Robustness in facility location

Author: Sander W.M. van Lokven
MSc Research report
Universidad de Málaga in co-operation with Wageningen University
and Universidad de Almería
Dpt. Computer Architecture, Operations Research and Logistics
and Computer Architecture and Electronics

Supervisors: Eligius M.T. Hendrix and Pilar M. Ortigosa
Examiner: Juana L. Redondo

March 9, 2009

Preface

First of all, I want to thank Eligius Hendrix for mentoring my thesis. For all problems and questions, he had an answer.
I want to thank both my supervisors for all the constructive meetings I had with them. Eligius Hendrix and Pilar Ortigosa helped me a lot with their knowledge and experience, while stimulating me to get the best out of myself, which finally resulted in this report.
Juana Redondo contributed by critically reviewing my thesis, which helped me restructuring and improving the final report.

Also, I want to thank the University of Almeria for facilitating my thesis. Special thanks go to the department of Computer Architecture and Electronics for providing me with all necessary equipment and helpful colleagues. Especially the head of the department, Inmaculada Garcia ensured that I had all what I need.
I also like to thank the Wageningen University, for granting me permission to perform my thesis in Almeria.

During just 6 months time, I came across many learning opportunities. First there was the international working group conference on locational analysis in Elche (EWGLA) where there was the opportunity to discuss and learn from many experts in facility location science. This conference gave a good overview of the variety of interesting subjects in facility location which was a real inspiration for me and my report.

After this conference, I followed a course on global optimization at the University of Almeria. This course provided me with a broad view on all sorts of optimization techniques which can be used to solve facility location models. Leocadio Casado gave an in-depth inside into the world of branch and bound algorithms and parallel computing. In between he also taught me the basics of programming in C++.
Pilar Ortigosa showed the opportunities of meta-heurstics for very large and hard to solve problems. Her expertise on the stochastic algorithms was very useful for the algorithm settings in this report. Eligius Hendrix provided the necessary theoretical backgrounds.

Finally, I want to thank Emilio Carrizosa, an expert in facility location science. Three times I had a meeting with him to discuss robustness in facility location science. He provided me with insight in the deviation robustness concept and helped me with developing the new model. After the new model was developed, I presented him the new model at the University of Sevilla. We discussed its properties which helped me by finalising my conclusions and sharpening my discussions.

Contents

Abstract ... 8

1. Introduction ... 10
  1.1 Introduction to the problem and problem definition ... 10
  1.2 Research objective(s) and research questions ... 10
  1.3 Mission and vision ... 10

2. Facility location ... 11
  2.1 Introduction ... 11
  2.2 The feasible area ... 11
  2.3 One or more new facilities ... 12
  2.4 Demand points and distances ... 12
  2.5 Attraction of demand points ... 12
  2.6 Competition ... 12
    2.6.1 Static competition ... 12
    2.6.2 Dynamic competition ... 13
  2.7 Time horizon ... 14
  2.8 Objectives ... 14
    2.8.1 The p-median problem ... 15
    2.8.2 The p-centre problem ... 16
    2.8.3 The uncapacitated facility location problem ... 17
    2.8.4 The quadratic assignment problem ... 18
  2.9 Extra constraints ... 19

3. Robustness in facility location ... 20
  3.1 A standard single facility location model ... 21
  3.2 A small numerical example ... 22
  3.3 Robustness concepts ... 23
    3.3.1 The Yes or No performance robustness concept ... 23
    3.3.2 The Probabilistic robustness concept ... 24
    3.3.3 The Deviation robustness concept ... 24
      3.3.3.1 The 1-norm ... 26
      3.3.3.2 The 2-norm ... 26
      3.3.3.3 The infinite-norm ... 27
    3.3.4 The Safety First robustness concept ... 28
    3.3.5 The Maximum Regret robustness concept ... 29

4. Studied Robustness Models in Literature ... 31
  4.1 Yes or No Performance Robustness models ... 31
    4.1.1 The standard Yes or No Performance robustness model ... 31
  4.2 Probabilistic concept Robustness models ... 32
    4.2.1 The standard Probabilistic robustness model ... 32
    4.2.2 A threshold satisfying competitive location model (2002) ... 32
    4.2.3 Probabilistic location problems with discrete demand weights (2004) ... 35
    4.2.4 Probabilistic 1-maximum covering problem with discrete demand weights (2008) ... 36
    4.2.5 The 1-center problem in the plane with independent random weights (2008) ... 37
  4.3 Deviation concept Robustness models ... 39
    4.3.1 The standard Deviation robustness model ... 39
    4.3.2 Robust facility location (2003) ... 39

4.4 Safety first concept Robustness models 40
4.4.1 The standard Safety first robustness model 40
4.4.2 The p-median problem in a changing network (1998) 40
4.4.3 Conditional median as a robust solution concept (2009) 41
4.5 Minimize maximum regret Robustness models 42
4.5.1 The standard Maximum Regret robustness model 42
4.5.2 Minmax regret p-centre location on a network with demand uncertainty (1998) 42
4.5.3 Facility location problems with uncertainty on the plane (2005) 42
4.5.4 The multi-criteria minisum location problem (2001) 43

5. A competitive deviation robustness model 44
5.1 Elements of the new model 44
5.2 The competitive deviation robustness model 45

