to be made when using a metaheuristic like the NM, choices about noise parameters, neighborhood etc. In this project the following will be discussed:

- Acceptance of best solution
  - $rate_{min}$
  - $rate_{max}$
  - *nb_trials_at_fixed_rate*
- Neighborhood
  - Definition of neighborhood
  - How to explore the neighborhood
- Other
  - Stopping criteria - Number of iterations
  - Restart
  - Descending (define)

Sometimes it is also necessary to decide upon an initial solution and how to obtain it. A pure random solution is used in this project for two reasons. The quality of the starting solutions is seldom an important factor for metaheuristics like NM [NMG p. 92] and it is not difficult to obtain a feasible solution (a list with $p$ different ambulance points).

### 12.3.1 Acceptance of Best Solution

The noising method is different from a descent heuristic in the way that it adds randomly generated noise to the solution value, so that sometimes the solution that are actually better than the current solution does not get accepted and solutions that may be worse than the current solution may be accepted. The reason for discarding solutions better than the current solution is to avoid descending too fast, and to be able to escape local minimas. The noise consists of three parameters, a start/maximum noise level, $rate_{max}$, an end/minimum level, $rate_{min}$, and a distribution from which the noise is randomly picked. The distribution could be any distribution normal, uniform etc. [NMG p. 91]. $rate_{max}$ is the noise level at the first iteration and $rate_{min}$ is the noise level at the last iteration. The noise level in an iteration is stored in the variable *rate* which is decreased at the end of an interval of *nb_trials_at_fixed_rate* iterations. *rate* is decreased by the value, $rate_{decrease}$ which is calculated as follows:

$$\frac{rate_{max} - rate_{min}}{((\,total\_number\_of\_trials \,/\, nb\_trials\_at\_fixed\_rate\,) - 1\,)} = rate_{decrease}$$

The decrease of the noise in this project is an “approximation” to linear. The consequence of using different values of *nb_trials_at_fixed_rate* can be seen in Figure 29 and Figure 30. $rate_{max}$ is set to 100 and $rate_{min}$ is set to 10.

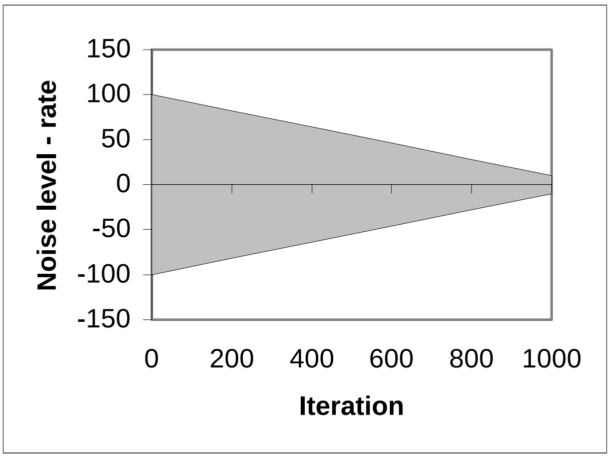



Figure 29. Noise level if ***nb_trials_at_fixed_rate*** = 1 (linear).

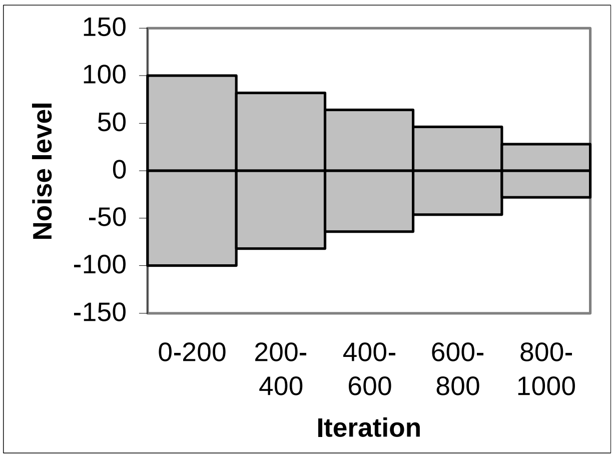



Figure 30. Noise level if ***nb_trials_at_fixed_rate*** = 200.

The noise rate could be decreased in many other ways, using logarithmic functions is one option, or another possibility could be not to decrease it at all. The effect on the performance of the heuristic when using different choices of noise reduction has not been determined.

### 12.3.2 Neighborhood

There are two things to be considered about neighborhoods. First of all it is necessary to define a neighborhood, secondly it must be decided how the neighborhood is to be explored.

One way of defining a neighboring solution could be to define the neighbors of each ambulance point, and then move an ambulance to one of the neighbors from its current location (ambulance point). A neighboring solution would be a solution where one of the ambulances moved to one of its neighbors.

That again would require that all demand points be reallocated to the nearest ambulance, hence a neighboring solution is found. A neighbor to an ambulance point could be any other ambulance point within a certain Euclidean distance, travel time, etc.

Obviously a solution has more than one neighbor, hence a choice must be made of which neighbor to pick. There are many options when considering how to explore a neighborhood. A randomly chosen neighbor could be picked (no exploration), the best neighbor could be picked (exploring the entire neighborhood) etc. In this project the neighborhood will be partially explored. The ambulance to be moved is picked at a random basis and a randomly chosen percentage (averaging 3 %) of the ambulances in the neighborhood is explored. The best neighboring solution encountered will be picked. The reason for choosing the partial exploration is based on the recommendation by Charon and Hudry [NMG p. 90] in order to avoid descending too fast. That it is an average exploration of 3 % of the neighborhood and not something else is an intuitive guess on a good value. A good value is expected to be one that gives some descend but not too much, there is no reason why any other value could not be used. The higher the percentage, the longer it takes to do an iteration.

### 12.3.3 Other Issues

The stopping criterion, i.e. what makes the algorithm terminate, could be many things. The easiest to implement would be a fixed number of iterations, as done in this project. The main advantage is, besides the implementation and easy control of the noise parameters, that it approximately uses the same amount of computational time for each run (depending on the neighborhood structure). The disadvantage is that the termination has nothing to do with the quality of the solution. If too few iterations are used there could still be much to gain by doing a few more. The disadvantage of doing too many iterations is that they take time and with very little improvement achieved, if any. Other choices could have been:

1. When the algorithm has not improved the solution for some X iterations.
2. When the improvements over Y iterations are smaller than some number $\alpha$.
3. Using a lower bound.
4. When the algorithm has run for a certain amount of time.

The main disadvantage of choosing alternatives 1, 2, 3 or 4 is that it makes the control of the noise parameters more difficult. Other reasons for not choosing alternatives 1 and 2 are that the amount of time used by the algorithm cannot be controlled, however it might be easier to say something about quality of the solution value and less time is spent on insignificant improvements. Alternative 3 would be the essential way to verify the quality of a solution, though this would require both a sharp lower bound and a lower bound relatively easy to calculate for large problems. There has not been a thorough investigation into this, though no indications in the literature used in this project showed any existence of very effective lower bounds, nonetheless lower bounds do exist [NCJB]. Alternative 4's only disadvantage is the more difficult implementation. The reduction of noise could be done using a *time_meter and time_at_fixed_rate* instead of *trial_meter* and *nb_trials_at_fixed_rate*. It should definitely be chosen in systems where a percent of the running time could be crucial.

Charon and Hudry [NMG p. 93] suggest that there might be an advantage in restarting the algorithm after a fixed number of iterations. By restarting is meant setting the *best_solution* to the current solution, *s*. Nonetheless such a method can always be implemented if the solution method does not perform as could be expected. Another alternative suggested by Charon and Hudry is to perform "unnoised" descents from the current solution at a given interval of iterations, like the one after the termination of the while loop. The descent ensures that a local minimum is found. Figure 31 shows how the descent is performed, *i* is an ambulance point from the set of ambulance points *I*, *P* is the set of ambulance points in the best solution.

```
boolean bl_has_improved ← true
While bl_has_improved do
        bl_has_improved ← false
        For Each i in P do
                i´ ← the best neighbor best neighbor of i
                if (i replaced with i´ is a better solution) then
                        i is replaced by i´ in P and bl_has_improved ← true
        End For Each
End of While
```

**Figure 31. Pseudo code for the descent procedure.**

The descent is performed by iterating through the ambulance points in the *best_solution*, *P*, exchanging each with its best neighbor. The iteration is repeated if an improvement is found during the last iteration. The descent is probably an implementation of the T&B algorithm (Teitz and Bart [TB]) [FL p. 181], though the original description has not been investigated. As for the "restart" it could always be implemented inside the while loop in Figure 28 at a later stage, if necessary. In theory the algorithm could continue to run as many times as there are possible solutions to the problem, even though it is unlikely ever to materialize.

### 12.3.4 Implementation and Running Time of NM

This section explains how the NM is implemented, divided into the following three smaller sections

- Calculating the cost between two points.
- The generation of a random solution.
- The neighborhood exploration and solution evaluation.

#### 12.3.4.1 The Cost $d_{ij}$

There are basically two ways to handle the network costs: Costs can all be calculated in advance and be stored in a matrix or they can be calculated on the fly (when needed by the NM). The first alternative seems to suit this project best, since the same distance will be needed several times due to the many runs needed for testing the heuristic. Unfortunately the number of distances needed is the product between the number of demand points, $|J|$, and ambulance points, $|I|$. For major problems with thousands of demand points and ambulance points, the number of distances reaches millions. The GIS software used for the calculation of the shortest path only knows the Dijkstra shortest path algorithm, which is fine when the shortest path from one point to another is needed, but when calculating the all-to-all shortest path it could become somewhat costly, since many of the calculations could have been used again instead of being recalculated. Table 5 shows the theoretic running times for various shortest path algorithms. $N_V$ is the number of vertices, nodes, points etc. in the graph $N_M$ is the number of edges, lines, shapes etc.

| Algorithm | One to One | One to All | All to All |
|---|---|---|---|
| Dijkstra (1) | $O\left(N_V^2\right)$ | $O\left(N_V^2\right)$ | $O\left(N_V^3\right)$ |
| Dijkstra (2) | $O(N_M \cdot \log_2(N_V))$ | $O(N_M \cdot \log_2(N_V))$ | $O(N_V \cdot N_M \cdot \log_2(N_V))$ |
| Ford-Bellman | $O(N_M \cdot N_V)$ | $O(N_M \cdot N_V)$ | $O\left(N_M \cdot N_V^2\right)$ |
| Floyd-Warshall | - | - | $O\left(N_V^3\right)$ |

**Table 5. Overview of some shortest path algorithms. The two versions of Dijkstra are two different ways of implementing Dijkstra, number two is not always mentioned in literature [SL].**

Theoretically the most efficient algorithm is Dijkstra, which version depends on the number of vertices and edges in the graph. Hence in theory there does not seem to be anything to gain from applying a different algorithm, tough practical experiments may provide different results.

Another problem is that points, between which the distances are needed, are not really part of the network, they might be on the middle of an edge. How ArcView handles that and how it will affect the running times in Table 5 is not known. Quite a bit could probably be gained by cleaning up the network so that it only contains the demand and ambulance points and the distances between them, though such a transformation in itself might be quite costly.

The implementation used, which is running Dijkstra (One to One) $|I|\cdot|J|$ times, gives the running time:

$$O\left(N_V^2 \cdot |I| \cdot |J|\right)$$

The cost of calculating 1053 demand points and 1053 ambulance points (the same as the demand points), was 297485 seconds on a high end desktop computer, which is roughly 82 hours or three and half days. In total there were 1,107,756 distances (the distance from a point to itself was set to zero). The problem of calculating the distance matrix has proven to be by far the largest challenge when applying the location allocation methods in this project. The demand points used had a distance between each other of 2140 meters (hexagonal tessellation), if that is reduced to about one kilometer it would probably multiply the calculation time with about 16, which is nearly two months, making it impossible to use on real life problems. The road network changes once in a while, which means that new calculations have to be carried out regularly. Nonetheless Falck and other companies may not be able to operate with an inaccurate road network for two months.

#### 12.3.4.2 The Random Solution

The random solution is generated by the means of picking $p$ different numbers from the interval $[0, |I|-1]$, as shown in Figure 32. Each number uniquely identifies an ambulance point.

```
Create empty list: random_solution
While random_solution.Size() < p do
        Pick a random number between 0 and |I| - 1
        If number is not in random_solution then add number to random_solution
End of While
```

**Figure 32. Generation of random solution.**

In theory the algorithm could run “forever” if the random generator constantly picks numbers/ambulance points, which have already been picked, though this is highly theoretical. In general this procedure is only problematic if $p$ is close to $|I|$. Though if $p > ½|I|$ it would be faster to pick those ambulance points which were not supposed to be in the solution. Nevertheless it is not likely to apply for $p$-median and $p$-center problems since the $p$ is expected to be a great deal smaller than $|I|$. The running time is set to $O(p \cdot r)+$, where + indicates that it may be slightly above $O(p \cdot r)$ if the a number is picked more than once.

### 12.3.4.3 The Neighborhood Exploration and Solution Evaluation

The exploration of the neighborhood is carried out using the following procedure: Pick a random number between one and *p*, alternatively between zero and *p – 1*, corresponding to which ambulance in the solution should move. Subsequently iterate through all ambulance points with a three percent chance of exchanging it with the "old" ambulance point and calculating the new solution value. The best solution of the examined solution is returned. The exploration of the neighborhood requires a fourth solution type to be introduced: *s"* the working solution.

| | **$O(\cdot)$** |
|---|---|
| Pick a random ambulance point *ap_old* from *s* | $O(r)$ |
| *s"* ← nil | $O(1)$ |
| *s'* ← nil | $O(1)$ |
| *f(s')* ← ∞ | $O(1)$ |
| For Each *i* in *I* do | |
|     *do_it* ← random number between 0 and 99 | $O(r)$ |
|     if ( *do_it* < *3* and *i* <> *ap_old* and i not in *s* ) then | $O(3)$ |
|         *s"* ← s where *ap_old* is replaced with *i* | $O(p)/O(1)$ |
|         For Each *j* in *J* do | |
|             allocate *j* to the nearest ambulance point in *s"* | $O(\|I\|\cdot\|J\|)$ |
|         End of For Each | |
|         if ( f(*s"*) < f(*s´*) ) then *s'* ← *s"* | $O(1)$ |
|     end if | |
| End of For Each | |
| *Running Time:* | $O(\|I\|\cdot\|J\|)$ |

**Figure 33. Pseudo code for neighborhood explorartion.**

If no neighbor is accepted, e.g. small neighborhoods, the exploration is restarted until a neighbor is examined.

The calculation of a new solution value, f(*s"*) is the most time consuming procedure in the heuristic. The reason is that each demand point must be allocated to the nearest ambulance point. With a large number of ambulance points the running time of $O(|I|\cdot|J|)$ is quite time consuming. The allocation for the best solution (*best_solution*), current solution (*s*), temporary solution (*s´*) and a working solution, *s"*, for the exploration of the neighborhood is remembered (stored) making a recalculation fast and easy. The running time of $O(|I|\cdot|J|)$ is based on iterating through all demand points and for each demand point iterating through all ambulance points to find the nearest one, as seen in Figure 33.

However with the stored current solution a comparison between the cost of traveling from the "old" ambulance point to the demand point and traveling from the "new" ambulance point to the demand point can be used instead. Of course if the "old" cost is less it is necessary to find the nearest ambulance point in the working solution. Such an implementation was attempted, but failed on the part of keeping track of which had changed and which had not. How much can actually be saved depends on the number of demand points that do not change facility, hence for small numbers of $p$ there is only little to gain, whereas for larger numbers of $p$ fewer demand points will be reallocated.

## 12.4 Checking All Possible Solutions for the p-center and p-median Problems

Finding the optimal solution is possible for a small number of ambulances, one or two in this project, or when using very few ambulance points (Falck's garages), since all possible solutions can be checked. The algorithm used to check all solutions is very primitive. This is simply done by perceiving a solution as a number from a different number system. Hence using two ambulance points corresponds to the binary system, and using 10 corresponds to using the traditional 10 number system. If two ambulances is placed among three ambulance points it would check the solution in the order given in Table 6.

| ***Solution number*** | ***Ambulance One*** | ***Ambulance Two*** |
|---|---|---|
| ***1*** | 1 | 1 |
| ***2*** | 1 | 2 |
| ***3*** | 1 | 3 |
| ***4*** | 2 | 1 |
| ***5*** | 2 | 2 |
| ***6*** | 2 | 3 |
| ***7*** | 3 | 1 |
| ***8*** | 3 | 2 |
| ***9*** | 3 | 3 |

**Table 6. Solution generated by the "Try All" algorithm. Numbers refer to ambulance points.**

It is clear that this is both a primitive and rather unintelligent method to check all possible solutions since the same solution will be checked more than once, {1,2} is the same as {2,1}. Furthermore solutions in which the same ambulance point is used more than once cannot be optimal. For four ambulances and 1000 ambulance points 25 times more solution than necessary would be checked (six times for three ambulances and twice for two ambulances).

There probably exist algorithms, which can generate the solutions easily, without producing irrelevant and redundant solution, though there has been no investigating into it. The reason for sticking with the primitive method was that it was expected to take longer time to find an implement the more intelligent methods than it would take solving the problems with the primitive method. An assumptions which has not been quite true, since a few of the problems expected to be solved by the method, has been solved using the Noising Method, as explained in chapter 14: "Allocation of Ambulance on Funen".

A variant of the solution method was implemented for $p > n/2$. The reason being that it is faster to check which ambulance points do not have an ambulance allocated to it. Hence the number of solutions from Table 6 can be reduced to three: {1}, {2} and {3}.

# 13 Test of Solution Methods

*This chapter deals with estimating the best parameters for the solution methods to ensure a good performance as well as validate models and solution methods.*

As long as a solution method takes parameters, there is always the choice of which value to choose. The main objective of testing parameters is to achieve the best possible performance. Equivalent to defining the optimal allocation of ambulances, there is also a choice of defining what an optimal performance is. Good performance can be a number of things. The following elements could be a measure of performance [DCE p. 14]):

1. What is the quality of the best solution found?
2. How long does it take to determine the best solution?
3. How quickly does the algorithm find good solutions?
4. How robust is the method? Does it always produce good solutions? Is it reliable?
5. How “far” is the best solution from those more easily found?
6. What is the tradeoff between feasibility and solution quality?

A seventh relevant addition for the Nosing Method (NM) could be:

7. How well does it escape local minima? Since this is what distinguishes it from just descending.

In short three factors seem to describe a set of parameters: solution quality, computational effort and robustness [DCE p. 14]. In this project the quality of the solutions and the robustness are the most important for a good performance, however in a real life situation Falck will without doubt appreciate getting good solutions fast if they need to apply the methods for dynamic allocation of the ambulances or if they need to do a lot of studies. Still, they also need a good quality, which they can rely upon, hence robustness is also important.

Essentially this project deals with three different problems, each with its own solution method:

| Problem | Solution method |
|---|---|
| MFLA | Multi Restart Cooper (MRC) |
| p-median | Noising Method (NM) |
| p-center | Noising Method (NM) |

That the same solution method is used for the p-median and p-center problems does not necessarily mean that the solution method, NM, should use the same parameters when solving the two different problems.

There are not really any parameters worth testing for the Multi Restart Cooper other than the number of restarts, i.e. computational effort, compared with value of the best solution found. The ε-value, used to avoid dividing with zero distance, has already been determined, though it has not been documented. The Noising Method (NM) on the other hand offers quite a number of parameters. Table 7 contains a listing of parameters.

| Solution method | Parameters |
|---|---|
| Multi Restart Cooper | Number of restarts<br>ε (Avoiding zero Euclidean distance) |
| Noising Method | Maximum Noise, $rate_{max}$<br>Minimum Noise, $rate_{min}$<br>Distribution of noise<br>How to decrease noise<br>Type of neighborhood<br>Percentage of neighborhood to explore<br>Neighborhood exploration<br>Frequency of restarting<br>Frequency of descending<br>Number of iterations |

**Table 7. Parameters in solution methods.**

The parameters are often known to be problem related. Hence, if a solution method is required to solve many different problems, the parameters need to be tested on many different problems first to ensure good performance for all parameters. Since the rest of this project will only deal with one dataset and a variable number of ambulances, the dataset used for testing parameters will be the same as the one being analyzed. It is also probable to believe that other datasets, not only from Funen, but any dataset with a clustered structure (accidents congested in smaller areas, towns located at a certain distance) can be solved using the same parameters.