6. Algorithms for solving robust location problems 46
6.1 Controlled Random Search 46
6.2 Genetic Algorithm 47
6.3 Simulated Annealing 48
6.4 Multi Start 49

7. Test examples 50
7.1 The feasible area 50
7.2 Test Case 1: The p-centre problems 51
7.3 Test Case 2: The robust p-centre problems 51
7.4 Illustration Case: Robust 1-median problem with competition 53

8. Stochastic analysis at a robust location test case 55
8.1 Test criteria, effectiveness and efficiency 55
8.2 Algorithm settings and results for Test Cases 1 and 2 55
8.2.1 Controlled Random Search settings and results 55
8.2.2 Genetic Algorithm standard settings 57
8.2.3 Simulated Annealing standard settings 58
8.2.4 Multi Start standard settings 60
8.3 Conclusions algorithm testing 61

9. Illustrating the competitive deviation robustness model 62
9.1 Simulated Annealing illustration of the optima 62
9.1.1 Two competitor case 62
9.1.2 Three competitor case 63
9.1.3 Results of the Simulated Annealing illustrations 63
9.2 Grid Search illustration of the objective function 64
9.2.1 Illustration of the objective function for 2 competitors 65
9.2.2 Illustration of the objective function for 3 competitors 66

10. Conclusion, discussion and further research 67
10.1 Conclusion 67
10.1.1 Facility location 67
10.1.2 Robustness Concepts 67
10.1.3 Algorithm testing 67
10.1.4 The competitive deviation robustness model 67
10.2 Discussion 68
10.3 Further research 68

References 69

Appendix A: Test Case Data 72
Appendix A.1: Data Demand Points Test Case 2 and 3 72
Appendix A.2: Data weight vector test case 2 72
Appendix A.3: Data weight matrix test case 3 72

Appendix B: Algorithm Matlab Files
Appendix B.1: General Matlab files of Case 1 and 2 .... cd-rom
Appendix B.1.1: The General 2-centre M-file .... cd-rom
Appendix B.1.2: The General 3-centre M-file .... cd-rom
Appendix B.1.3: The General robust 2-centre M-file.. cd-rom
Appendix B.1.4: The General robust 3-centre M-file.. cd-rom
Appendix B.2: The Grid Search Matlab files .... cd-rom
Appendix B.2.1: The GS robust 2-centre M-file .... cd-rom
Appendix B.2.2: The GS robust 3-centre M-file .... cd-rom
Appendix B.3: The Controlled Random Search Matlab Files .... cd-rom
Appendix B.3.1: The CRS 2-centre M-file.. cd-rom
Appendix B.3.2: The CRS 3-centre M-File .... cd-rom
Appendix B.3.3: The CRS robust 2-centre M-File... cd-rom
Appendix B.3.4: The CRS robust 3-centre M-File... cd-rom
Appendix B.4: The Genetic Algorithm Matlab Files... cd-rom
Appendix B.4.1: The GA 2-centre M-file .... cd-rom
Appendix B.4.2: The GA 3-centre M-file .... cd-rom
Appendix B.4.3: The GA robust 2-centre M-file .... cd-rom
Appendix B.4.4: The GA robust 3-centre M-file .... cd-rom
Appendix B.5: The Simulated Annealing Matlab files .... cd-rom
Appendix B.5.1: The SA 2-centre M-file .... cd-rom
Appendix B.5.2: The SA 3-centre M-file .... cd-rom
Appendix B.5.3: The SA robust 2-centre M-file .... cd-rom
Appendix B.5.4: The SA robust 3-centre M-file .... cd-rom
Appendix B.6: The Multi Start Matlab files .... cd-rom
Appendix B.6.1: The MS 2-centre M-file.. cd-rom
Appendix B.6.2: The MS 3-centre M-file.. cd-rom

Appendix C: Algorithm Test Results
Appendix C.1: Controlled Random Search test results .... cd-rom
Appendix C.1.1: CRS 2-centre problem test results .... cd-rom
Appendix C.1.2: CRS 3-centre problem test results .... cd-rom
Appendix C.1.3: CRS robust 2-centre problem test results .... cd-rom
Appendix C.1.4: CRS robust 3-centre problem test results .... cd-rom
Appendix C.2: Genetic Algorithm test results .... cd-rom
Appendix C.2.1: GA 2-centre problem test results.. cd-rom
Appendix C.2.2: GA 3-centre problem test results.. cd-rom
Appendix C.2.3: GA robust 2-centre problem test results....................................... cd-rom
Appendix C.2.4: GA robust 3-centre problem test results....................................... cd-rom
Appendix C.3: Simulated Annealing test results .... cd-rom
Appendix C.3.1: SA 2-centre problem test results .... cd-rom
Appendix C.3.2: SA 3-centre problem test results .... cd-rom
Appendix C.3.3: SA robust 2-centre problem test results .... cd-rom
Appendix C.3.4: SA robust 3-centre problem test results .... cd-rom
Appendix C.4: Multi Start test results.. cd-rom
Appendix C.4.1: MS 2-centre problem test results .... cd-rom
Appendix C.4.2: MS 3-centre problem test results .... cd-rom

Appendix D: Illustration Matlab Files
Appendix D.1: General Matlab files for the illustration case .... cd-rom
Appendix D.1.1: General M-file for 1-norm by 2 competitors .... cd-rom
Appendix D.1.2: General M-file for 2-norm by 2 competitors .... cd-rom
Appendix D.1.3: General M-file for ∞-norm by 2 competitors .... cd-rom
Appendix D.1.4: General M-file for 1-norm by 3 competitors .... cd-rom
Appendix D.1.5: General M-file for 2-norm by 3 competitors .... cd-rom
Appendix D.1.6: General M-file for ∞-norm by 3 competitors .... cd-rom
Appendix D.2: Simulated Annealing Matlab files for the illustration case .... cd-rom
Appendix D.2.1: SA M-file for 1-norm by 2 competitors .... cd-rom
Appendix D.2.2: SA M-file for 2-norm by 2 competitors .... cd-rom
Appendix D.2.3: SA M-file for ∞-norm by 2 competitors .... cd-rom
Appendix D.2.4: SA M-file for 1-norm by 3 competitors .... cd-rom
Appendix D.2.5: SA M-file for 2-norm by 3 competitors .... cd-rom
Appendix D.2.6: SA M-file for ∞-norm by 3 competitors .... cd-rom
Appendix D.3: Grid Search Matlab files for the illustration case .... cd-rom
Appendix D.3.1: GS for 1-norm by 2 competitors .... cd-rom
Appendix D.3.2: GS for 2-norm by 2 competitors .... cd-rom
Appendix D.3.3: GS for ∞-norm by 2 competitors .... cd-rom
Appendix D.3.4: GS for 1-norm by 3 competitors .... cd-rom
Appendix D.3.5: GS for 2-norm by 3 competitors .... cd-rom
Appendix D.3.6: GS for ∞-norm by 3 competitors .... cd-rom

Appendix E: Results for the illustration case
Appendix E.1: Simulated Annealing results for the 2 competitor case .... cd-rom
Appendix E.1.1: SA results for 1-norm by 2 competitors .... cd-rom
Appendix E.1.2: SA results for 2-norm by 2 competitors .... cd-rom
Appendix E.1.3: SA results for ∞-norm by 2 competitors .... cd-rom
Appendix E.2: Simulated Annealing results for the 3 competitor case .... cd-rom
Appendix E.2.1: SA results for 1-norm by 3 competitors .... cd-rom
Appendix E.2.2: SA results for 2-norm by 3 competitors .... cd-rom
Appendix E.2.3: SA results for ∞-norm by 3 comp .... cd-rom

Abstract

Facility location concerns the placement of facilities, for various objectives, by use of mathematical models and solution procedures. Almost all facility location models that can be found in literature are based on minimizing costs or maximizing cover, to cover as much demand as possible. These models are quite efficient for finding an optimal location for a new facility for a particular data set, which is considered to be constant and known in advance.
In a real world situation, input data like demand and travelling costs are not fixed, nor known in advance. This uncertainty and uncontrollability can lead to unacceptable losses or even bankruptcy. A way of dealing with these factors is robustness modelling. A robust facility location model aims to locate a facility that stays within predefined limits for all expectable circumstances as good as possible.

From literature search, five concepts of robustness are found. These are:

1. The Yes or No performance robustness concept,
2. The Probabilistic robustness concept,
3. The Deviation robustness concept,
4. The Safety First robustness concept,
5. The Maximum Regret robustness concept

The deviation robustness concept is the most interesting concept as it suits uncertainty and uncontrollability of data the best. The deviation concept needs only a nominal prediction of the data instead of the whole range or probability distribution, which is needed for the four other robustness concepts.
The deviation robustness concept is used as basis to develop a new competitive deviation robustness model. The competition is modelled with a Huff based model, which calculates the market share of the new facility. Robustness in this model is defined as the ability of a facility location to capture a minimum market share, despite variations in demand.

A test case is developed by which algorithms can be tested on their ability to solve robust facility location models. Four stochastic optimization algorithms are considered from which Simulated Annealing turned out to be the most appropriate.
The test case is slightly modified for a competitive market situation. With the Simulated Annealing algorithm, the developed competitive deviation model is solved, for three considered norms of deviation.
At the end, also a grid search is performed to illustrate the landscape of the objective function of the competitive deviation model. The model appears to be multimodal and seems to be challenging for further research.

1. Introduction

1.1 Introduction to the problem and problem definition

Almost all models that can be found in literature on facility location are based on minimizing costs (to optimise profit) or maximizing cover (to cover as much demand as possible). These models are quite efficient for finding an optimal location for a facility for a particular data set. However, these models lack the ability to see how robust the optimal facility location is. What happens if the data set is changing, for instance due to increasing transport costs because of the rising oil prices, or due to a decrease in demand caused by an increase of competition. Would the 'optimal' facility location under changing conditions still be the optimal one?
When the facility is placed, the facility location cannot be adapted to changing circumstances on short term, as a facility location is a long term strategic decision. Therefore it is interesting to search for a facility location that has low costs or is making profit under all expectable circumstances.

A robust facility location model is a model that deals with uncertainties of uncontrollable factors, such as demand and transportation costs. A robust facility location model tries to optimize the robustness of the outcome by finding a location that stays within predefined limits for all expectable circumstances as good as possible. In literature, the predefined limits that are most often used are a threshold (also called budget) for maximum allowed costs or minimum foreseen profit (Vlajic et al., 2008, Olieman, 2008 and Hendrix, 2008).
This research focuses on the 1-facility location problem, with the question how to locate a facility as robust as possible.