The only parameters tested will be the $rate_{max}$ and the size of the neighborhood. $\text{rate}_{\text{min}}$ is set to zero and the *nb_trials_at_fixed_rate* (rate of decreasing noise) is set to ten (every 10 iterations) using the scheme presented in section 12.3.1: "Acceptance of Best Solution", no restarting or descending (beside after termination of the main loop), the exploration of the neighborhood is set to 3%. In general it could be argued that the number of iterations should increase, when the number of ambulance, *p*, increases (as the problem becomes more difficult to solve). However for simplicity 2000 iterations are used in general, though for smaller numbers of *p* less is used, typically 1000. It will be explained along the way. Although a small test for the effect of using more iterations will be presented for the p-median problem.

The dataset used is the snapped hexagonal tessellation of the "Spring at Work"-accidents with a distance of 2140 meters between the demand points before snapping as mentioned in section 10.7: "Conclusion on Accident Analysis". To each point (1053 in total) is added the weight of one, to ensure that all points have a weight, so that they are included in the solution value for all models. The previously zero weighted demand points could of course have been removed from the demand point set. The reason for not just removing the zero weighted demand points is that it would jeopardize the value of solving the p-center problem, since the maximum response time would then only be valid for those demand points that had an accident allocated to them, Falck is expected to cover the entire Funen. The dataset also makes a good combination of accidents in the past and the geographic extent of Funen.

## 13.1 Testing the MRC

As mentioned in section 12.1.4: "Implementation", it is difficult to say anything about the running time of the algorithm. In real time it solves the problem within a half to a third of a second, not including loading the data which takes much longer time. There seems to be a slight increase in the solution time when the number of ambulances increases, though it has not been documented. The testing of the MRC was carried out doing 250 runs for each value of *p* between two and twelve, and one run for one ambulance, since the Weiszfeld produces an optimal solution for one ambulance. The result in terms of lowest (Min), average (Mean) and highest (Max) solution values are presented in Figure 34.

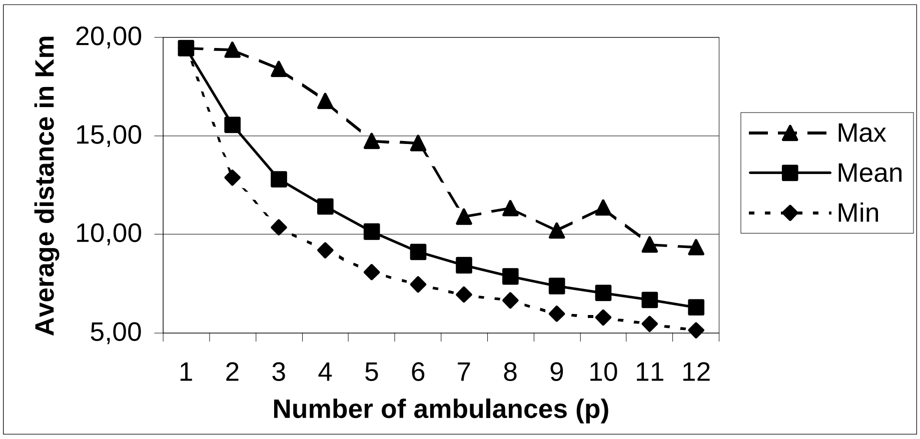



**Figure 34. Best (Min), Worst (Max) and average solution values for 250 runs of MRC for each value of p.**

The difference between the average and best solutions does not seem to be that big, only between 18% and 26% compared with the best solution. The worst solution on the other hand is more than 96% worse than the best solution (for six ambulances), and the "best" of the worst solutions is a whole 50% off. Which is much worse than the 41% (here 96%) what Eilon, Watson-Gandy and Christofides found in a practical experiment [FL. p. 17], section 11.1: "Continuous Model". In order to investigate this further the quartiles have been plotted in Figure 35.

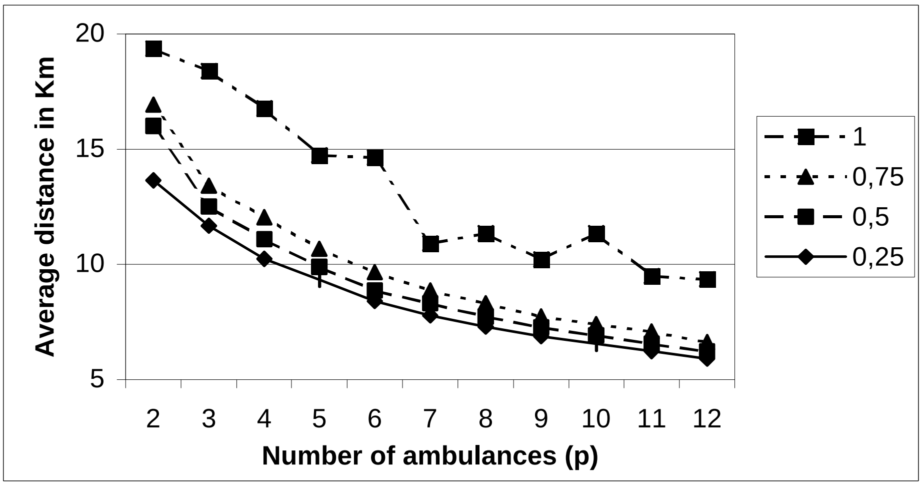



**Figure 35. Quartiles for 250 runs of MRC for each value of p.**

From Figure 35 it is cleat that it is only the 25% worst values, which are relatively off the mark, or in other words it is only a few runs which are totally unacceptable. The best 75% are within 33% of the best run. Nonetheless 33%, even 20% or 10% is quite a noteworthy risk to run.

### 13.1.1 Model Validation - MFLA

The model for the MFLA was, as mentioned in section 11.1: "Continuous Model", a very primitive model for the real life problem of allocating ambulances. One of the problems with the MFLA is that it places ambulances everywhere, even in lakes and at sea as shown in Figure 36.

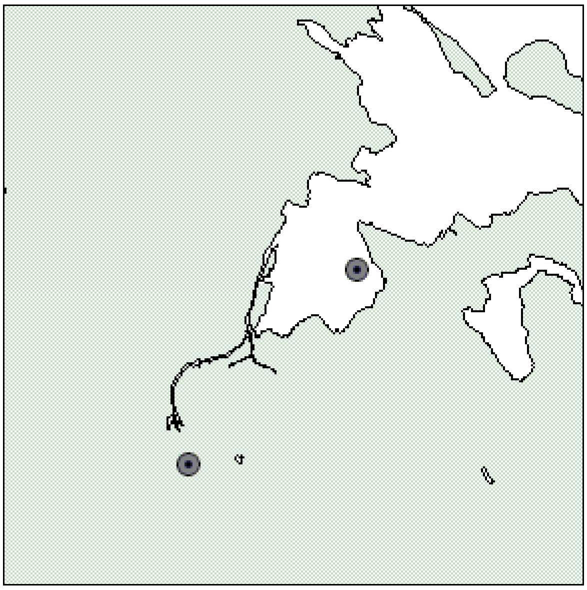


**Figure 36. One ambulance in Odense and one in the water, from the best solution of eight ambulances. Θ marks the position of the ambulances.**

The best results of placing one to twelve ambulances, using the MFLA model, are shown in Appendix IV.

Another problem with the model is that it needs a reference to travel time. The most obvious solution is to snap each location to the road network, and use the snapped location to calculate the travel time. However, it takes about 25 minutes to calculate the travel time to all demand points, consequently the travel times were only calculated for the best solution. A discussion of results with travel time can be found in chapter 14: "Allocation of Ambulance on Funen", together with a comparison with discrete results.

### 13.1.2 Conclusion on the MFLA Model and MRC Solution Method

The MRC is a fast but not very reliable algorithm, which combined with the possible location errors of the MFLA makes it risky to use the MFLA and MRC for placing ambulances.

## 13.2 Testing the Noising Method – Early Results and Assumptions

When implementing the Noising Method a minor test problem was created in order to verify that data was loaded correctly and to check that the algorithm actually worked. The chosen test problem comprised the southeastern area of Funen, including the island of Tåsinge, with about 200 demand points, which were also used as ambulance points. The problem tested was allocating three ambulances for the p-median problem. Once it got working it performed quite oddly, the idea is that the current solution should display something like (Figure 37):

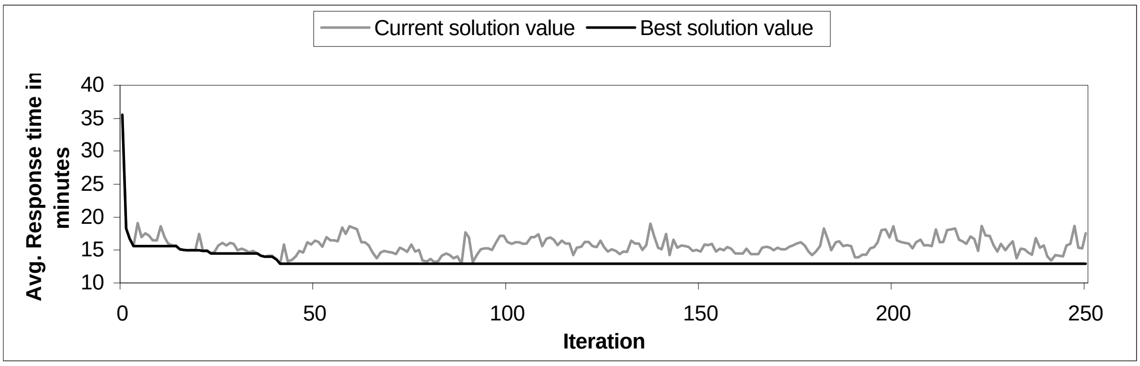



**Figure 37. How a run should perform.**

However, most runs performed as shown in Figure 38:

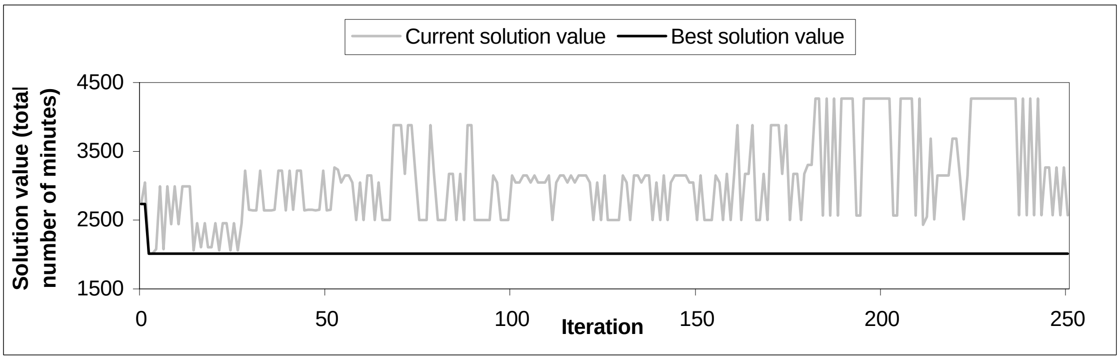



**Figure 38. “Strange” run performed on a test dataset.**

The best solutions were often found within five to ten iterations when one or two of the ambulances “jumped“ on to Tåsinge, from where they could not escape. The best results were found with only one ambulance on Tåsinge. Another odd thing was that even when running a thousand iterations it completed in around 10 seconds.

Despite the “poor” results, a test was performed on the p-median problem (on the same dataset as for the MRC) over four different values of $rate_{max}$, five different neighborhood sizes and two different numbers of ambulances. The chosen parameter values are presented in Table 8.

| *p* | *Iterations* | *Runs* | *Noise values - $rate_{max}$* | *Neighborhood sizes in km* |
|---|---|---|---|---|
| 3 | 1000 | 25 | {18.86, 37.72, 75.44, 150.88} | (6.25, 12.5, 25, 50, 100} |
| 7 | 2000 | 25 | {18.86, 37.72, 75.5, 150.91} | (6.25, 12.5, 25, 50, 100} |

**Table 8. Parameters used for testing neighborhood size *and $rate_{max}$*.**

The choice of $rate_{max}$ was purely random since the heuristic was quite unstable and difficult to interpret. The odd $rate_{max}$ values are due to implementation (programming) of the solution value, which is calculated as the total weighted response time and not the average response time. The parameters for the neighborhood size were chosen to ensure that neighborhoods of all sizes were tested. The results are presented in Figure 39, Figure 40, Figure 41 and Figure 42. The number of iterations for three ambulances is set to less than the number of iterations for seven ambulances, since it should be easier to solve.

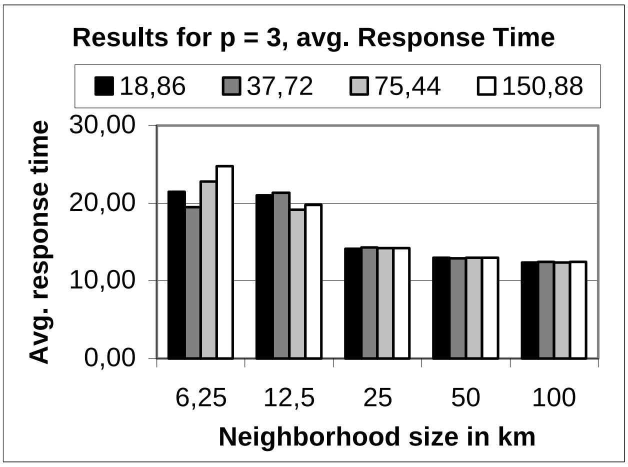



**Figure 39. Average response time is in minutes.**

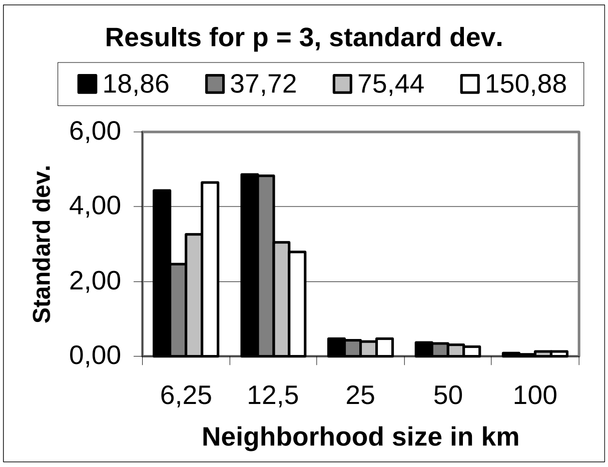



**Figure 40. Deviations are in minutes.**

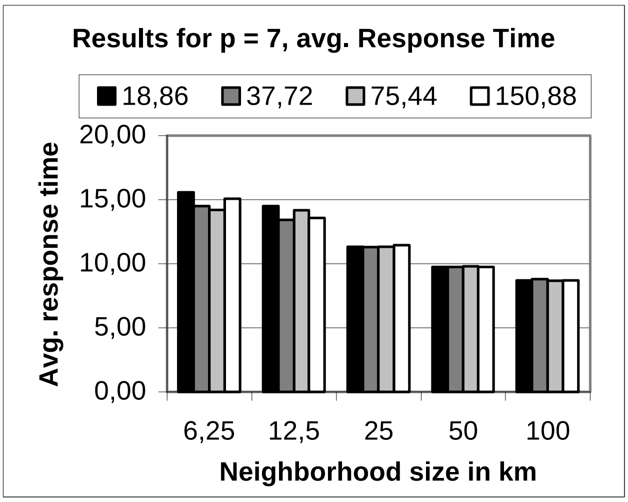



**Figure 41. Average response time is in minutes.**

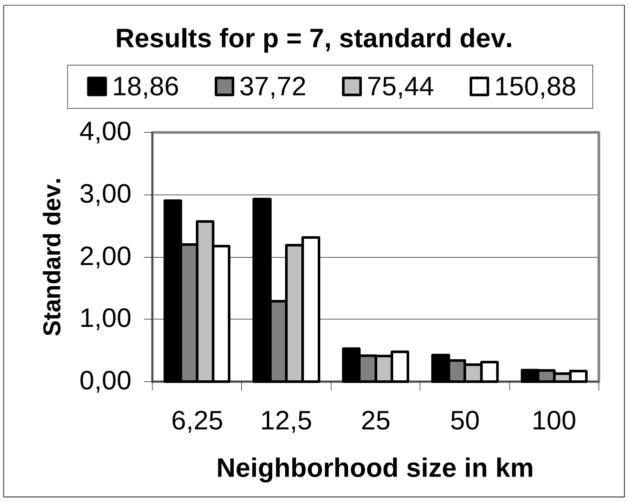



**Figure 42. Deviations are in minutes.**

A quick glance at Figure 39, Figure 40, Figure 41 and Figure 42 gives the explanation for the poor/unstable performance by the heuristic: The definition of the neighborhood is the reason. All solutions with a neighborhood of a 100 km radius, which for most ambulance points would be all other ambulance points, is both superior when it comes to average solution value and standard deviation. A low standard deviation is an indication of a robust set of parameters. The conclusion is that the neighborhood definition should be changed, so that a neighboring solution is a solution in which one of the ambulances is moved to any other ambulance point (which is not already in the solution), refer to section 13.2.1: "New Neighborhood." for more details.

It seems impossible to conclude anything on $rate_{max}$, it does not seem to play any importance at all, which is another indication that the heuristic was not performing at all or that all values of $rate_{max}$ were set either too high or too low.

Since the p-center problem was suffering from the same problems as the p-median, its neighborhood definition was changed as well.

### 13.2.1 New Neighborhood Definition

The consequence of the previous test is that a new neighborhood structure is defined:

> *A neighbor to a solution is a solution where one and only one of the ambulances is moved to another ambulance point where there is no ambulance.*

*As a consequence it takes more time to run the heuristic per iteration, since testing 3% of 100 neighboring solutions is faster than testing 3% of 1053 solutions, roughly 10 times. Instead of taking about one or two seconds for each run it now takes about a minute. However that does in no way compromise the heuristic, since it is still fairly fast.*

A new test on $rate_{max}$ is carried out and the results are presented in the next section.

## 13.3 Testing the Noising Method for the P-median Problem

*This section deals with the testing of the Noising Method (NM) on the p-median problem, using the revised neighborhood definition from section 13.2.1:”Test of Solution Method”.*

### 13.3.1 Testing the Effect of Noise, $rate_{max}$

Despite that the $rate_{max}$ had no effect in the previous test, a new test is carried out with the new neighborhood definition using the following parameters:

| *p* | *Iterations* | *Runs* | *Noise values - $rate_{max}$* |
|---|---|---|---|
| 3 | 1000 | 25 | {0, 0.75, 2.26, 6.79, 20.37, 61.10} |
| 7 | 2000 | 25 | {0, 0.75, 2.26, 6.79, 20.37, 61.10} |
| 9 | 2000 | 25 | {0, 0.75, 2.26, 6.79, 20.37, 61.10} |

The choice of the $rate_{max}$ values was primarily based on new test runs to see how they performed. It is clear that for the $rate_{max}$ set to zero is basically a decent, however different from a straight descent since only 3% of the neighborhood is tested at a time. The highest noise value was chosen based on the amount of worse solutions accepted by the heuristic. For the highest $rate_{max}$ value of 61.1 and $p$ set to 3 ($p$ set to three has the highest solution value for the three $p$-values), the following run seemed typical. More extreme values have also been tested, but they did not make a difference. The effect of $rate_{max}$ = 61.1 can be seen in Figure 43 and Figure 44.