1.2 Research objective(s) and research questions

The research objective of this thesis is to investigate continuous facility location models that can be used to optimize the facility location on the basis of lowest cost or maximum cover and location robustness.

1) What is facility location science?
    a. What are general issues in facility location science?
    b. How is dealt with these general issues?
    c. Which ´common´ 1-facility location models do exist in literature?

2) What is Robustness in facility location science?
    a. How is dealt with robustness in literature?
    b. Which method of dealing with robustness is the most interesting?
    c. How is robustness modelled in facility location?
    d. Are there robustness models missing in literature?

3) How to solve robust facility location models?
    a. What are suitable parameters to test robustness?
    b. What are suitable test cases to test robustness?
    c. Which algorithms are applicable to solve a robust model?
    d. Which applicable algorithms solves robustness the best?

4) How can a new robustness model be build?
    a. What should this new model contain?
    b. How should it be build up?
    c. How does the new model work?
    d. Can the new model be solved by algorithms from research question 3?

1.3 Mission and vision

This thesis report aims to contribute with scientific knowledge in the area of facility location problems. The research will explore the opportunities of combining robust objective concepts into economic optimising facility locations, while taken uncertain and unknown data into account. New and unique models can be used to find an economic strategic facility location for all expectable circumstances.

2. Facility location

This chapter presents a brief introduction on facility location. In the introduction (Section 2.1) the concept of facility location is explained and an insight on the variety of facility location problems is given. In the next sections and subsections, several components of facility location models are briefly explained and described.

2.1 Introduction

Facility location concerns the placement of facilities, for various objectives, by use of mathematical models and solution procedures. If a company wants to open a new facility (a shop, a distribution centre, a factory, a power plant, a train station, a radio tower or a satellite), the company is interested in finding a location for this facility that will have an optimal coverage or will make maximum profit or it only will have minimum costs. General facility location models are models that are designed to reach these goals (maximum cover or profit) as good as possible for the placement of one or more facilities (Plastria, 2004).

The unknown new facility location will be denoted by *X*. A company tries to find an optimal location *X*, where the objectives of the company can be met as good as possible. This optimal facility location *(X*)* depends on the function the facility has and the goal the owner of the facility wants to reach with it. The objective of the placement of a supermarket differs from the objective of placing a fire station, and the optimal location of a shoe shop can lay elsewhere for shoe shop owners, depending on their objective (minimum costs or maximal profit can lead to an other *X**).

In literature there are many approaches used to model facility location. Depending on the problem situation, one can distinguish many concepts which play a role in facility location science. Therefore choices have to be made on how to:

- determine the set of candidate points for the new facility location (the feasible area),
- model for one or more facilities,
- model costs and distances,
- model importance of demand points,
- model competitors and competiveness in the model,
- model a time horizon,
- decide what is optimal and what is less optimal (the objective),
- take the capacity of different facilities into account,
- take correlation of similar variables into account,
- take uncertainty and variability of data into account

The following sections discuss these points and describe how other scientific researchers deal with them.

2.2 The feasible area

Before a model is selected, it is important to determine the set of candidate points for the location of the new facility. The set of candidate points is called the ¨feasible set¨ of the problem which represents the area in which the new facility can be located. The feasible area of a location problem can be:

- <u>Discrete;</u> a discrete facility location problem is a problem where *X* has to be placed on any of a finite (manageably large) preselected set of possible facility locations.
- <u>Network;</u> a network facility location problem is a problem in which any point along a network is accepted as a possible site for locating *X*. In case of a railroad network, *X* can be placed at every station (the node of the network) or along every railway (the edges of the network) between the stations.
- <u>Continuous;</u> a continuous facility location problem is a problem where *X* can be located everywhere within a geographical region (an entire country or province).

(Plastria, 2004 and Klose and Drexl, 2005)

The feasible set may be restricted by certain ¨forbidden zones¨, in which facilities should not be located or other types of constraints (Eiselt and Laporte, 1995 and Redondo, 2008).

2.3 One or more new facilities

In literature, a distinction is made between the locations of one or more facilities. In single facility location only one facility has to be located, while in a multi facility location problem more than one facility has to be located. When locating more than one new facility, effects of these facilities and (their locations) must be considered, which make multi facility location problems harder to model and to solve (Plastria, 2004).

2.4 Demand points and distances

The optimal location *(X^*)* of a new facility depends on the relation the facility has with its demand points. These demand points can be customers, suppliers, or other relevant location objects. The location of a demand point is denoted by P_i *($i \in I$)*, where *i* is the demand point and *I* is the set of all demand points (Plastria, 2004).
The location of a new facility carries costs (or gains) with respect to the demand. These costs will be denoted by $c_i(X)$ for the effect that demand point *i* will have on the new facility location *X*. When the costs are measured by distances, $d_i(X)$ is used instead of $c_i(X)$.