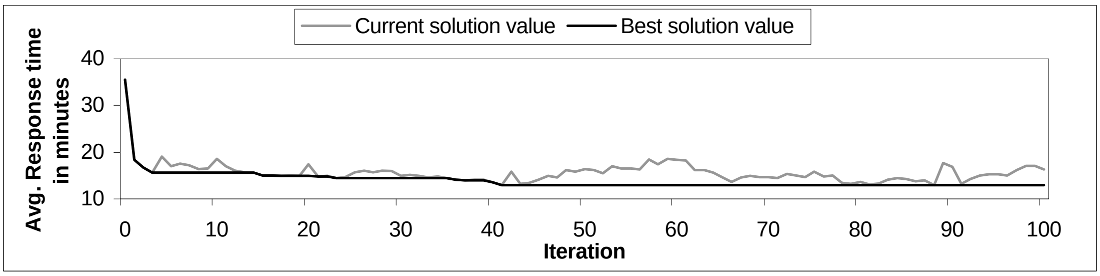



**Figure 43. The first 100 iterations of a run (p-median) with $rate_{max}$ = 61.1, and $p$ = 3.**

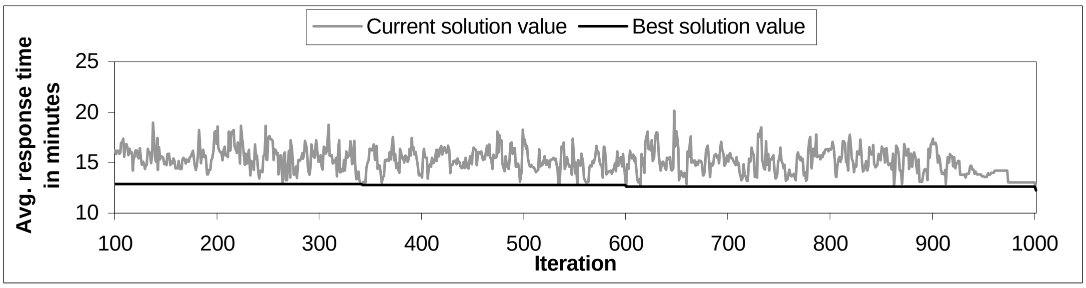



**Figure 44. The last 900 iterations of a run (p-median) with $rate_{max}$ = 61.1 and $p$ = 3.**

It is clear that the 61.1 is a high value since the current solution is allowed to vary quite a bit, not till around the 900th iteration ($rate_{max}$ = 6.11) does there seem to be a significant change in the pattern for the current solution. A summary of the test results is presented in Figure 45 with three ambulances ($p$ = 3), Figure 46 with seven ambulances ($p$ = 7) and Figure 47 with nine ambulances ($p$ = 9).

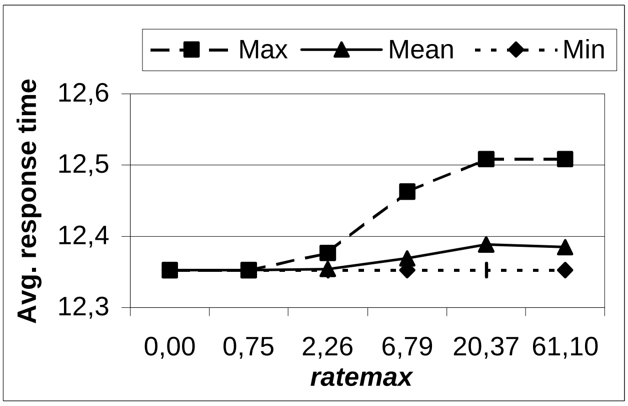



**Figure 45. Comparison of solution value and max noise level for p = 3, with 1000 iterations.**

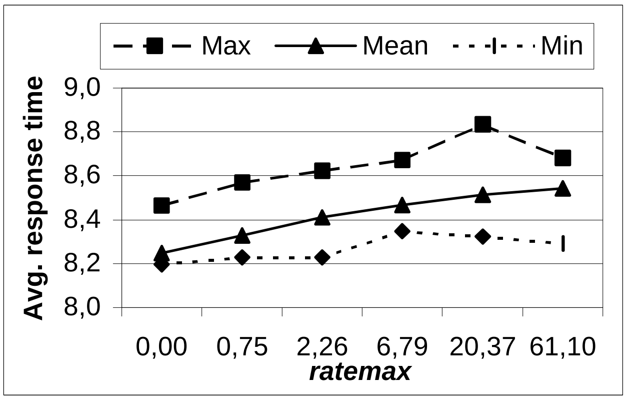



**Figure 46. Comparison of solution value and max noise level for p = 7, with 2000 iterations.**

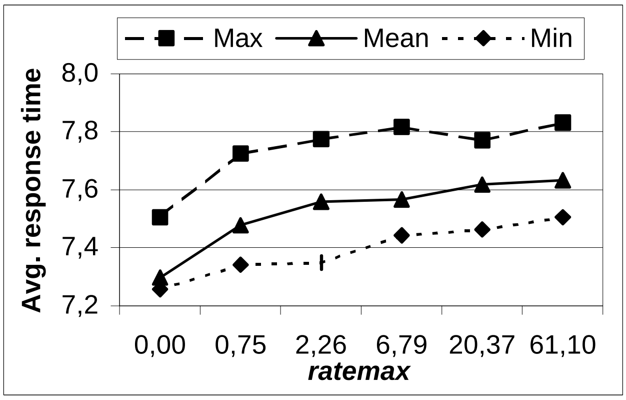



**Figure 47. Comparison of solution value and max noise level for p = 9, with 2000 iterations.**

The reason for testing only 1000 iterations for p = 3, was based on minor previous experiments that always yielded the same value for $rate_{max}$ = 0. At least once for every 25 runs at each $rate_{max}$, a solution of the value 12.35 minutes was found (the same solution). This is believed to be the optimal solution. For the two lowest $rate_{max}$ this solution was found every time. The performance of the heuristic clearly seems to be best at the lowest $rate_{max}$. A conclusion, which is supported by the runs for seven and nine ambulances, is shown in Figure 46 and Figure 47. There are minor deviations from this tendency. The extreme solution values for $rate_{max}$ = 61.1 and $p$ = 7 are clearly better than for $rate_{max}$ = 20.37, though the mean value is worse.

Having no noise, $rate_{max}$ = 0, corresponds to descending, though the neighborhood structure makes it an unusual descent, since it will not necessarily end up with the same solution when given the same starting solution. Why the descent is better than a real Noising Method is not really clear, one reason though could be that the number of iterations carried out are too few to allow the noise to take effect. The noise is only expected to have a beneficial effect once the solution is at a local minimum close to one.

Another guess could be that it is the unnoised descent on the best found solution that makes the difference, though in that case there should not be that big a difference between the different values of $rate_{max}$. The unnoised descent, however does have an effect, for seven and nine ambulances the descent always improved the solution, though never by more than that 12 seconds. For three ambulances only eight out of the 25 runs are improved by the unnoised descent.

To verify the effectiveness of NM, despite the fact that the noising had no effect, 25 descents were performed on random solutions with three ambulances, and 1000 iterations. The results can be seen in Figure 48.

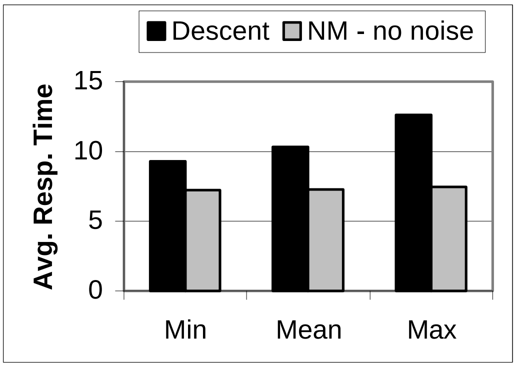



**Figure 48. Effect of descent algorithm (full neighborhood explorartion) only on *p* = 3.**

Figure 48 confirms that it is not the descent in itself that makes the good solution. However using the descent algorithm corresponds to setting the neighborhood exploration to 100% instead of 3%. The 100% exploration is clearly less effective than the 3%, combined with the 100%. It could of course be interesting to investigate the percentage as well as descending until no improvement is found using the 3% exploration. Nevertheless descending on the 3% level might fail if a bad 3% is chosen $p$ times in a row, since the 3% is chosen on a random basis.

### 13.3.2 Evaluating the Effect of the Number of Iterations

The effect of doing 1000, 2000 and 4000 iterations was tested for four, seven, nine and twelve ambulances with zero noise over 25 runs. The result for four ambulances has been left out since it found the same solution every time no matter the number of iterations. The results for $p$ = 7, 9 and 12 are presented in Figure 49, Figure 50 and Figure 51.

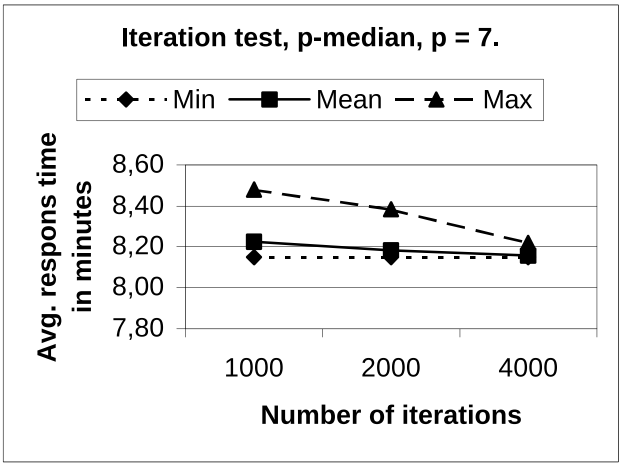



**Figure 49. Effect of number of iterations, for p = 7.**

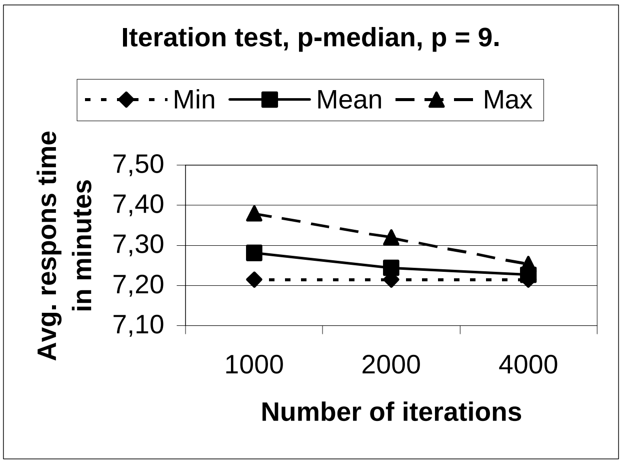



**Figure 50. Effect of number of iterations, for p = 9.**

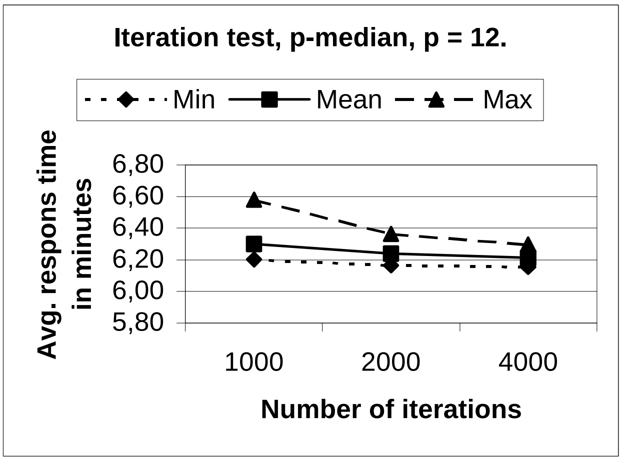



**Figure 51. Effect of number of iterations, for p = 12.**

It does not come as a surprise that the solution value decreases when the number of iterations increases. More surprisingly the impact is not that big on the mean and minimum values. The minimum value found in 25 runs was only improved for $p$ = 12, from 1000 to 4000 iterations, and then by less than one percent, 0.65% to be exact. The most significant effect of increasing the number of iterations is seen when reducing the worst solution over 25 runs. The biggest improvement on the maximum value was also found for p = 12, a mere 6.82% was the benefit of quadrupling the number of iterations. For p = 12 the difference between the maximum value and the minimum value was as small as 2.28% (less than 10 seconds) for 4000 iterations.

In general more iterations provide more reliable solution values, but not necessarily better. That even solutions with 1000 iterations were quite good suggest that the noise should have been able to work on the previous runs with 2000 iterations, hence it does not seem to be the reason why zero noise is the most effective setting for $rate_{max}$.

### 13.3.3 Model Validation - p-median Problem

In order to obtain a better understanding of the consequences of using the p-median problem, with a 2140 hexagonal tessellation, solutions for three to twelve ambulances were found using the NM with zero noise and 2000 iterations. For one to two ambulances the "Try All Algorithm" from section 12.4: "Checking All Possible Solutions for the p-center and p-median Problems". The results confirm the robustness of the solution method, as presented in Figure 52:

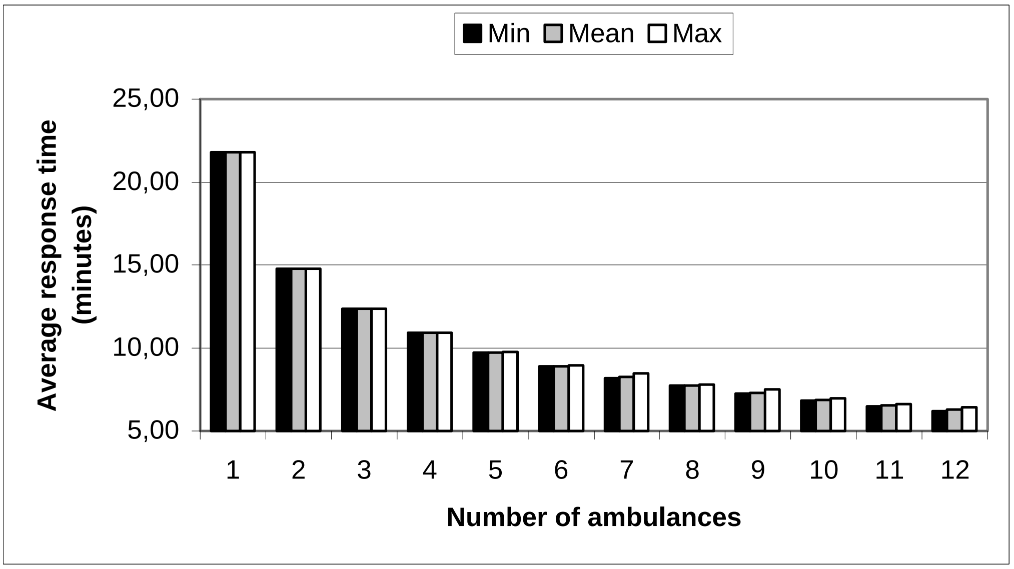



**Figure 52. Min, Mean and Max solution values for 25 runs, with 2000 iterations and NM-zero-noise.**

The differences between worst and best solutions are so small that only from six ambulances and up is it possible actually to see a difference in Figure 52. The largest deviation is for nine ambulances 3.43%, less than 15 seconds. During the tests of the p-median problem, several tests, 25 runs of seven and nine ambulances with zero noise, were performed more than once. A difference could be observed though very small. The 25 run sets with the best min-solution are used in Figure 52 and the rest of this report.

The best solutions for each o the three different problems are presented in Appendix IV.

An interesting thing to note about the solutions is that despite the very similar solution values, the allocation of the ambulances can actually be quite different. Figure 53 shows an example of this:

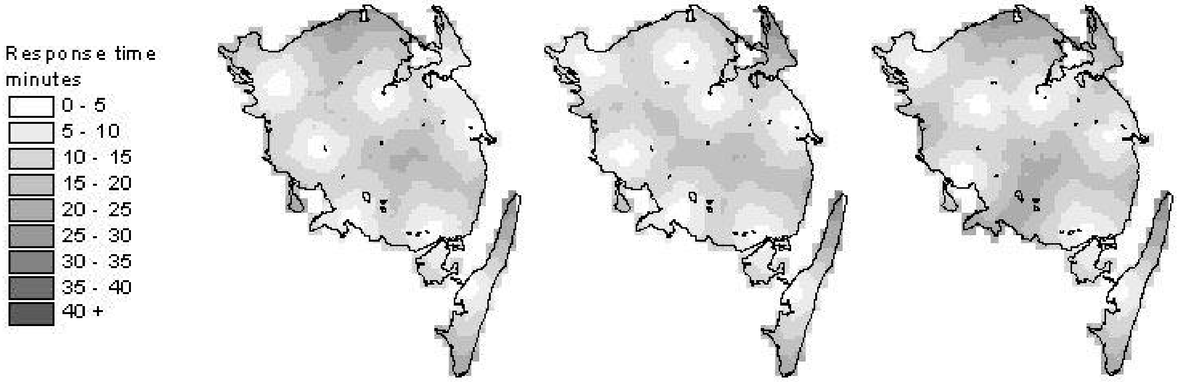



**Figure 53. Response for best found solutions of 8 ambulance, from left p-median - Falck garages (NM) , p-median - demand points - NM, MFLA- MRC. (Ambulance are located (roughly) in the center )**

It might be difficult to see that there are actually eight ambulances on the left and right image. They are there northeast/east of Odense, at the towns of Munkebo/Kerteminde. The very similar solution values and the different allocations is an indication of that there are many possibilities of finding good solutions to the location allocation problems, hence finding a good solution might not be that difficult. An explanation of how the maps are generated is explained in Appendix IV.

#### 13.3.3.1 Use of Demand Points

An interesting thing to note about the p-median problem is that only 26 different demand points were used as positions for the ambulances for the twelve different problems out of 78 possible. It is especially interesting that four demand points were used in eleven, ten, seven and six of the solutions. The one used in eleven of the solutions is a demand point in Svendborg, the one used in ten solutions in Odense, the one used in seven solutions is in Rudkøbing (on Langeland) and the one used in six solutions is placed in Faaborg,Figure 13 shows the position of the towns on Funen. The four locations are presented in Figure 55, together with the Falck garages in those four towns, the gray line are roads, the ticker the line the higher the speed limit.

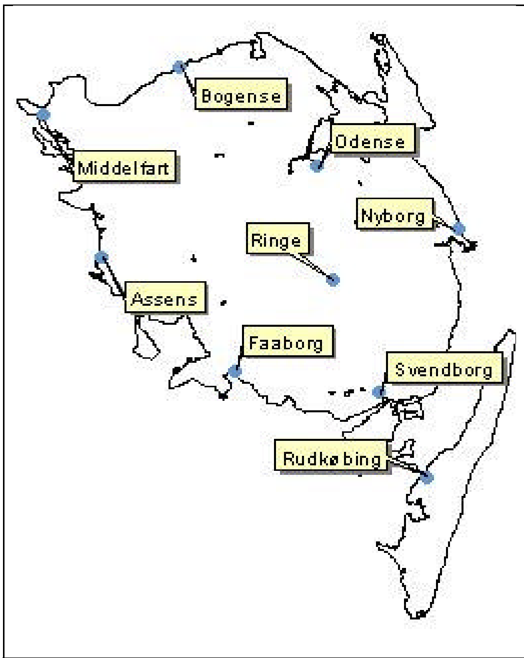



**Figure 54. Positions of major towns on Funen.**

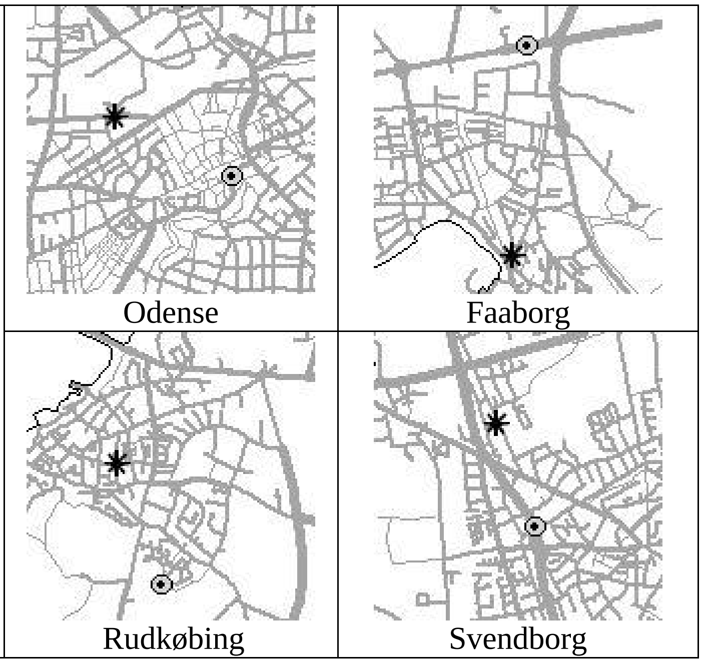



**Figure 55. Positions of frequently used demand points (Θ) and Falck garages (*). In 1: 50,000.**

The maps of the towns are made in 1:50,000, hence two centimeters equal one kilometer. The positions of three of the demand points seem to be quite good (Odense, Faaborg and Svendborg) all located on a road with a high speed limit close to other roads with high speed limits, hence a good infrastructure. In Rudkøbing on the other hand the position is not so good, it seems to be located at the outskirts of the town, requiring a travel of at least some hundred meters to get near a good position, a consequence of using a distance of 2140 meters between demand points. It is also interesting that for all four towns the distance (Euclidean) is about a kilometer between the demand points and the Falck garages.

### 13.3.3.2 Capacity

None of the three models applied in this project has made use of capacity constraints, even though they are all expected to suffer from the same problem. With a high concentration of demand in the Odense area, one ambulance is likely to have to cover a significantly larger demand than the other ambulances, perhaps so large that it is unrealistic in real life, though Falck's dynamic resource allocation might be able to make up for it. The results from the p-median problem illustrate the disadvantage of not using capacity constraints optimally. Figure 56 shows the demand (load) covered by each of the ambulances in the best solution for twelve ambulances. Figure 57 shows the load on the ambulances, which covers the highest demand in the best solutions from two to twelve ambulances.

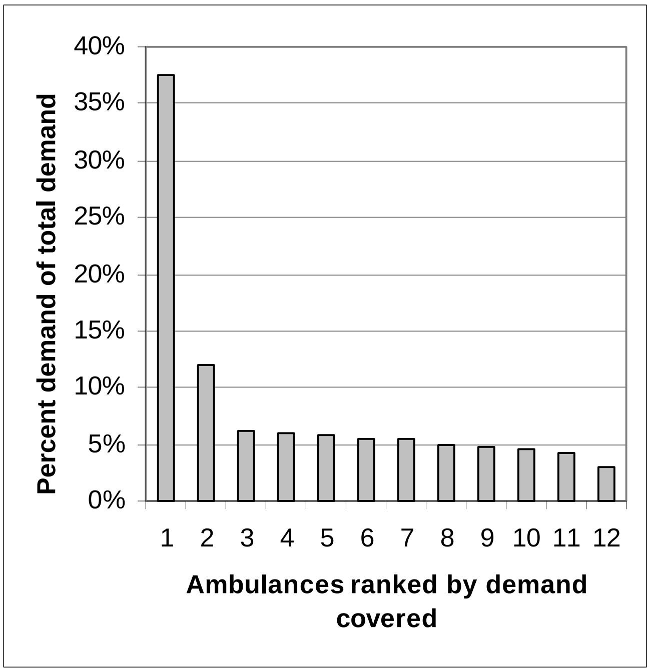



**Figure 56. Demand covered by each ambulance (ranked by load) for the best solution with twelve ambulances.**

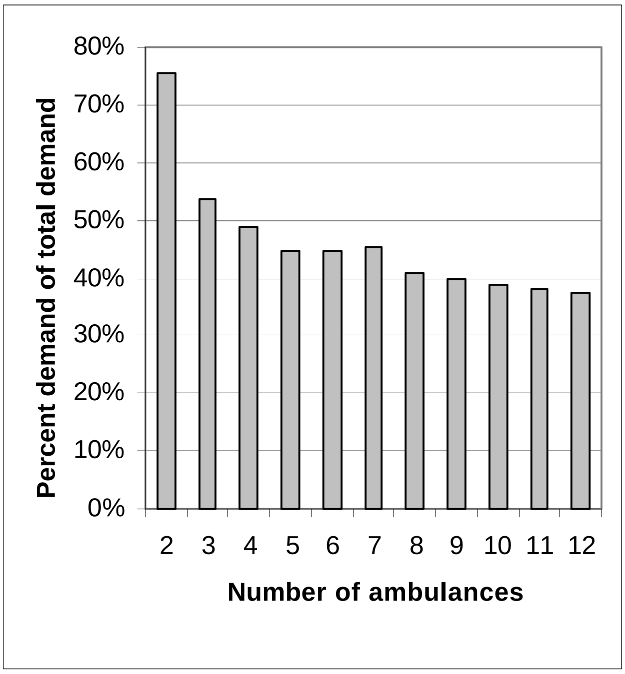



**Figure 57. Demand covered by the highest loaded ambulance in the best solutions from two to twelve ambulances.**

Though some of the demand is a virtual demand (1053 accidents were added), it is quite clear that one out of twelve ambulances covering 37% of the demand might not be a feasible solution in real life. On the other hand if the total number of ambulances available were only three or less 37% would not be a high number, which also illustrates the problem of introducing capacity constraints. What is the capacity of an ambulance? Should the capacity vary depending on the number ambulance available?

The first question cannot be answered without answering the second question. The second question involves the problem of explaining why an ambulance sometimes can cover more than other times. The answer could be “To ensure the best coverage with the resources available.” Though I still find it troubled to introduce capacity constraints, especially considering that demand points can be allocated to a non-nearest ambulance, as mentioned in chapter 11: “Models”, capacity constraints might still solve the problem of allocating too “few” ambulances to areas like Odense.

### 13.3.4 Conclusion on the p-median Problem

In general there does not seem to be that many problems with the p-median problem and the solution method, NM. Nonetheless further investigations into why the noise has a negative effect on solving the p-median problem and the size of the neighborhood would be advantageous, despite the robustness of the method. The main problems with the p-median problem are the positions of the demand points and low number of ambulances in areas where there is a high demand.

## 13.4 Testing the Noising Method for the p-center Problem

*This section deals with the testing of the NM on the p-center problem, using the revised neighborhood definition from section 13.2.1:"Test of Solution Method".*

As for the p-median problem a test is carried out to estimate the best value of $rate_{max}$. Using the following parameters:

| *p* | *Iterations* | *Runs* | *Noise values - $rate_{max}$* |
|---|---|---|---|
| 3 | 1000 | 25 | {0, 25, 200, 10,000, 1,000,000} |
| 7 | 2000 | 25 | {0, 25, 200, 10,000, 1,000,000} |
| 9 | 2000 | 25 | {0, 25, 200, 10,000, 1,000,000} |

The result is quite surprising as can be seen on Figure 58, Figure 59 and Figure 60. It is important to note that allocation (of accidents) error found in section 9.3: "Evaluating WPA and CCPA" was larger than four minutes, hence the actual solution value could be both more or less than four minutes from the solution values presented for the p-center problem. There has been done no attempt to investigate the actual error. Though a study into how the error for the p-center problem could be very useful to ensure a more reliable model.

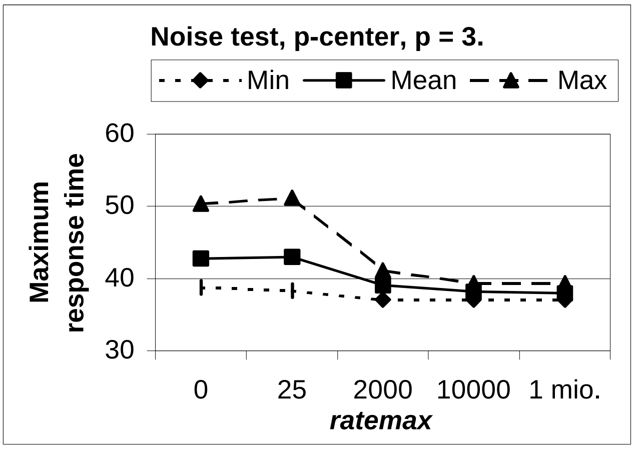



**Figure 58. Effect of *$rate_{max}$*. p-center problem, p = 3.**

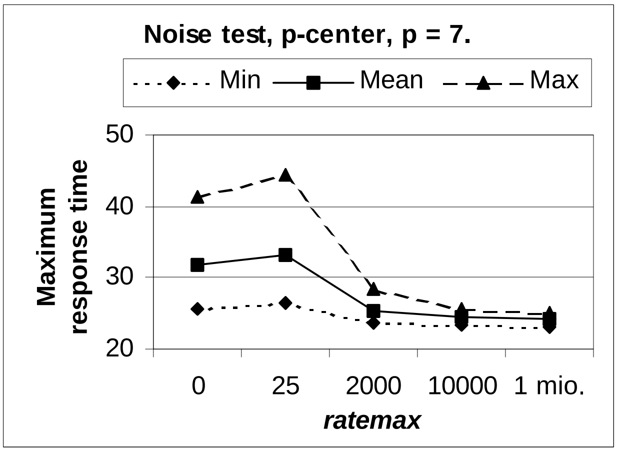



**Figure 59. Effect of *$rate_{max}$*. p-center problem, p = 7 .**

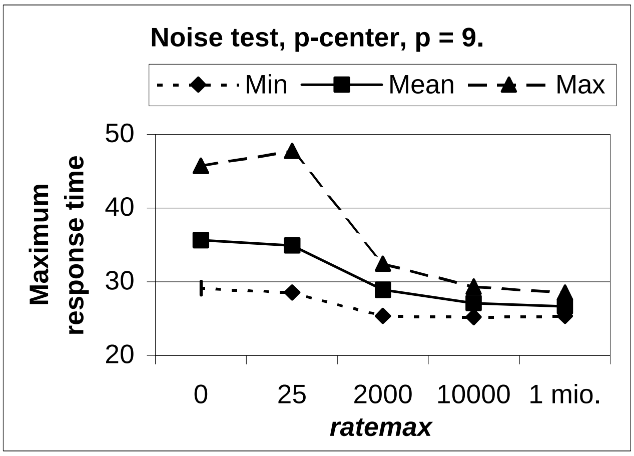



**Figure 60. Effect of *$rate_{max}$*. p-center problem, p = 9.**

Opposite of the p-median problem the most effective parameter for $rate_{max}$ seems to be as high a value as possible. High values of $rate_{max}$ are nearly the same as a random search, which is a bit worrying. The reason could be that when exploring only 3% of the neighborhood there is a larger risk of examining only very bad solutions. Hence with low $rate_{max}$ values the NM stays in the same spot until an improvement is met, in that way a lot of iterations could be "wasted". The solution value for the p-center is only determined by one demand point, consequently moving an ambulance to another ambulance point does not necessarily change the solution, which in most cases is the opposite of the p-median problem. This could be the explanation of the difference in the "optimal" value for $rate_{max}$. No further tests of the p-center problem were performed. An iteration test was carried out but the result was very similar to the one from the p-median problem, and the effect of descending is not expected to be much different than from the p-median problem.

### 13.4.1 Model Validation p-center Problem

The validation of the p-center model will not be as thorough as for the p-median model, the reason being that the p-center model in itself is not preferable as a method of allocating ambulances, as it in no way considers the average response time. As a consequence Odense and other very demanding areas can be left with high response time. Figure 61 illustrates the problem in the best p-center solution for seven ambulances.

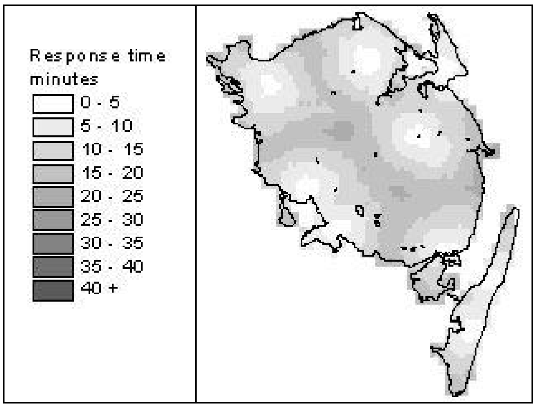



**Figure 61. Best p-center solution for seven ambulances.**

Some areas of Odense have more than 15 minutes in response time, instead two ambulances are placed on Langeland, many would probably find such an allocation of resources unacceptable. This is not a problem for all the best solutions for the p-center problem, though it is a practical as well as a theoretical problem, which must be taken care of. For the same reason the average response time for the p-center problem is irrelevant, because the p-center problem does not try to control the average response time. However the p-center solutions operate with a limit for how low the maximum response time can be, which can be used in a p-median problem with limits on the maximum response time. A suggestion on how a reduction of the maximum response time can be introduced is presented in section 14.2.1: "Marginal cost on maximum response time".

### 13.4.2 Conclusion on the p-center Problem

The p-center problem as a model for ambulance allocation is not relevant, though the results for the p-center problem might be used in combination as limits on the maximum response time for the p-median problem. The NM for the p-center proved to be reliable, though not as robust as the NM for the p-median problem.

# 14 Allocation of Ambulance on Funen

*This chapter deals with a comparison of the placement of ambulances on Funen using either the Falck garages as location for the ambulances or the 1053 ambulance points, from the “Spring at Work” dataset.*

Results ranging from one to twelve ambulances have been obtained using the solution methods from 11: “Models” and the best parameters from chapter 13: “Test of Solution Methods”. Only the best results are used.

The results of using the demand points as ambulance points will be compared using the 14 Falck garages on Funen as ambulance points. The Moising Method (NM) is used with the same parameter setting as for the p-median, and p-center problems with seven and eight ambulances, when using the Falck garages as ambulance points and for all other number of ambulances the optimal solution was found using the “test all” algorithm from section 12.4: “Checking All Possible Solutions for the p-center and p-median Problems.” The reason for using NM on seven and eight was that it would be too time consuming using the “Try all algorithm” (more than two days each).

The reason for testing no more than twelve ambulances is based on an upper bound set by Falck to 10 ambulances available at a time. The number was increased to twelve some way along the project due to a misunderstanding that there were twelve Falck garages, though at some point it was discovered that there were in fact fourteen. With the results for twelve already available, there was not reason to leave them out.

A full presentation of the best solutions can be found in Appendix IV for the following solved problems:

- MFLA (Multi Facility Location Allocation model)
- p-median, demand points (“Spring at Work”) as ambulance points
- p-center, demand points (“Spring at Work”) as ambulance points
- p-median, Falck garages as ambulance points
- p-center, Falck garages as ambulance points

all found for one to twelve ambulances, except “P-center, Falck garages as ambulance points.” where only the six best are presented (no improvement beyond seven),

Two comparisons will be made, one comparison on the average response time and one comparison on the maximum response time.

## 14.1 Minimizing the Average Response Time

The best results for the NM p-median, MRC - MFLA and Falck garages p-median are presented in Figure 62.

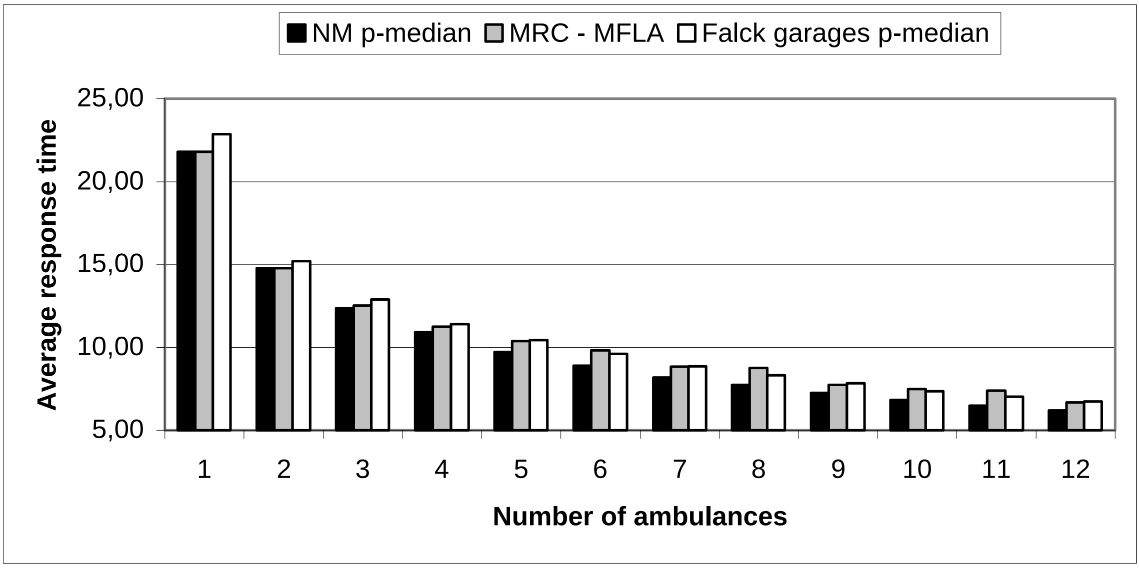



**Figure 62. Best results of NM, MRC and using Falck garages.**

The NM is superior for all number of ambulances, though the difference between the MRC (Multi Restart Cooper) and the NM is less than 0,1% (in favor of NM for one ambulance). The MRC is as much as 14% worse than NM (eight ambulances) and using the Falck garages proves to be up to 9% worse (twelve ambulances). Nevertheless the differences are so small that all three allocation-types must be considered to be good. Indicating that the positions of the Falck garages are quite good when minimizing the average response time.

## 14.2 Minimizing the Maximum Response Time

The comparison of the maximum response time will include data from minimization of the average response time. By comparing the two different objectives, the “cost” of the maximum response time on the minimization of the average response time can be evaluated. The results on the maximum response time are presented in Figure 63.

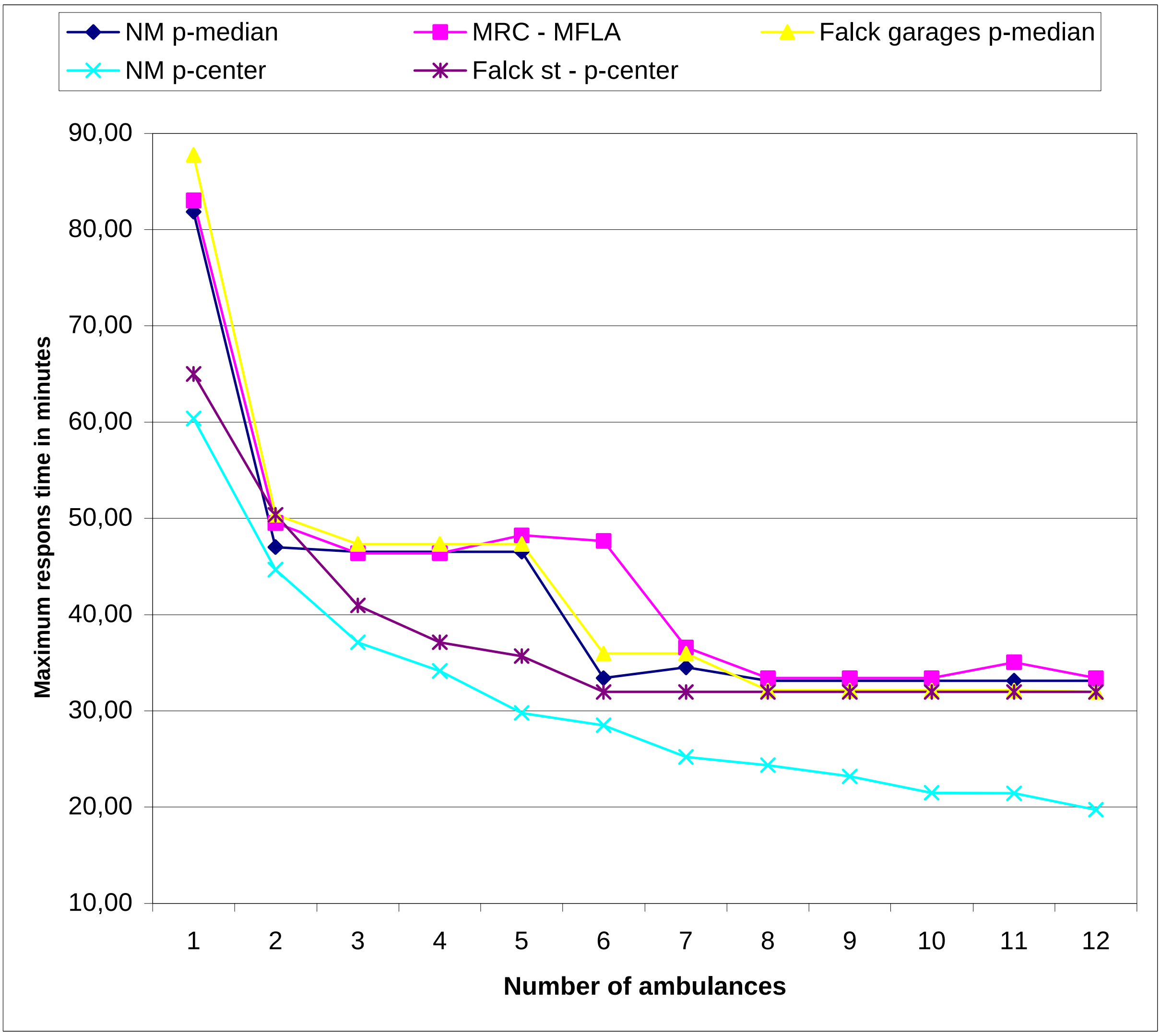



**Figure 63. Maximum response time for the five different problems.**

The positions of the Falck garages are not as good when the objective is minimizing the maximum travel distance. From one to six ambulances the maximum response time can be improved by three to six minutes, from seven it is 6.7 minutes and increases to 12.2 minutes for twelve ambulances. The reason that the difference increases from six ambulances and up is that when using Falck’s garages the maximum response time cannot get any lower than 32.0 minutes, whereas using the demand points as location for ambulances allows for improvement. When it comes to models that minimize the average response time a different pattern is shown. The time does not decrease "continuously" as it did for the p-center model in steps. The reason being that a single or a few points with a high response time does not have a heavy influence on the solution value, in theory the maximum response time for the p-median problem could have been the same whether one or twelve ambulances were used. It is also noteworthy that for eight or more ambulances the maximum response time, when minimizing the average response time using the Falck garages as location for ambulances, is nearly the same 32.1 minutes (32.0 for twelve ambulances) for the p-median model and 32.0 for the p-center model, so it is actually not the same solution from eight to eleven ambulances. It is also interesting to see that the maximum response time, when using the p-median model and the demand points, is actually higher than when using the p-median model and the Falck garages for eight and more ambulance (about a minute).