In literature two distance measurements are often used; Euclidean and rectangular distances. Both these measurements use two coordinates for the location of the demand points and the facilities. A facility *X* has a longitude coordinate *(x_1)* and latitude coordinate *(x_2)*. Demand point P_i also has a longitude and a latitude coordinate p_{i1} and p_{i2}.
The Euclidean distance between two points is the direct distance between two locations on a map without taking roads or blockades into account. The Euclidean distance between *X* and P_i is represented by $d_e(X,P_i)$, which is calculated by:

$$d_e(X,P_i) = \sqrt{(x_1 - p_{i1})^2 + (x_2 - p_{i2})^2} \tag{2.1}$$

The rectangular distance (also known as the Manhattan distance) is the distance corresponding to the shortest path between two points if you only can move horizontally or vertically. The rectangular distance between *X* and P_i is represented by $d_r(X,P_i)$, which is calculated by:

$$d_r(X,P_i) = |x_1 - p_{i1}| + |x_2 - p_{i2}| \tag{2.2}$$

2.5 Attraction of demand points

Besides the feasible area and the way of calculating distances, it is useful to model the importance of different demand points in the feasible area. For example, if the demand points are representing customers and customer *A* and customer *B* have different volumes of demand, then a company might want to consider customer *A* with a higher demand as more important than customer *B* with less demand. This relation is called attraction. This attraction is denoted by w_i, representing the weight representing the attraction of demand point *i*.

2.6 Competition

The standard location models take only existing demand points into account, in a field where no competitors exist. In most real cases however, similar facilities already exist in the region. These existing facilities influence the behaviour of the demand points. The competition model can be static or dynamic (Redondo, 2008).

2.6.1 Static competition

Static competition models consider the competitors as static, which means that the presence of the competitive facilities is modelled, but the competitors will be inactive during and after the placement of the new facility (or the reaction of the competitor is assumed to be known in advance) (Redondo, 2008).

A very popular static competition model was introduced by David Huff in 1963. This so-called Huff model uses the attraction of the competitive facilities and the distances of a demand point to the different competitive facilities as a measurement for the behaviour of the demand point (for example a customer). The probability that a customer selects a facility (for example a retail outlet) is proportional to its attractiveness and inversely proportional to the distance to it. Consequently, a more attractive facility will attract customers from greater distances (Huff, 1963 and Drezner at al., 2002).

The model of Huff can be seen as a ´gravity model´ which splits the total demand of all the demand points proportionally over the facilities, based on the quality of the facilities and the distance between the demand points and the facilities (Fernandez at al., 2007 and Saiz et al., 2008).

In formula this can be represented by the following model, where P_{ij} represents the proportion that customer *i* will spend at retail outlet *j*:

Indices:

i	index of the demand points (customers),	$i = 1, 2, .., n$.
j	index of the facilities (retail outlets),	$j = 1, 2, .., m$.

Data:

a_j	is a measure of attractiveness of retail outlet *j*
d_{ij}	is the distance from customer *i* to retail outlet *j*
w_i	demand size of demand point *i*

Formula:

$$P_{ij} = w_i * \frac{a_j * d_{ij}}{\sum_{j=1}^{m} a_j * d_{ij}} \quad \text{for all } i \text{ and for all } j \qquad (2.3)$$

(Huff, 1963 and Huff, 2003)

From the viewpoint of the facilities, the Huff model can calculate the market share M_j captured by each retail outlet *j* in the area (Saiz et al., 2008).
In formula M_i can be represented by the following model:

$$M_j = \sum_{i=1}^{n} P_{ij} = \sum_{i=1}^{n} w_i * \frac{a_j * d_{ij}}{\sum_{j=1}^{m} a_j * d_{ij}} \quad \text{for all } j \qquad (2.4)$$

(Huff, 1963, and Huff, 2003)

2.6.2 Dynamic competition

Dynamic competition models consider the competitors in the market as dynamic, which means that the competitors available or not, can act or react on the placement of the new facility that they want to place. The most common problems where dynamic competition plays a role is where two or more players wants to locate their facilities in the market in such a way that their own gain is maximized (Avella et al., 1998).

One of the most interesting models in the dynamic competition area is the ´Stackelberg leadership´ model of Heinrich von Stackelberg in 1934. This model considers two players in the market, and tries to find an optimal solution by anticipating on the placement of the competitor.
First the leader places its facility and then the follower will place his facility sequentially. If one considers itself as leader, the leader will locate its facility first, taken into account the actions of the competing follower (which locates his new facility after the leader). This means that the leader has to decide on its facility location, given the optimal reaction of the follower. As this reaction of the follower depends on the location of the leader, this is a hard problem to solve (Stackelberg, 1934, Redondo, 2008 and Saiz et al., 2008).

2.7 Time horizon

The ´common´ facility location models place one or more facilities at optimal locations at the moment as requested, under the conditions as assumed. This mainly results in the placement of one or more facilities at the given destination as soon as possible, without taking a time horizon into account.
In dynamic competition, time dependencies are taken into account, as interchanges between the moves of other players are formulated into the problem (see subparagraph 2.6.2).

Dynamic, non competition, location models also exist. These dynamic models aim on the placement of more facilities by time dependent placement or opening costs. The main reason for this time dependent placement of facilities is that there is not enough budget available yet to place all desired facilities at once, and therefore an order of placement has to be chosen from which the company will benefit the most (Avella et al., 1998).

2.8 Objectives

Depending on the objective of the new facility, a standard location model can be chosen. The objective can be either a pull or a push objective. A pull objective is an objective that pulls a desired facility (due to the attraction of demand points) to a location that is more or less central in between the demand points. A push objective is an objective that pushes an undesired facility (due to the repulsion of demand points) to a location far away from the demand points (Krarup et al., 2002).
Push objectives are used for:

- the placement of obnoxious facilities: where one wants to maximize the distance to the closest neighbour i.e. maximize the minimum $d_i(X)$, where X can be the location of a nuclear plant, and destinations i can represent civilians.
- dispersion of facilities over a certain area: this is a possible optimization strategy for optimizing competitive market advantage.
- to minimize the cover of a facility i.e. *minimize the sum of* $c_i(X)$, where c_i is the effect of X on i: for example the placement of a waste collection centre, in the case where the range of the smell must cover (reach) as few people as possible.

The pull objectives are by far the most occurring and studied objectives in facility location science. Four of these pull objective location problems play a dominant role in literature and therefore are at times referred to as the prototype location problems;

- the p-median problem minimizes the total (or average) distance between the p facilities and all demand points i.
- the p-centre problem minimizes the maximum distance between the p facilities and the demand point i that is the most remote for all p facilities.
- the uncapacitated facility location problem minimizes all costs belonging to the placement of the facilities taking fixed costs into account.
- the quadratic assignment problem assigns n facilities to n locations in such a way that the total transport cost between these facilities is at the minimum.

(Krarup et al., 2002 and Plastria, 2004)

As the four prototype location models are of main interest in location literature, they will be extended in the following subsections.

The prototype location models are described taken a discrete set of candidate points as feasible area. However, one can also use the same concepts locating on a network or in continuous space.

2.8.1 The p-median problem

The p-median problem is a location-allocation model. This model does not only represent the optimality of a certain facility location, but the model also computes the optimality of the allocation of the demand points to the facilities. The discrete p-median problem deals with the placement of one or several facilities (for example distribution centres) to be located among a given list of sites, indicated by $j = 1, 2, .., m$.

The distribution centres have to serve several demand points (for example retail outlets), indicated by $i = 1, 2, .., n$. The volume of the demand of each retail outlet i is denoted by w_i, and is assumed to be known and fixed. The demand of each retail outlet has to be fully met by the distribution centres, where it is allowed that more than one distribution centre facilitates the same retail outlet.

For each possible distribution centre j and each retail outlet i, the cost it takes to serve one unit of good from location j to retail outlet i is represented by c_{ij}.

The variable y_j is a binary variable (**0 or 1**) representing the presence of a distribution centre at possible location j. If a distribution centre is opened at location j, y_j is **1**, if not y_j is **0** (Plastria, 2004).

Indices:

i	index of the demand points (retail centres), $i = 1, 2, .., n$
j	index of the possible facility locations, $j = 1, 2, .., m$

Variables:

x_{ij}	allocation variable, one for each combination of a retail centre i and a facility j
y_j	location variable, for each possible facility location j

Data:

c_{ij}	cost to serve one unit from location j to retail outlet i
w_i	demand of retail facility i
p	the exact number of distribution centres that should be placed

Objective function:

$$\underset{x}{Min} \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} w_i x_{ij} \qquad (2.5)$$

Subject to:

$$\sum_{j=1}^{m} x_{ij} = 1 \quad \text{for all } i \qquad (2.6)$$

$$x_{ij} \le y_j \quad \text{for all } i \text{, and for all } j \qquad (2.7)$$

$$\sum_{j=1}^{m} y_j = p \qquad (2.8)$$

$$y_j \in \{0,1\} \quad \text{for all } j \qquad (2.9)$$

$$0 \le x_{ij} \le 1 \quad \text{for all } i \text{, and for all } j \qquad (2.10)$$

The p-median problem is used for the case where fixed costs either are not relevant or are equal for all sites. It is assumed that the number of plants is fixed or at least limited and no capacities constraints are concerned. The p in p-median problem represents the exact number of distribution centres (facilities) that should be opened. Therefore Constraint (2.8) is added which states that the sum of all y_j´s must be equal to p (Plastria, 2004).

If the cost represents distances then the model calculates the optimal facility location, which will be located where the average distance is at a minimum. This results in a spatial efficient solution, as it is based on averaging. These solutions are often discriminating for low dense and remote areas (Ogryczak and Zawadzki, 2002).

2.8.2 The p-centre problem

Public services are not allowed to discriminate low dense and remote areas. Therefore the objective for placing a public facility (an alarm siren or a fire station for example) does not aim at a spatial efficient solution but on a spatial equity solution, which must be fair for everyone.
The discrete p-centre problem is developed to ensure equity in servicing users spread on a wide geographical area. To do so, the *p*-centre problem uses a minmax criterion which corresponds to finding the location of a central facility so that the distance to the farthest demand point is as small as possible (Ogryczak and Zawadzki, 2002 and Ghiani et al., 2004).