### 14.2.1 Marginal cost on maximum response time

As mentioned in section 13.4.2: "Conclusion on the p-center Problem" it is not advisable to calculate the average response time when using the p-center model. The reason for investigating the p-center model was only to get an estimate of the maximum response, which can then be used in the p-median model as a bound on the maximum response time. The problem with five ambulances is chosen with the bounds 30, 32.5, 35, 37.5 and 40 minutes, based on the value from the best p-center solution being just below 30 and the maximum response time for the best solution to the p-median problem with five ambulances being 46.6 minutes. The bounds are enforced by setting the cost of response times between demand points, which were above the limit, to 10,000, it could not be set any higher due to numerical problems. 25 runs were made for each limit. For 32.5 and 30 minutes some of the runs returned unfeasible solution, though also several feasible solutions were found. The results for the best solutions found over 25 runs using the same parameters as for the p-median problem (2000 iterations) are presented in Figure 64.

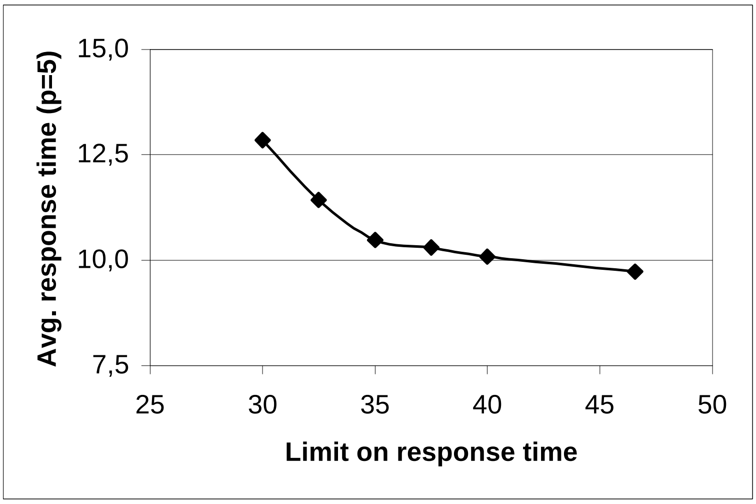



**Figure 64. Best solutions for five ambulances with limits, NM p-median.**

The best solution with no limits had a 9.7 average and a 46.6 maximum response time, it has been included on the graph. As could be expected the marginal cost on the average response time is not that great for the high limits 40, 37.5 and 35, averaging 3.9 seconds in average response time per minute. From 35 to 30 minutes as a limit the marginal cost averages 28.5

seconds per minute, a substantial increase comparing 46.6 to 35 minutes. The results have been plotted to study the geographic change in allocation, Figure 65.

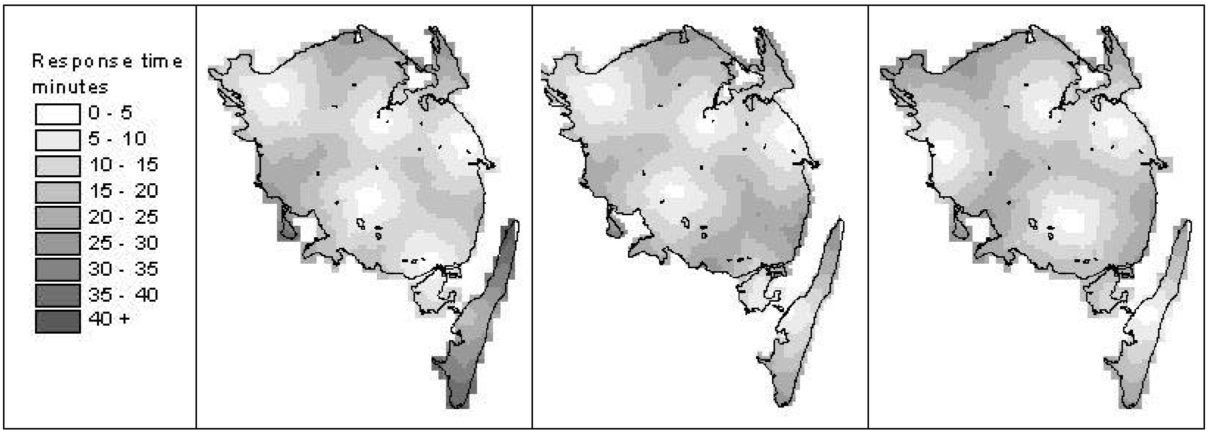



**Figure 65. Best solutions for p-meidan models, from left no limit, 35 minutes limit and 30 minute limit (nearly the same as best p-center solution).**

From the no limit situation to the 35 minutes limit situation, only the two southern ambulances have been moved. One has been moved a little bit so the west to cover the two small peninsulas within the limit, and the other has been moved from the island of Tåsing to the island of Langeland, to ensure the coverage there. From the 35-minute limit situation to the 30-minute limit situation all ambulances have had to move, which makes sense since the increase in average response time was rather high. All three northern ambulances are moved from highly “congested” areas to positions where they can help fulfill the limit.

This way of using the p-median model and the Noising Method could prove to be a new powerful tool for Falck, since it provides an idea of the cost of reducing the maximum response time.

The test could of course also have been made on other p-values (with different limits), providing Falck with valuable information on how much they can agree to meet in future contracts. Figure 64 shows that somewhere between the limit on 32.5 and 35 minutes the cost on the average response time increases dramatically.

## *14.3 Other Possible Uses of the Models and Solution Methods*

There are several other possibilities in which the models could be used to provide additional information. One of them is to investigate not the primary response time as done in this project, but the response time from 2$^{nd}$ and 3$^{rd}$ closest ambulances. Such information could help determine the vulnerability of a solution, when an ambulance is “removed” from the solution, this being the weak side of the models used. Perhaps the solution methods for the p-center and p-median problems could be altered to optimize for 2$^{nd}$ or 3$^{rd}$ closest ambulance, to reduce the weakness of the solutions. The relationship between the p-center and the p-median problems could also be investigated by combining the two objectives into a multi objective model, setting a priority factor on each objective. Such a model might be solved using the Noising Method.

# 15 Dynamic Allocation of Ambulances

The project started with the intention of developing a model that could be used to dynamically allocate ambulances. As the project advanced problems, which had not been foreseen, arose:

- A reference for comparison
- Behavior and positions of the ambulances
- Rules for reallocation of ambulances

In order to evaluate the performance of a dynamic allocation, a reference is needed. The reference should have been the Falck garages as positions for ambulances. Placing ambulances in Falck garages or on the demand points turned out to be equally effective, hence there is not much reference left, since no major improvement could be expected.

Another problem was modeling the behavior of the ambulances while they are moving. It is necessary to be able to calculate the position during reallocating, because ambulances, which are ”done” with a job/accident are then available to be assigned to a new job. In real life the positions of the ambulances can be found using GPS, a technique of which Falck is already using. In a simulation in GIS, the position needs to be calculated. The travel time depends on the speed limit, which varies over the road network, making it rather complicated and time consuming to calculate, compared to using a GPS position.

Defining the rules for the ambulances of when and how to reallocate the ambulances is another problem. It is clear that both for Falck and it’s crew, moving all ambulances around constantly to ensure an optimal allocation is not feasible. It is costly to have ambulances moving all the time, they consume petrol and the crew is likely to get very displeased by constantly moving a few kilometers, secondly having too many ambulances outside Falck’s garages conflicts with the dynamic allocation of resources within Falck’s organization.

There could both be limits on the:

- Number of ambulances that are allowed to move at a time
- Number of ambulances allocated outside a Falck garage
- Distance or amount of time that an ambulance is allowed to move when reallocated

Setting a limit on the number of ambulances that are allowed to move raises more difficulties. Is an ambulance, which is returning from an accident on the ”move”, or not? Is an ambulance, which is reallocating on the move, or not? If the answer to both questions is no, then the result might be that all ambulances end up driving around anyway. If the answer is yes to any of the questions, then it might become impossible to reallocate any of the idle ambulances. A discussion with Falck revealed that one perhaps two ambulances could be allowed to be reallocated each time the number of idle ambulances changes, i.e. an ambulance is called away to an accident or one ambulance returns from an accident. With so few ambulances moving, about a thousand demand points and the other limits on reallocating the ambulances, the number of possibilities becomes so low that from an operations research point of view it is not that interesting. Hence it falls outside of the subject, which this project was intended to cover, since it will be possible to try all possible solutions.

# 16 Perspectives

This project has shown that relatively simple heuristic methods can be applied in many ways in order to provide Falck with new information and confirm assumptions on the quality of the service that Falck provides on Funen as well as other areas in Denmark and abroad. Falck and other companies will also be able to use methods applied in this project to other areas than ambulance service. Falck's security service is an area, which could be very interesting to investigate.

This project has also revealed some problems in using GIS in the allocation of ambulances, which could be investigated further to increase the reliability of other studies.

The altered road network used in this project is a simple model of the travel time / response time of Falck's ambulances. A further investigation into that area will provide models which are more reliable, and comparable to the real life performance of Falck. The stochastic elements of traveling on the road network and the influence of rush hour, holiday traffic and other time depending variations in traffic could also prove useful.

Another study worth doing is a research into the effect of choosing better demand point positions, as mentioned in section 9.3: "Evaluating WPA and CCPA" as well as increasing the number of demand points. If more demand points are introduced, or areas with a greater extent than Funen are to be investigated, a more efficient way of calculating the all-to-all shortest paths is crucial.

The study from chapter 10: "Accident Analysis" of the geographic pattern of accidents in different time periods and their influence on ambulance allocation would also be worth investigating further. The possibility of combining the accident data with other data such as demographic data, land use etc. could be very interesting.

A study into the effects of dynamic allocation could also be interesting, though also difficult to handle since it might be necessary to explore a variety of different strategies to get a useful impression of what is the best dynamic allocation strategy. However a method for testing all possible solutions efficiently will be useful.

The algorithms used in this project could also benefit from further studies. Especially an explanation of why the $rate_{max}$ is best at two opposite extremes. A study into the quantity exploration percentage of the neighborhood might provide even better algorithms. A study of introducing capacity constraints or other methods to ensure coverage when an ambulance in highly demanding areas is called to an accident might prove valuable.

# 17 Conclusions

The first part of this project concluded that even though using the travel time to allocate accidents to the nearest demand points is preferable when there is enough time, the Euclidean distance can also be used to allocate accidents although it involves a little more risk. Travel time should be used if a large number of accidents take place near lakes, rivers and other areas outside roads, to avoid accidents being allocated to demand points, which in terms of travel time are distant. The first part of this project also showed that the type of tessellation, hexagonal or quadratic, does not seem to play any role at all, only the distance used for the tessellation is important.

It was also shown that there is a pattern for the accidents over time as well as in terms of geography. Furthermore it was made probable that there is a difference in the geographic occurrence over time, which might be significant for the allocation of ambulances.

The second part of this project provided fast, efficient, robust and easy to implement algorithms for the two discrete problems: the p-median (minimization of the average response time) and the p-center (minimization of the maximum response time) problems, though the parameter settings for the two problems are a bit odd. The continuous allocation of ambulances using Euclidean distance, proved to be very efficient on Funen, though it suffers from the same problem as the Euclidean distance allocation of accidents to demand points, as well as it might place ambulances in areas, which traditionally are inaccessible to ambulances such as the sea. Still the algorithm was also much faster than its discrete competitors but also far less robust and reliable. The recommendation is undoubtedly to use the p-median problem to allocate ambulances, perhaps with limits on the maximum response time, to allocate ambulances. Using the p-center model involves the risk of high response time to highly demanding areas.

The use GIS and OR combined has proven powerful and in the future it might be possible to develop systems, which can slightly improve the current performance of Falck’s ambulance service, though more complicated models will have to be applied.

# References


AM "ArcView GIS" " (Manual). Environmental Systems Research Institute, Inc. 1996.

DCE "Designing and Reporting on Computational Experiments with Heuristic Methods". Barr, R. S.; Golden, B. L.;Kelly, J. P.; Resende, G. C. M.; Stewart, W. R, jr. Journal of Heuristics, 1: 9-32 (1995), Kluwer Academic Publishers.

DLT "Discrete Location Theory". Mirchandani, Pitu B.; Francis, Richard L. (editors). Wiley-Interscience, 1990. ISBN: 0-471-89233-5.

DSA "Data Structures, Algorithms and Applications in C++". Sahni, S. McGrawe-Hill Book Co. ISBN: 0-07-109219-6. 1998.

DZ This reference is taken from [FLMM]. Is has not been examined. "The Planar Two-center and Two-median Problems", Drezner, Z. Transportation Science, 18:315-361 (1984),

FL "Facility Location: Applications and Theory". Drezner, Zvi; Hamacher, Horst W. (editors). Springer-Verlag, 2002. ISBN: 3-540-42172-6.

FLMM "Facilities Location: Models & Methods". Love, Robert F. (et al.). North-Holland, 1988. ISBN: 0-444-013031-9.

HA This reference is taken from http://www.math.gatech.edu/~shenk/constrainedGA.pdf . Is has not been examined. "Cooling Schedules for Optimal Annealing". Hajek, B. Math of Operations Research, 13 (1988).

HJ http://www.imm.dtu.dk/courses/02721/oh7.pdf. (26th of August 2002).

HS "Løsning af Lokationsproblemer med GIS". Sørensen, Heino. ELTEK, Technical University of Denmark, 2000. (Master thesis)

MATH http://mathworld.wolfram.com/StirlingNumberoftheSecondKind.html (31st of July 2002).

NCJB This reference is taken from [DLT]. Is has not been examined. "A tree search algorithm for the -median problem." Nicos Christofides and John E. Beasley. European Journal of Operational Research, 10:196-204, 1982

NMG "The noising methods: A generalization of some metaheuristics". Charon, Iréne; Hudry, Olivier. European Journal of Operational Research, issue 135 (2001) pages 86-101.

SAM "ArcView Spatial Analyst" (Manual). Environmental Systems Research Institute, Inc. 1996.

SAP "Using Simulated Annealing in the Solution of Probabilistic Location Problems". Galväo , Roberto D.; Chiyoshi, Fernando; Morabito, Reinaldo. (Unknown source)

SL "Project 1". Salling, K. B; Lindeskov, C. K. Project report for the course: "Network and Integer Programming" at DTU, supervised by Clausen, J. 2001.

TB This reference is taken from [FL]. Is has not been examined. "Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph", Teitz, M. B.; P. Bart. Operations Research (1968), 16, 955-961.

## Definitions

| Term | Description |
|---|---|
| Coverage | In most cases it is an expression of how all ambulances in a given situation is able to respond to emergencies in general. |
| Demand points | Points representing accidents in an aggregated form. |
| Digraph | Is a graph/network in which is directed. Hence the cost of moving from A to B, might not be the same as moving from B to A. |
| Euclidian distance | The length of a direct line between two points. To most people it is just known as the length. |
| Modulo | The remainder when one number is divided by another. The remainder of 10 / 8 is 2, because 8 times 1 plus 2 equals 10. |
| Response time | The time between an emergency call is received to an ambulance reaches the injured. Both used about the travel time on the road network, the Euclidean distance and the real life response time. |
| Run | A run of an algorithm, which produces a random number, gives one random number. 25 runs give 25 random numbers. |
| Solution Space | A solution space is an abstract term for the set of feasible solutions. |
| Tessellation | Dividing an area into smaller parts (usually of equal size and geometry). |

# Appendix I Data

This appendix contain a few technical details on the data which might interest some readers of the project. Accidents and road network data will be explained.

### *Accidents*

The accident has been retrieved from one of Falck's databases. Each accident is actually not a record of the accident but a record of the assignment/job carried out by an ambulance, hence it includes the address of the accident.

| Field | Type | Description |
|---|---|---|
| OPGNR | Integer | Job/assignment number. (Unique) |
| VCNR | Integer | Vehicle number. |
| TIDMODTAGE | Date&Time | Time of call. |
| TIDOPTAGET | Date&Time | Time of arrival at accident. |
| TIDFRIREG | Date&Time | Time when done with assignment. |
| FRAGADENAV | String | Street name of accident. |
| FRAHUSNR | String | Street number of accident. |
| FRAPOSTNR | String | Postal code of accident. |
| VFRGADENAV | String | A more precise definition of FRAGADENAV. |
| VFRHUSNR | String | A more precise definition of FRAHUSNR. |
| VFRPOSTNR | String | A more precise definition of FRAPOSTNR. |
| TILGADENAV | String | Street name of where the injured is transported to. |
| TILHUSNR | String | Street number of where the injured is transported to. |
| TILPOSTNR | String | Postal code where the injured is transported to. |
| VTLGADENAV | String | A more precise definition of TILGADENAV. |
| VTLHUSNR | String | A more precise definition of TILHUSNR. |
| VTLPOSTNR | String | A more precise definition of TILPOSTNR. |
| UFSTNR | Integer | The garage, which the ambulance belongs to. |

"VFRGADENAV", "VFRAHUSNR" and "VFRAPOSTNR" are address information entered at a later point in time (after the call is finished). This is usually used when the information in "FRAGADENAV", "FRAHUSNR" and "FRAPOSTNR" is not an address, such as "In the mall", "At the Q8 gas station north of Svendborg" etc.

This is not the only information that Falck has in their database.

H. Sørensen [HS] discovered some errors, in his project on ambulance and Falck, in both address and time data [HS]. There has not been a check for these errors or any other errors expect for the large number of errors encountered by the early data “TIDOPTAGET” prior to 11th of January 2000 22:41 (10:43 pm), as mentioned in section 8.2.1: “Geocoding the Accidents”. The problem is illustrated below in Table 9.

| | |
|---|---|
| OPGNR | 315611 |
| TIDMODTAGE | 12-01-1999 12:12 |
| TIDOPTAGET | 08-02-2001 10:59 |
| TIDFRIREG | 08-02-2001 11:25 |

**Table 9. Error in accident data.**

Apparently the call is received some time early in 1999, but is the ambulance does not become idle before early 2001, hence it has taken more than two years to complete the assignment, which cannot be true. The data was deleted on the basis that it might have been corrupted.