Indices:

i	index of the demand points, $i = 1, 2, .., n$
j	index of the possible facility locations, $j = 1, 2, .., m$

Variables:

x_{ij}	allocation variable, one for each combination of a demand point *i* and a facility *j*
y_j	location variable, for each possible facility location *j*

Data:

c_{ij}	cost to serve one unit from location *j* to demand point *i*
w_i	demand of demand point *i*
p	the exact number of facilities that should be placed

Objective function:

$$\underset{x}{Min}\ \underset{i}{Max}\{w_i c_{ij} x_{ij}\} \quad \text{for all } i \text{ and for all } j \qquad (2.11)$$

Subject to:

$$\sum_{j=1}^{m} x_{ij} = 1 \quad \text{for all } i \qquad (2.12)$$

$$x_{ij} \le y_j \quad \text{for all } i\text{, and for all } j \qquad (2.13)$$

$$\sum_{j=1}^{m} y_j = p \qquad (2.14)$$

$$y_j \in \{0,1\} \quad \text{for all } j \qquad (2.15)$$

$$0 \le x_{ij} \le 1 \quad \text{for all } i\text{, and for all } j \qquad (2.16)$$

(Caruso et al., 2003 and Mladenovic et al., 2003)

An often used variant of the p-centre problem is the minimum set covering problem. This problem, minimizes the total number of facilities needed to reach all users. Where a_{ij} is a cover marker and *r* is the cover rate.
The cover marker a_{ij} is 1 if demand point *i* is reached by facility location *j*, if not a_{ij} is 0. The cover rate *r* is **1**, if all demand points have to be covered by at least **1** facility. In some safety systems, always a back up is needed and *r* can be 2 or more (Plastria, 2004 and Hendrix and Toth, 2009).

Objective function:

$$Min \sum_{j=1}^{m} y_j \qquad (2.17)$$

Subject to:

$$\sum y_j a_{ij} \ge r \qquad (2.18)$$

$$y_j \in \{0,1\} \quad \text{for all } j \qquad (2.19)$$

2.8.3 The uncapacitated facility location problem

The problem of the discrete uncapacitated facility location problem deals with the placement of one or several facilities (for example distribution centres) to be located among a given list of sites, indicated by *j = 1, 2, .., m.* For each possible site *j* a fixed charge and/or operating cost f_j is to be paid if a plant is placed on site *j*. This fixed cost is considered to be independent of any other variables.
The distribution centres have to serve several demand points (for example retail outlets), indicated by *i = 1, 2, .., n*. The volume of the demand of each retail outlet *i* is denoted by w_i, and is assumed to be known and fixed. The demand of each retail outlet has to be fully met by the distribution centres, where it is allowed that more than one distribution centre facilitates a retail outlet. For each possible facility location *j* and each retail outlet *i*, the cost it takes to serve one unit of good from location *j* to retail outlet *i* is represented by c_{ij} (Plastria, 2004).

The uncapacitated facility location problem can be described by (2.20) to (2.24). This model is similar to the *p*-median problem, except for the added fixed setup costs f_j for locating a facility at candidate location *j*.

<u>Objective function:</u>

$$\underset{x}{Min} \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} w_i x_{ij} + \sum_{j=1}^{m} f_j y_j \qquad (2.20)$$

<u>Subject to:</u>

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad \text{(for all } i\text{)} \qquad (2.21)$$

$$x_{ij} \le y_j \qquad \text{(for all } i\text{, and for all } j\text{)} \qquad (2.22)$$

$$y_j \in \{0,1\} \qquad \text{(for all } j\text{)} \qquad (2.23)$$

$$0 \le x_{ij} \le 1 \qquad \text{(for all } i\text{, and for all } j\text{)} \qquad (2.24)$$

This model decides on which sites *j*, distribution centres have to be opened, and how all retail outlets are served by which distribution centre(s), so as to minimize the total cost of the operation.
The uncapacitated plant location problem is often extended with capacity constraints (see Section 2.9).

2.8.4 The quadratic assignment problem

The quadratic assignment problem (QAP) is one of the most challenging combinatorial optimization problems. The discrete quadratic assignment problem aims to locate *n* facilities to *n* locations with a minimum total cost. The total cost consist of the total transportation costs between the facilities to be placed and the placement cost of each facility.

This assignment problem can be modelled with the use of three *n*-by-n matrices, *A*, *B* and *C*. Matrix *A* represents the amount of units that has to be transported between the facilities to be placed. The distances between the available locations are represented by matrix *B*. Matrix *C* is a cost matrix carrying the placement costs of each facility at each available location (Burkard, 2009).

<u>Indices:</u>

i, k	index of the facilities to locate, $i, k = 1, 2, .., n$
j, l	index of the possible facility locations, $j, l = 1, 2, .., n$

<u>Variables:</u>

x_{ij}, x_{kl}	location variable, for facility i, k and possible location j, l

<u>Data:</u>

a_{ik}	flow from facility i to facility k
b_{jl}	distance from location j to location l
c_{ij}	cost of placing facility i at location j

<u>Objective function:</u>

$$\underset{x}{Min}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}a_{ik}b_{jl}x_{ij}x_{kl}+\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}x_{ij} \qquad (2.25)$$

<u>Subject to:</u>

$$\sum_{j=1}^{n}x_{ij}=1 \quad \text{(for all } i\text{)} \qquad (2.26)$$

$$\sum_{i=1}^{n}x_{ij}=1 \quad \text{(for all } j\text{)} \qquad (2.27)$$

$$x_{ij}, x_{kl} \in \{0,1\} \quad \text{(for all } i\text{, and for all } j\text{)} \qquad (2.28)$$

2.9 Extra constraints

This section discusses equations which are commonly used in literature but are not mentioned before. These equations are written for allocation variable x_{ij} and location variable y_j under the following two conditions, Equation (2.29) and (2.30).

$$0 \le x_{ij} \le 1 \qquad \text{for all } i\text{, and for all } j \tag{2.29}$$

$$y_j \in \{0,1\} \qquad \text{for all } j \tag{2.30}$$

For the location of certain facilities, the placement is not efficient if the use of the facility is below a certain level. M_j is the minimum activity level of facility *j* under which it will not be opened.