The “work” done on data to have a higher geocoding percentage, is very simple. First all data that could  be geocoded using the fields “FRAGADENAV”, “FRAHUSNR” and “VFRAPOSTNR” was geocoded, those that did not match, were then geocoded using the “VFRGADENAV”, “VFRAHUSNR” and “VFRAPOSTNR” fields. Readers familiar with H. Søresen’s project [HS] will know that his geocoding percentage was lower, the second geocoding is probably the reason why, and it is not likely to have anything to do with the quality of data.

### *Road network*

The road network from Kraks Forlag A/S. The road network used has been cut out of the entire Danish road network. The part used contains about 52000 road segments. The attributes are as follows (Table 10):

| Field | Type | Description |
|---|---|---|
| SHAPE | Shape | The shape and position of the road. |
| LENGTH | Real | The length of the road. |
| TYP | Integer | The type of road. |
| VEJNAVN | String | The street name. |
| FROMLEFT | Integer | The first street number from left. |
| TOLEFT | Integer | The last street number from left. |
| FROMRIGHT | Integer | The first street number from right. |
| TORIGHT | Integer | The last street number from right. |
| FROMLEFT_B | String | The first street number, alphabetic part, from left. |
| TOLEFT_BOG | String | The last street number, alphabetic part, from left. |
| FROMRIGHT_ | String | The first street number, alphabetic part, from right. |
| TORIGHT_BO | String | The last street number, alphabetic part, from right. |
| V_SOGNENR | Integer | Parish number on the left side of the road. |
| H_SOGNENR | Integer | Parish number on the right side of the road. |
| V_POSTNR | Integer | Postal code on the left side of the road. |
| H_POSTNR | Integer | Postal code on the right side of the road. |
| KOMMUNENR | Integer | Municipality number for the road. |
| VEJKODE | Integer | Street code. |
| VEJKLASSE | Integer | Type of road. Highway, primary road etc. |
| SUBNET | Boolean | ? (Major transportation road network of DK) |
| RUTENR | String | Route number. |
| FRAKOERSEL | Integer | Exit turn on highway. |
| ZONE | Integer | City or non-city zone. |
| SPEED | Integer | Speed limit on road. |
| DRIVETIME | Real | The time it takes to pass the road. |
| ONE_WAY | String | One-way information. |
| VEJNR | Integer | Road number. |
| AENDR_DATO | Date | Last date for change. |
| TJEK_ID | Integer | Unique id number. |

**Table 10. Road network attributes.**

The accuracy of the road network is expected to lie within 5 meters, and the number of errors in the geometric and the attribute data are very few, none seemed to be a problem in this project.

# Appendix II Sum of Differences

When deciding which periods to compare, the method “sum of differences” from section 10.4.3: “Which to Compare” was used. The method revealed the following results:

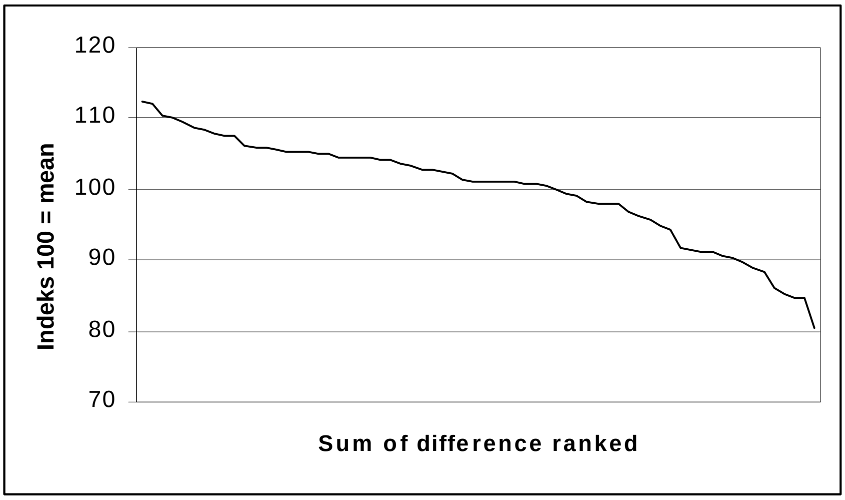



**Figure 66. The sum of difference ranked. Highest score to the left.**

It seems that there in general is quiet a difference between the periods. There are 60,000 accidents in each comparison (30.000 each set) and the lowest score was 12442 (20.7% of maximum 60,000), . The highest score was 17378 (30.0%). That it is only two pairs of datasets, which are compared in this project, does not mean that there could not be other dataset, which would also show an interesting difference. Figure 66 gives a clear indication that there are rather large variation over the period chosen in chapter 10: “Accident Analysis”.

# Appendix III How it works – Software Engineering

*This appendix deals with the technical aspects of how the GIS software, ArcView 3.2a, is expanded to handle the proposed models and how data is imported to and exported from a database system.*

First a short introduction to the software used.

## *ESRI's ArcView 3.2a*

ArcView 3.2a's role in this project is to display, manipulate and "interpret" data. ArcView 3.2a is a Geographic Information System designed for workstations, developed by ESRI (Environmental Systems Research Institute). It has a point and click interface using the mouse as input device. There is no command line in ArcView 3.2a, but it has a scripting language called Avenue.

Avenue is an object orientated scripting language. As such the source code is compiled concurrently when running a program. Development in Avenue is typically done by dividing smaller parts of functionality into scripts, which can call each other, much like functions calling other functions in other programming languages. Typical for scripting languages Avenue is easy to use/program but slow when it comes to calculations. Another problem with Avenue is that as such it is not possible to create data structures, it is necessary to use the included containers such as list and dictionaries, which makes Avenue very slow.

Using Avenue it is possible to make extensions that expand the functionality of ArcView 3.2a, such as Networks Analyst, which, among other things, makes it possible to calculate the shortest path in a network, and Spatial Analyst, of which the grid features are used. Both extensions are developed by ESRI. ArcView 3.2a is supplied with some extensions such as a Database extension that enables the user to communicate with Relational Database Management Systems (RDBMS) such as MS SQL * Server and Oracle using ODBC (Object Data Base Connection).

ArcView 3.2a is not the latest generation of GIS workstation software from ESRI, the latest generation is called ArcGIS 8. This previous generation is used primarily because it has not been possible to gain any experience with ArcGIS 8 opposed to ArcView 3.2a and because the limitations and flaws of ArcView 3.2a are well known, hence minimizing the risk of unexpected problems.

## *Microsoft's Access, Visual Studio 6.0 (C++) and Excel*

### MS Access

The RDBMS used in this project, MS ACCESS 2000 is not a real RDBMS and if anything else, such as MS SQL * Server or Oracle, is available it is recommend to use that instead. MS Access 2000 will crash when "stressed" or when being loaded with hundreds MB of data. The reason for using Access anyhow is that it was the only RDBMS with a reliable ODBC driver available to this project. An alternative could have been MySQL but the ODBC driver would not work, and MySQL does not come with and easy to use Graphical User Interface (GUI).

The role of the RDBMS in this project is to collect and store the results from the testing of the solution methods. Together with Excel Access provides a relatively easy interface to summarizing the test results using the querying language SQL (Microsoft Jet SQL, not the SQL92 standard).

During the project a strange thing was observed when using the standard deviation function in Access, *stdev()* . When calculating the standard deviation on 25 numbers with the same value, a standard deviation of about 13 is returned, in general the function seemed to have some numerical problems with low or non-existing deviation.

### MS Visual Studio 6.0 (C++)

Visual Studios C++ module is used to develop Dynamic Link Libraries (DLLs), which contain the data structures for the models and the algorithms used to solve the models. C++ is an object orientated programming language, which compared with Avenue is more demanding to learn and master, but the advantage is that the level of operational freedom is very high and most important of all it is extremely fast compared with Avenue when dealing with data structures not included in Avenue.

### MS Excel

Excel is used for data analysis and creating charts. Excel can relatively easily be integrated with an MS Access database, however the easy-to-use tools in Excel turned out not as flexible and not as integrated with Access as could have been desired. Primarily queries were carried out in MS Access and copied to Excel for further analysis.

### *Software Integration*

When software development reaches a certain size, as for this project, it is essential to create some kind of overall structure. One way to develop an overall structure is to divide the total project into a number of smaller projects/modules, each with its on purpose and interface. Figure 67 shows the organization of the modules, or at least how it is was designed to work. Modules in ArcView are groups of scripts with similar names such as DataBase.WriteSolution and Database. Initialize rather than extensions. Modules in C++ are dlls and the RDBMS modules are one Access file for each database.

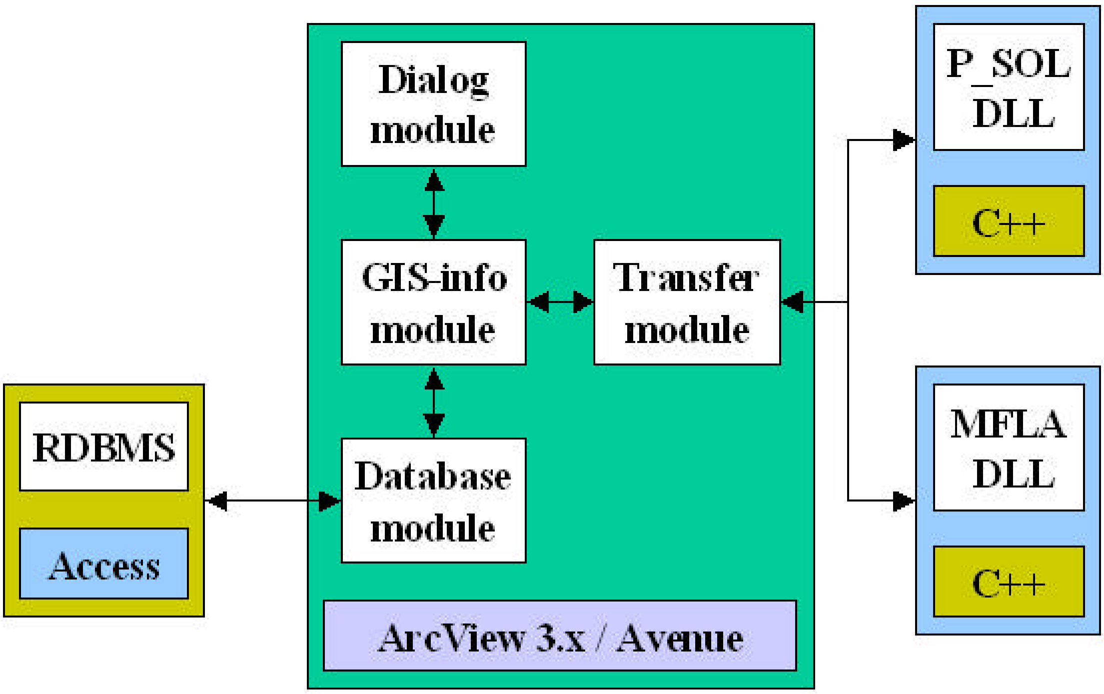



**Figure 67. Overall structure and modules in this project (this is an idealized drawing). Arrows show the communication paths**

The original idea was to create a user-friendly interface, using dialogs, however the development never got that far for all tasks. Instead scripts in Avenue are used as interface. The implementation used has a very clear grouping of the assignments: The dlls takes care of the optimization, the RDBMS takes care of storing the data and the GIS takes care of the communication, data handling and visualization of the results. Though as mentioned in section 11.1: “Continuous Model”, it might be an advantage to incorporate the GIS in order to verify the feasibility of solutions and perhaps to suggest possible changes in a solutions in order to improve it.

## *The programming*

Quite a lot of programming has been carried out in this project. The programming in ArcView had been divided into several smaller projects: one dealing with the tessellation of data, and other types of manipulation which required programming, another for testing the different models and solutions methods, also other minor project was created to generate maps, though most of the maps has been created manually. Two dll's has been created suing C++, one for the MFLA model and a second for the p-median and center models and a third for the "Try All Algorithm".

As demonstrated in "Software Integration", there is an overall design of how it all works. A design which for the most part has been complied with, though some of the code has been "bent", so that works and no more than that. In general the source code in this project is made to work and not to be undergoing an investigation of "pretty" programming and is not designed to be used by others, hence the documentation both in the project and in the code itself is very slim.

### The ArcView projects

*This section contains a small description/documentation, on what the two major ArcView projects do.*

#### Data manipulation

The data manipulation is the best structured ArcView project, it has been developed so far that it has a GUI (Graphical User Interface):

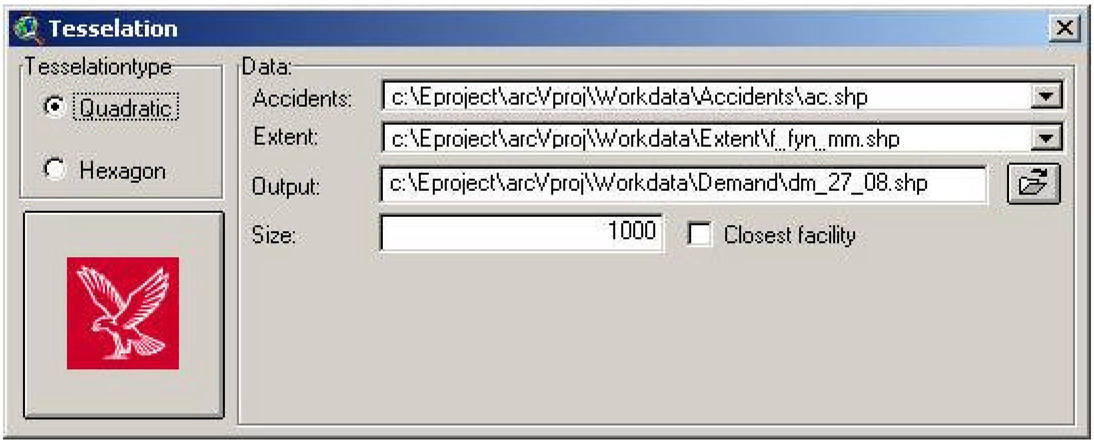



**Figure 68. Dialog used for tessellations.**

As can be seen in Figure 68, the project is able handle quadratic and hexagonal tessellation, either by Euclidean distance or to "Closest facility (demand point)" using travel time on the road network. Any distance can be used as tessellation, on any area (Extent) with any set of accidents. It all happens by pressing the button with the Falck symbol in the lower left corner.

The project contains more than over 30 scripts, each with their own job to do. Totaling about 2200 lines of code, including headers for each script, about 20 lines each.

**Solution Methods - ArcView**

The ArcView project for the Solution methods has five jobs:

- Create a model.
    - MFLA
    - p-median
    - p-center
    - parameters
    - etc.
- Load data into the dll.
- Activate the solution method.
- Retrieve data from the dll (the solution).
- Export data to the database system.

The programming of the “solution method”-project was not all that difficult, most of what is does it moving data around between different programs (DLLs, ArcView and RDBMS). Nevertheless it still contains more than 30 scripts, about 3200 lines of code, including headers for each script, about 20 lines each.

## The dlls

C++ code in the two dlls do have some similarities, though they are very different, since the discrete model rely on a cost/response time found by ArcView (stored in a file) and the continuous model rely on the Euclidean distance hence it needs the coordinates of the demand points. Both dll’s contain the following features:

- An object orient approach to modeling the relation ship between a customer (demand point) and a facility (ambulance/ambulance point). Such that a customer can be asked, which facility it belongs to in one of the four implemented solutions (best, current, temporary and working solution) (only the discrete dll.).
- Method for importing and exporting data. Including information on allocation of customers, position of facilities, solution history (stored values from each iteration of best and current solution value, only the discrete dll) and other information such as the travel time between customers and facilities.
- Relatively flexible interface to solution methods, with a majority of the solution method parameters as arguments to the solution methods.

The programming of the dlls was relative difficult, not only because C++ is a difficult language to program, but also because the debugging/testing of the dlls was difficult since the testing was carried out through ArcView scripts. Most of the time when the dll had an error, ArcView either crashed or returned “Segmentation violation” with no further information available on where and why it went wrong.

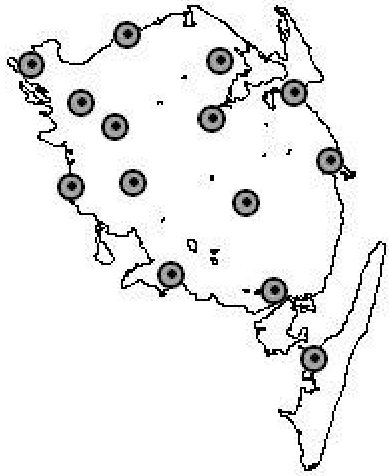
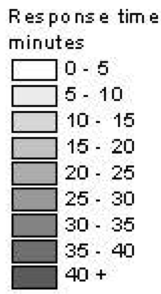
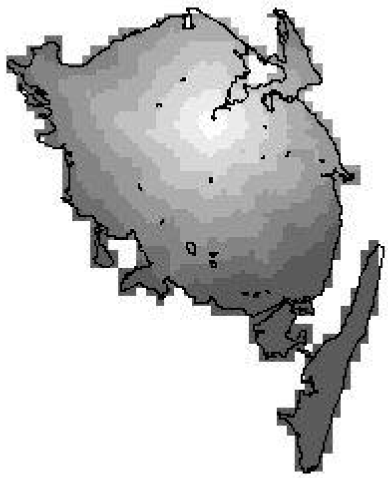
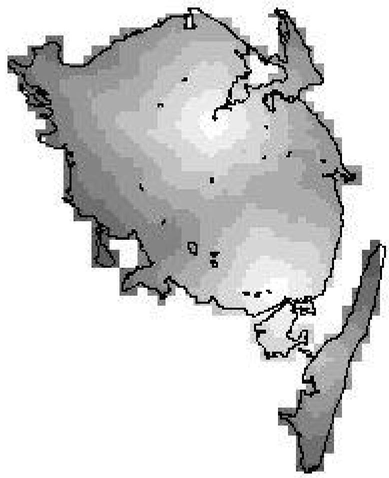
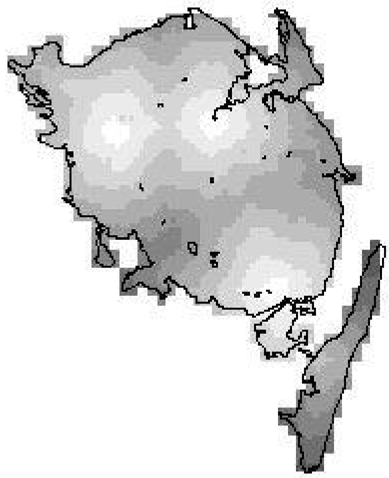
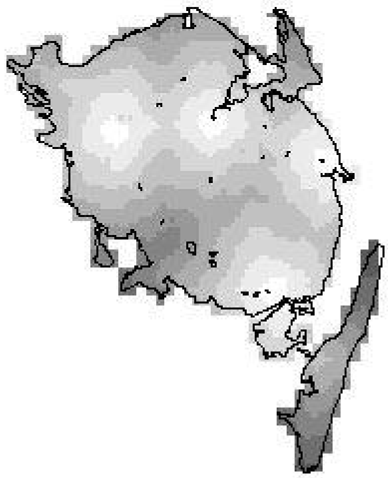
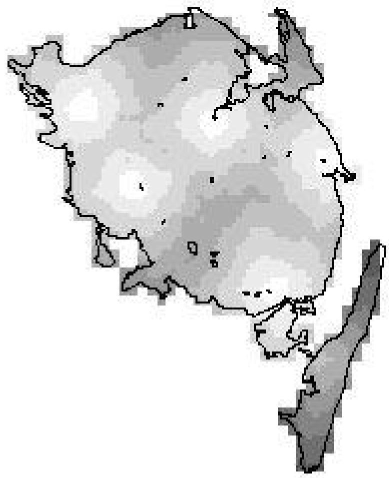
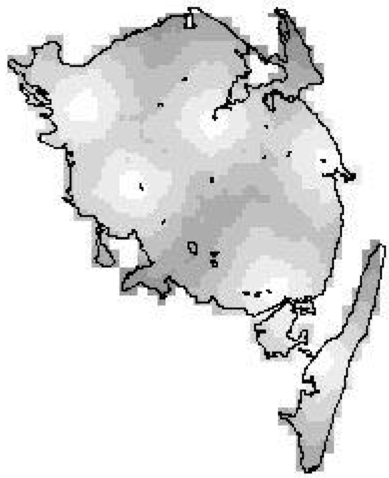
The source code is guessed to be between 2000 and 3000 lines code, with relatively few headers, comments etc. The data structure used for storing the costs for the Noising method is taken from the book: “Data Structures, Algorithms and Applications in C++” by S. Sahni [DSA].

# Appendix IV Best Solutions

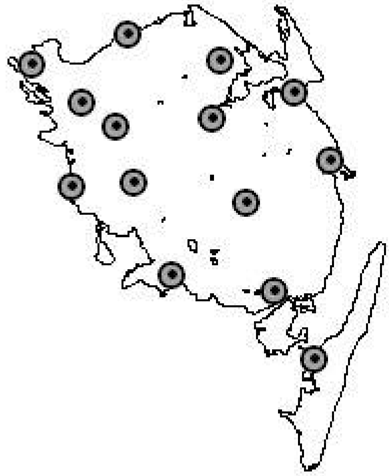
The solutions presented here are the best solutions found during the project. There are basically two ways to represent a solution, either by resenting the ambulance points / positions where the ambulances are placed or by presenting the response time for the demand points. The solutions will be presented using the response time since, this gives a better impression of the solution than just *p* points on a map. However placing 1053 points on a map is not a very good way of visualizing the response time, since some points will be covering other points. Alternatively a grid has been interpolated using the ArcView's Spatial Analyst surface function Inverse Weighted Distance (IDW) [SAP p. 92]. The idea of IDW is points distant from the a cell has less influence on the value of the cell than those close. The interpolation is calculated as mean of the 12 nearest neighbors (how those are defined is not known), with a grid cell size of one by one km. It has been checked that the grids are a fair representation of solutions.

In some of the solutions it might be difficult to find all the ambulances, however a close look in the area northeast/east of Odense (Munkebo/Kerteminde), will probably reveal the missing one.

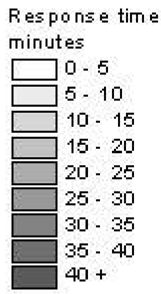
# Best Results - Falck’s Garages - p-median – NM

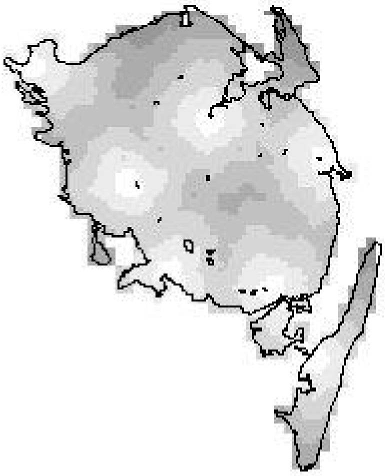
<table>
<tr>
<td>Best solutions for allocating ambulances to Falck garages, when using the p-median model. “Spring at work” dataset used as demand.</td>
<td><br>Falck’s 14 garages on Funen</td>
<td><br></td>
</tr>
<tr>
<td><br>One ambulance</td>
<td><br>Two ambulances</td>
<td><br>Three ambulances</td>
</tr>
<tr>
<td><br>Four ambulances</td>
<td><br>Five ambulances</td>
<td><br>Six Ambulances</td>
</tr>
</table>

Best solutions for allocating ambulances to Falck garages, when using the p-median model. "Spring at work" dataset used as demand.

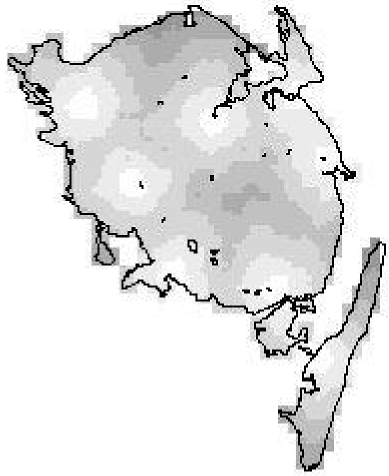


Falck's 14 garages on Funen

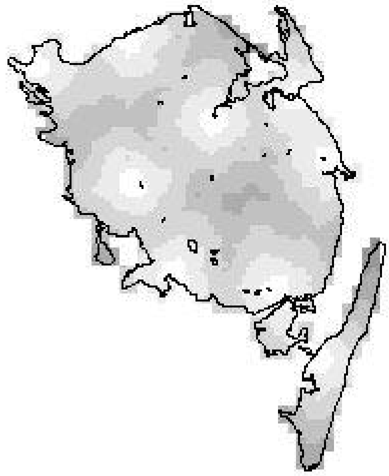


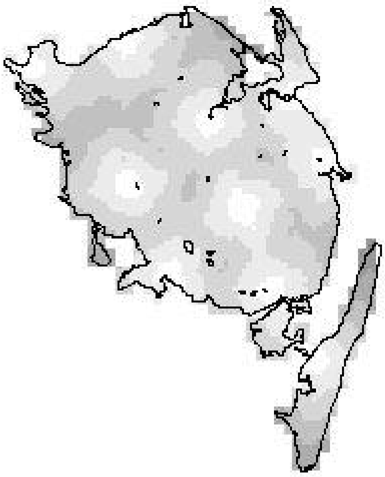
Seven ambulances



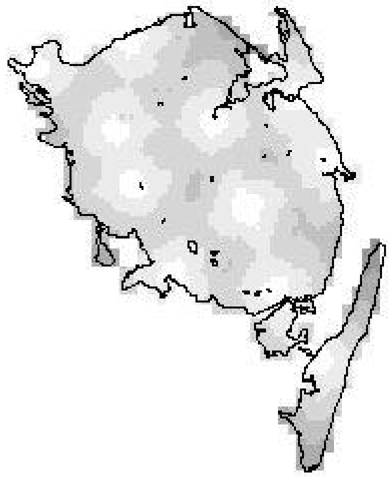
Eight ambulances



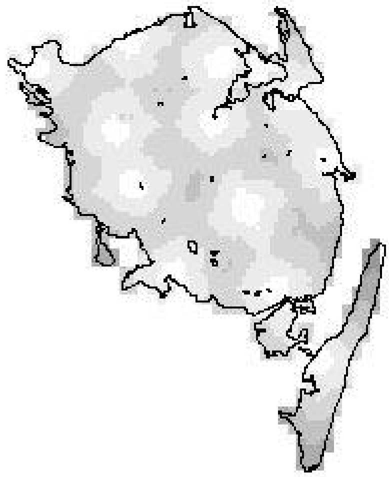
Nine ambulances



Ten ambulances



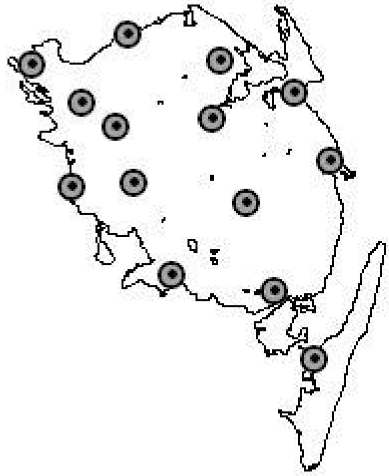
Eleven ambulances



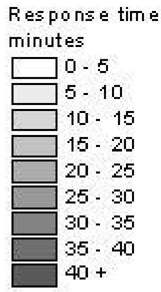
Twelve Ambulances

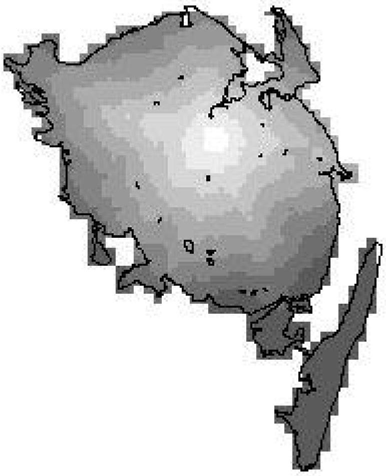
# Best Results - p-median - NM

Best solutions for allocating ambulances to the demand points, when using the p-median model. “Spring at work” dataset used as demand.

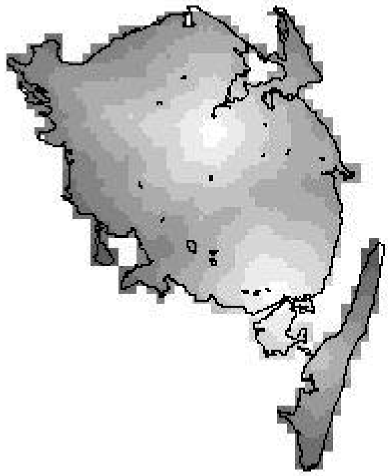


Falck’s 14 garages on Funen

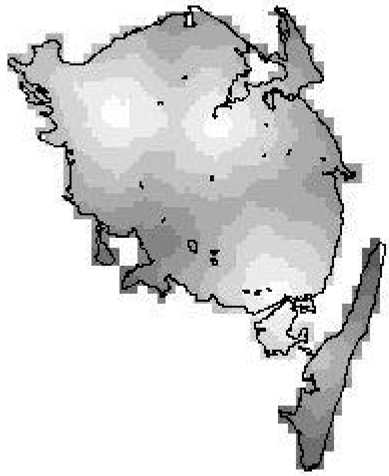


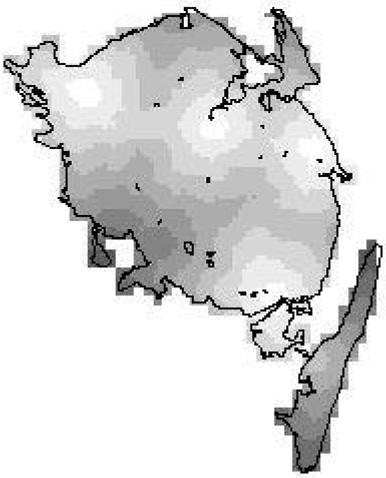
One ambulance



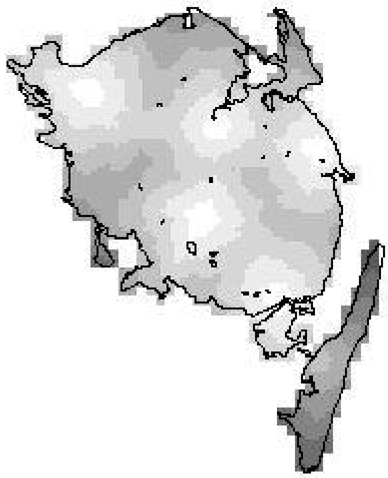
Two ambulances



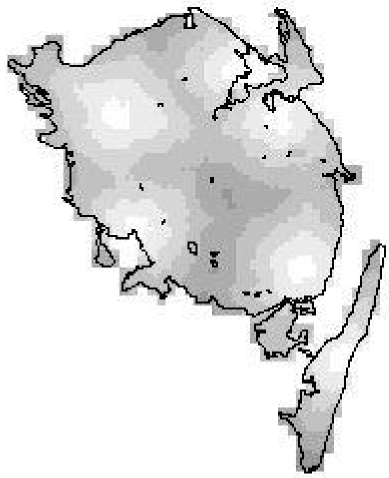
Three ambulances



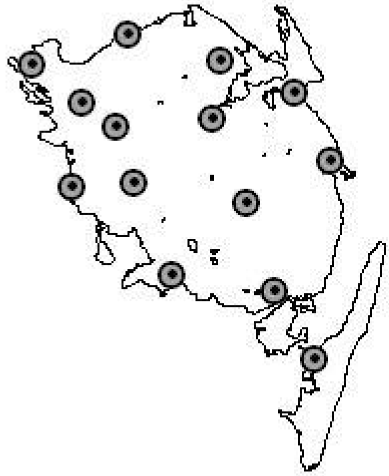
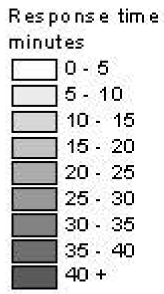
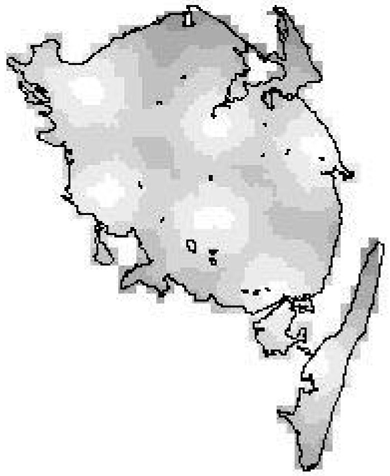
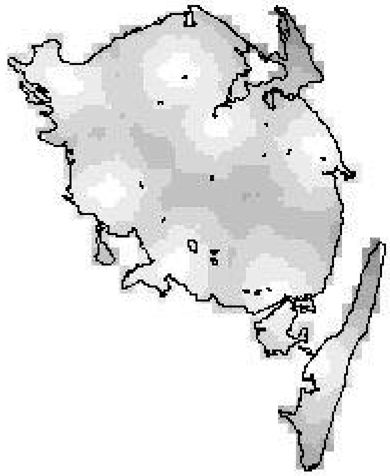
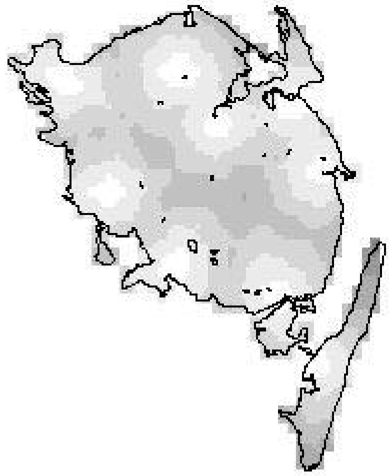
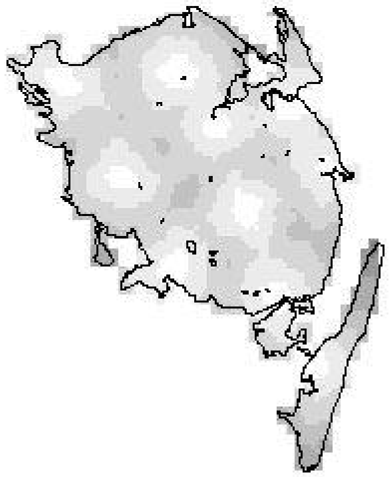
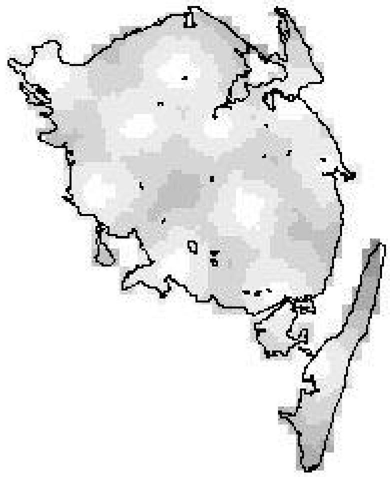
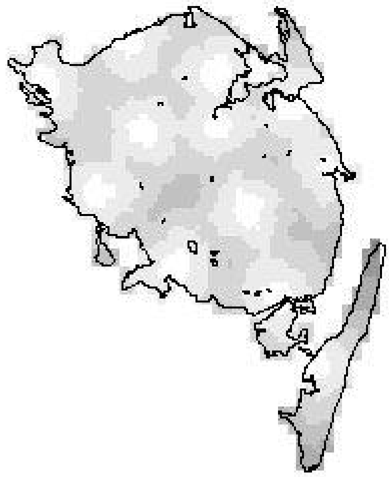
Four ambulances



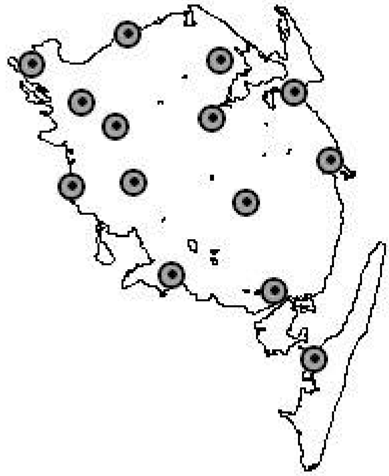
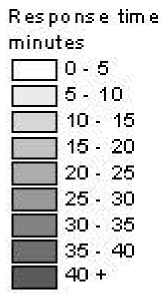
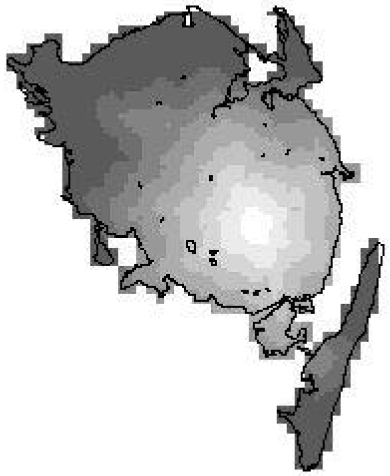
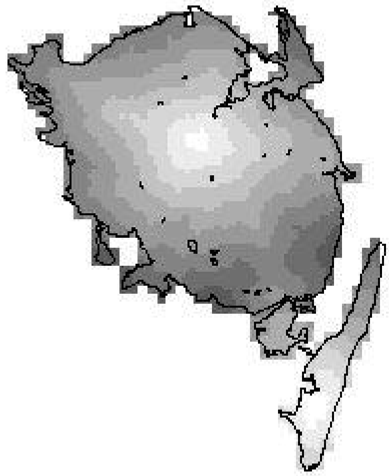
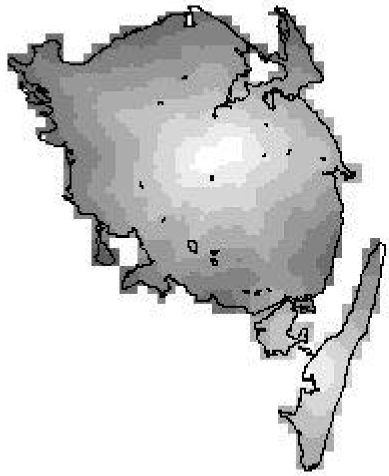
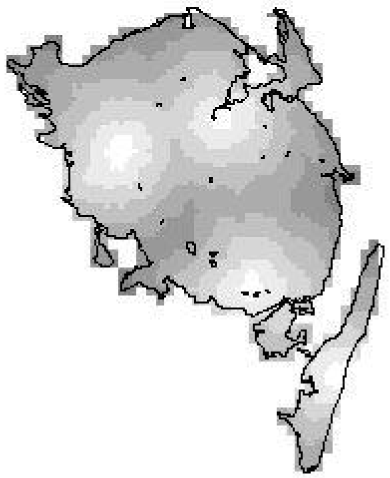
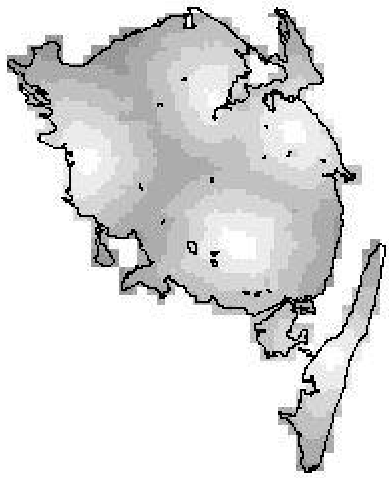
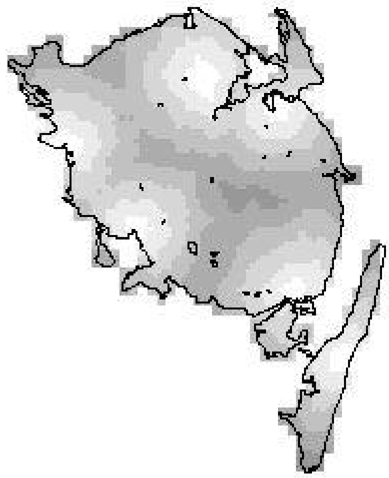
Five ambulances

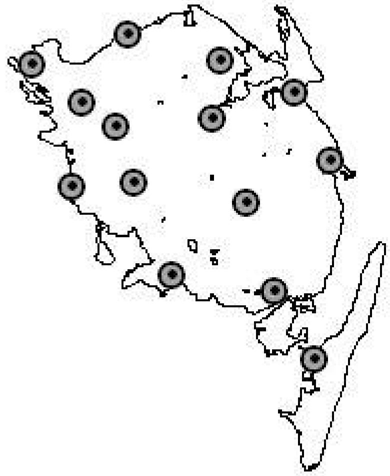
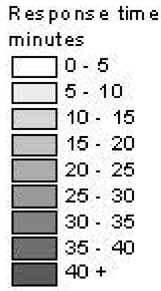
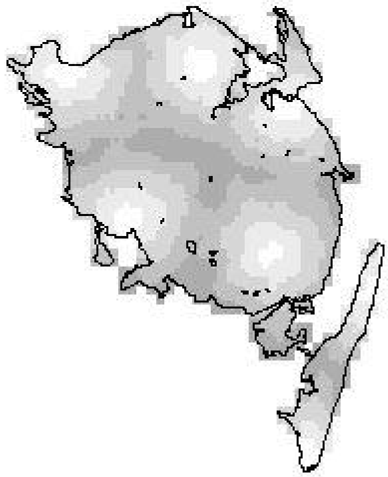
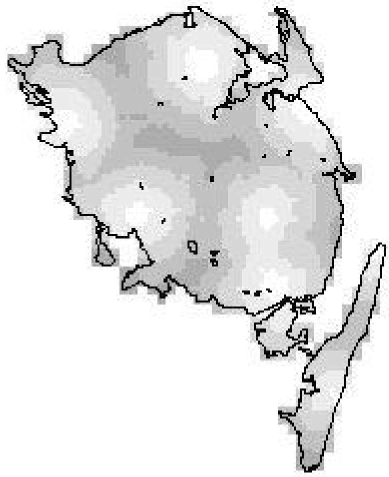
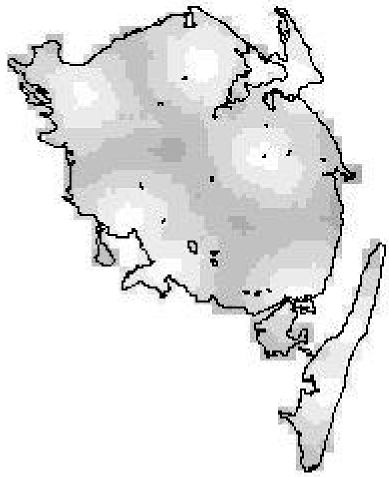
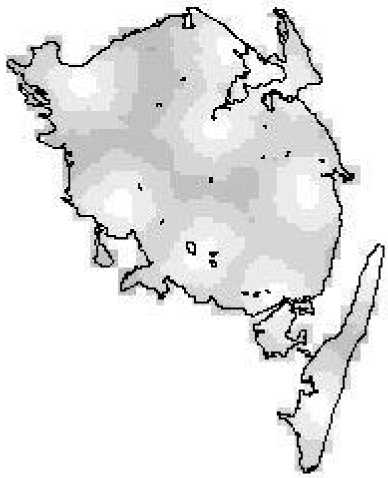
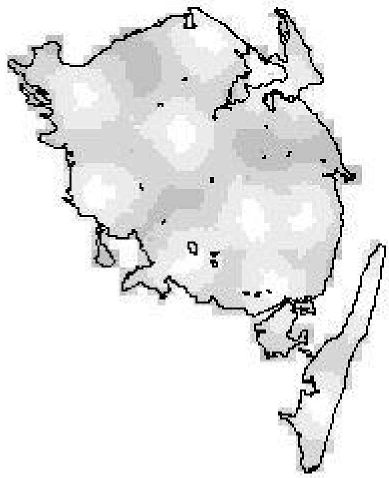
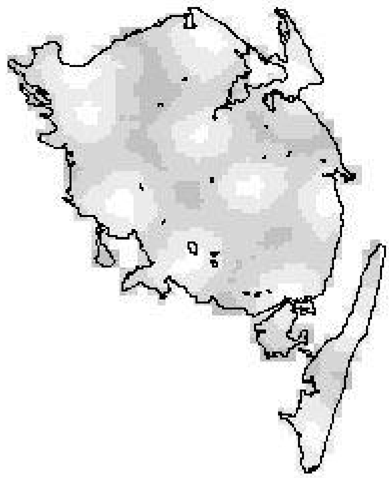


Six ambulances

| | | |
|---|---|---|
| Best solutions for allocating ambulances to the demand points, when using the p-median model. "Spring at work" dataset used as demand. |  Falck's 14 garages on Funen |   |
|  Seven ambulances |  Eight ambulances |  Nine ambulances |
|  Ten ambulances |  Eleven ambulances |  Twelve ambulances |

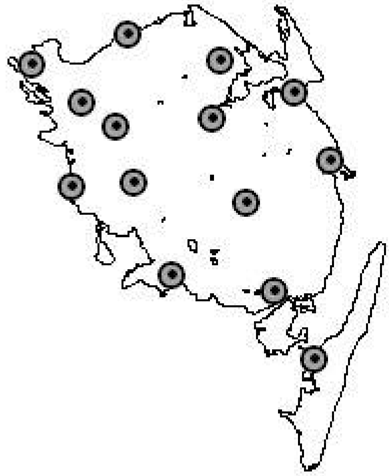
# Best Results - p- center - NM

| | | |
|---|---|---|
| Best solutions for allocating ambulances to the demand points, when using the p-center model. “Spring at work” dataset used as demand.<br><br>For the solution with three ambulances there are two ambulances in Odense, though it is difficult to see. | <br>Falck’s 14 garages on Funen | <br> |
| <br>One ambulance | <br>Two ambulances | <br>Three ambulances |
| <br>Four ambulances | <br>Five ambulances | <br>Six ambulances |

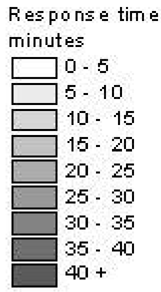
| | | |
|---|---|---|
| Best solutions for allocating ambulances to the demand points, when using the p-median model. “Spring at work” dataset used as demand. |  Falck’s 14 garages on Funen |   |
|  Seven ambulances |  Eight ambulances |  Nine ambulances |
|  Ten ambulances |  Eleven ambulances |  Twelve ambulances |

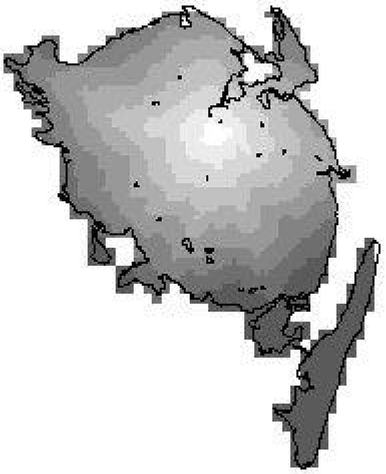
# Best Results - MFLA - MRC

Best solutions for allocating ambulances to the demand points, when using the MFLA model. “Spring at work” dataset used as demand.

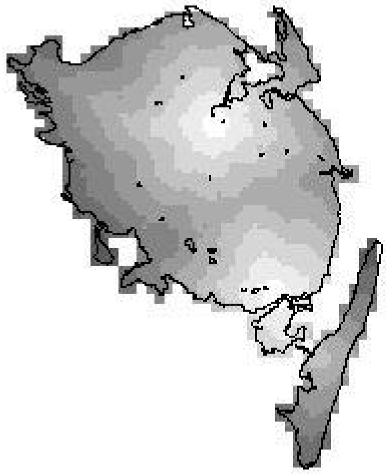


Falck’s 14 garages on Funen

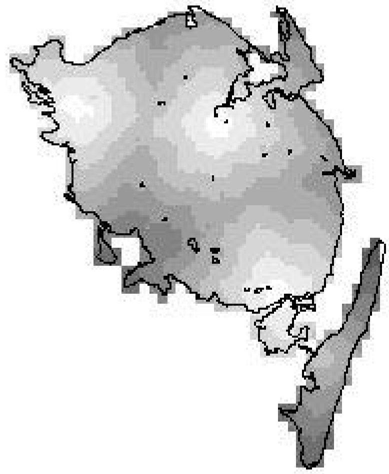


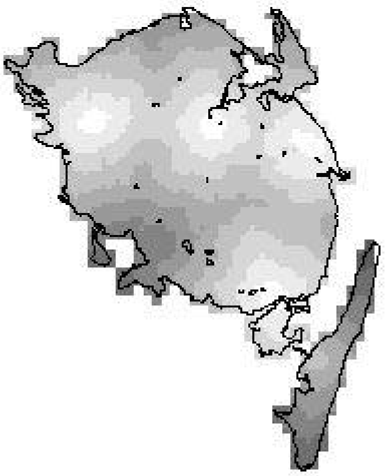
One ambulance



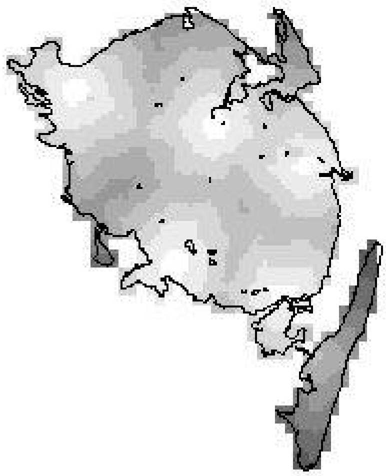
Two ambulances



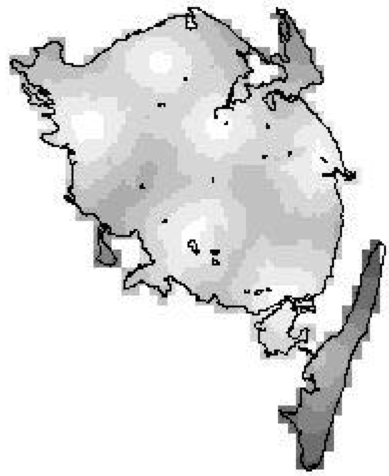
Three ambulances



Four ambulances

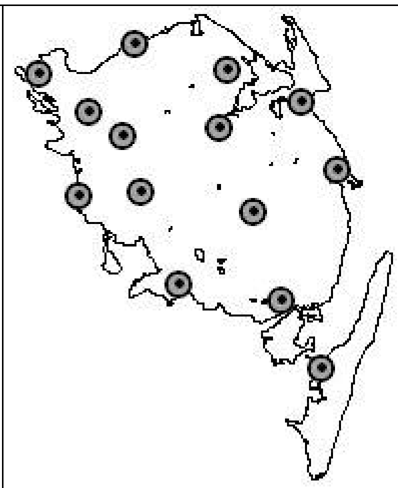


Five ambulances

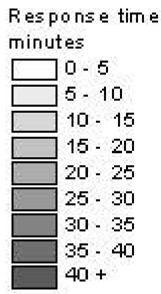


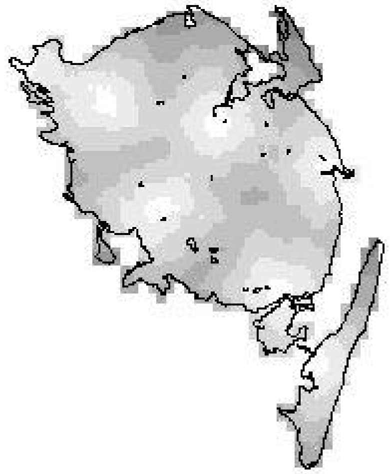
Six ambulances

Best solutions for allocating ambulances to the demand points, when using the MFLA model. “Spring at work” dataset used as demand.

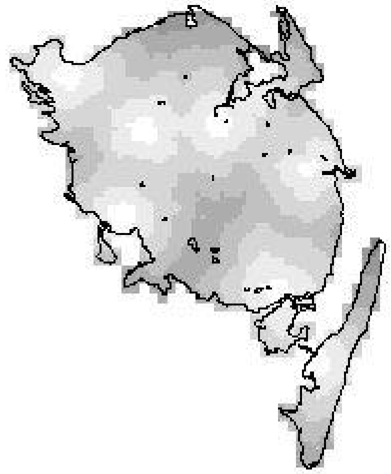


Falck’s 14 garages on Funen

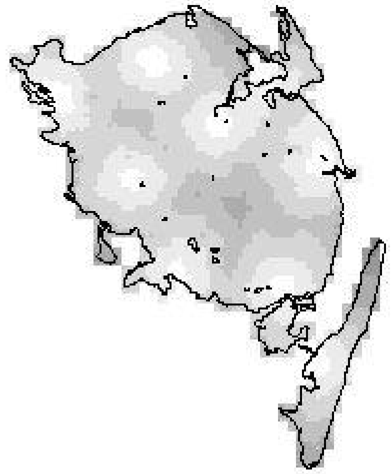


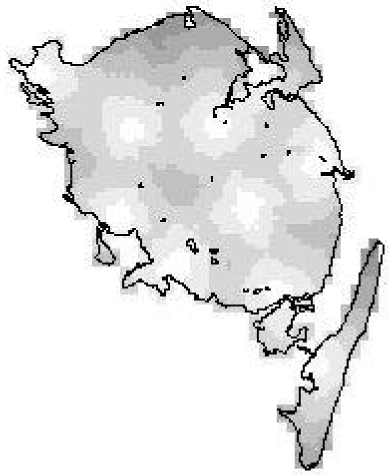
Seven ambulances



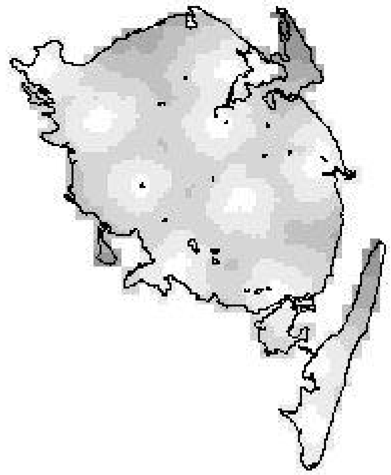
Eight ambulances



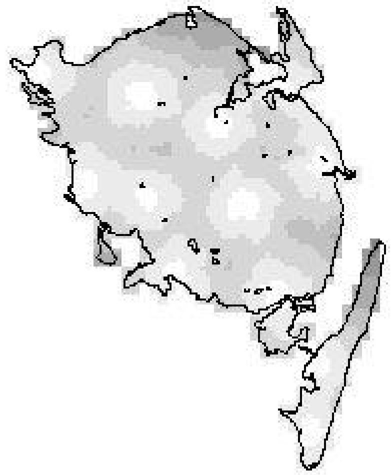
Nine ambulances



Ten ambulances



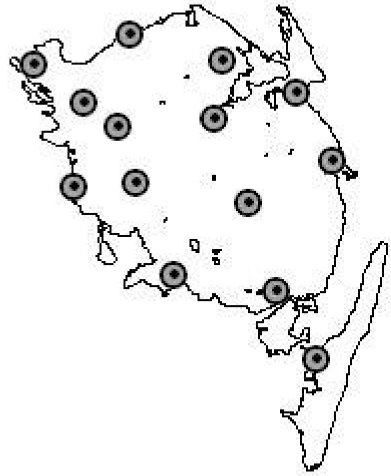
Eleven ambulances



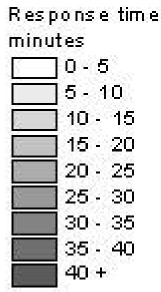
Twelve ambulances

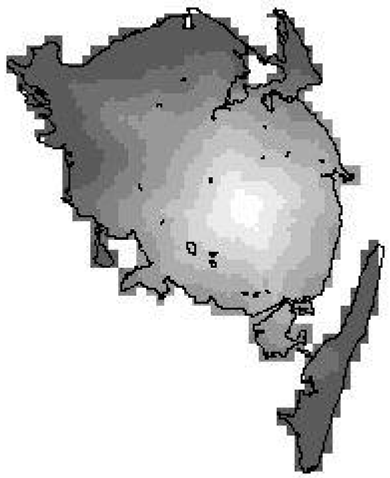
## Best Results - Falck’s Garages - p-center – NM

Best solutions for allocating ambulances to Falck garages, when using the p-center model. “Spring at work” dataset used as demand.

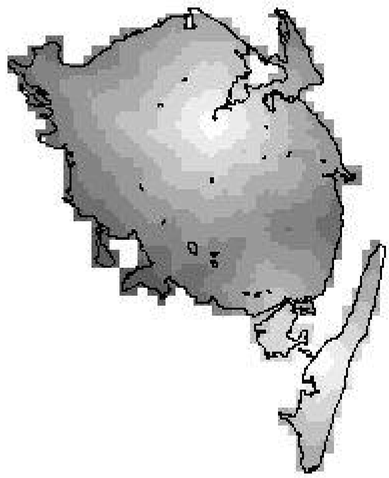


Falck’s 14 garages on Funen

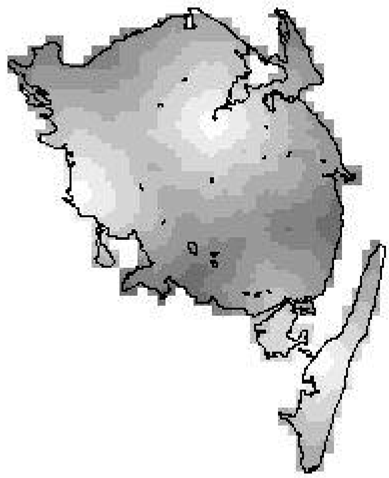


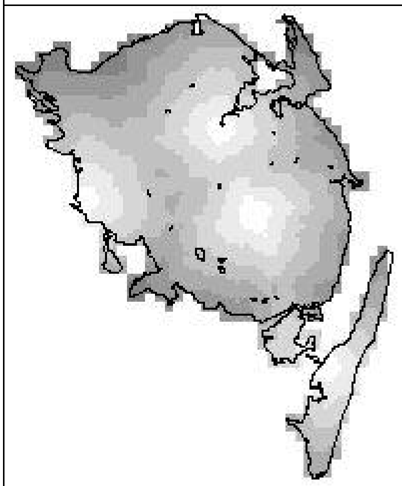
One ambulance



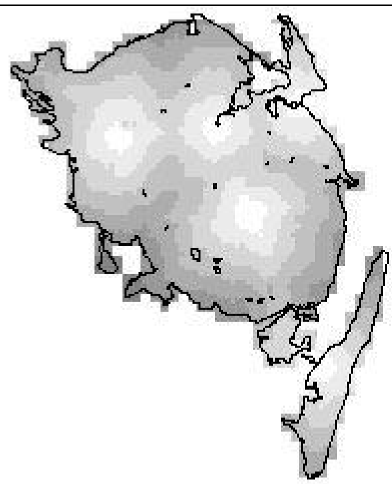
Two ambulances



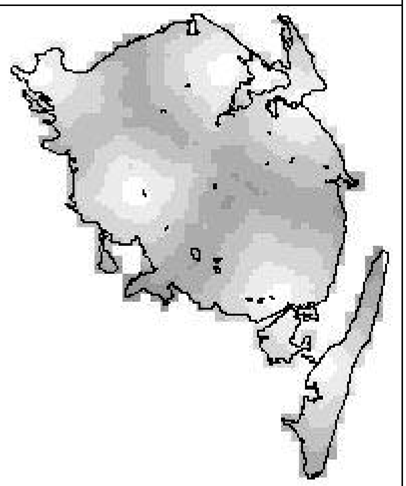
Three ambulances



Four ambulances



Five ambulances



Six Ambulances