$$M_j y_j \le \sum_{i=1}^{n} w_i x_{ij} \qquad \text{for all } j \tag{2.31}$$

Equation (2.32) is used to model the maximum capacity of a facility, where cap_j represents the capacity of facility *j*.

$$\sum_{i=1}^{n} w_i x_{ij} \le cap_j y_j \qquad \text{for all } j \tag{2.32}$$

Transport capacities like Equation (2.33) are used as a limit on the capacity of the amount of goods a truck or other transporter can carry.

$$w_i x_{ij} \le t_{ij} \tag{2.33}$$

Equation (2.34) limits the number of plants in area *S* to be below a predefined quantity *k*.

$$\sum_{j \in S} y_j \le k \tag{2.34}$$

The other way around, Equation (2.35) obliges that in subset *T*, at least one plant must be opened.

$$\sum_{j \in T} y_j \ge 1 \tag{2.35}$$

The number of facilities to be placed can also be an objective. For example Equation (2.36) minimizes the number of plants to be opened.

$$Min \sum_{j=1}^{m} y_j \tag{2.36}$$

For a retailer it can be important that all shops *j* should be within a maximum range of the distribution centre. Equation (2.37) ensures that all shops are in range.

$$y_j \le \sum_{k \in D_j} z_k \qquad \text{for all } j \tag{2.37}$$

This chapter described an introduction on facility location and discussed several issues which play a role in location modelling. The next chapter explains what robustness is and which robustness concepts are used in facility location.

3. Robustness in facility location

For a company, locating a new facility is a strategic decision (mostly for 15 years or even longer). The standard facility location models that are described in Section 2.8, are used with fixed input data that is assumed to be known in advance. In the real situation, input data like demand and travelling costs are not fixed, nor known in advance. The input data is heavily under pressure of time and developments in many fields and by many actors which are uncontrollable. If these disturbances are not properly managed or considered, the chosen facility location can be less optimal as thought. In a bad case, this not so optimal location can lead to unacceptable losses or even bankruptcy.

A way to deal with variations of input data in facility location science is maximizing the robustness of a facility location. Robustness in this context is a measure for the ability how good a facility on a chosen facility location will perform under all expectable circumstances (Snyder, 2006).

In literature, several approaches are used to model robustness. Five robustness concepts are formulated to distinguish between these approaches.

- The Yes or No performance robustness concept
- The Probabilistic robustness concept
- The Deviation robustness concept
- The Safety first robustness concept
- The Maximum regret robustness concept

In this report, the size of demand of all demand points is uncertain and uncontrollable. All other data is considered constant.

This chapter is structured as follows: Section 3.1 presents an uncapacitated facility location model. This model is used as standard model. In Section 3.2, a small numerical example is introduced with uncertain and uncontrollable demand to illustrate the definition of robustness. In Section 3.3, the 5 concepts of robustness are introduced. The numerical example of Section 3.2 is used to illustrate how the robustness concepts determine robustness of a certain facility location.

3.1 A standard single facility location model

This section presents an uncapacitated single facility location model by Equations (3.1) to (3.3). This model is similar to the uncapacitated single facility location model of Section 2.8.3, except for two changes.

1. The fixed costs are removed from the objective function and the constraints.
2. Model (3.1) to (3.3) is in a continuous space, while the model of Section 2.8.3 considers a discrete set of candidate points.

In the model of Section 2.8.3 two data components are available, the size of demand and the location of each demand point. The location of the demand points (for example cities) can be reasonably considered to be constant, but the size of the demand w_i of them can vary easily and are in this model uncertain and uncontrollable variables.

Indices:

i	index of demand points	$i=1,2,\ldots,n$

Variables:

x	location variable for the new facility

Data:

p_i	location of demand point i
V	feasible area for the facility location x
w_i	uncontrollable demand of demand point i, with set W of realisations

Objective function:

$$\underset{x}{Min}\left\{TC(x,w)=\sum_{i=1}^{n} w_i d_i(x)\right\} \tag{3.1}$$

Where $d_i(x)=\sqrt{((x_1-p_{i1})^2+(x_2-p_{i2})^2)}$ *for all i* (3.2)

Subject to:

$$x \in V \tag{3.3}$$

This location model is not yet a robustness model as the goal is minimizing total cost. It is used to illustrate various measures of robustness. In the next section an example will be presented which is used to explain five robustness concepts applicable for facility location science. All five robustness concepts deal with the uncontrollable demand in an alternative way which has consequences for their definition and measurement of robustness.

3.2 A small numerical example

This section presents a small numerical example. This example is of an extremely small size to gain insight in robustness models used. This example is used four times.

1. To illustrate Model (3.1) to (3.3) In this section
2. To illustrate robustness in general In this section
3. To demonstrate the five robustness concepts In Section 3.3
4. To illustrate robustness models found in literature In Chapter 4

The small numerical example has a feasible area *V* of size 5 by 5, where two demand points (cities) are located. A company wants to locate a new facility *x* in feasible area *V* to comply with the demand w_i of each city *i*. The company wants to minimize the total travelling cost (distance*demand) between their facility and the two demand points. Figure 3.1 represents the feasible area with the demand points and two possible facility locations, location *x=(3,3)* is represented in green, *x=(2,2)* in pink.

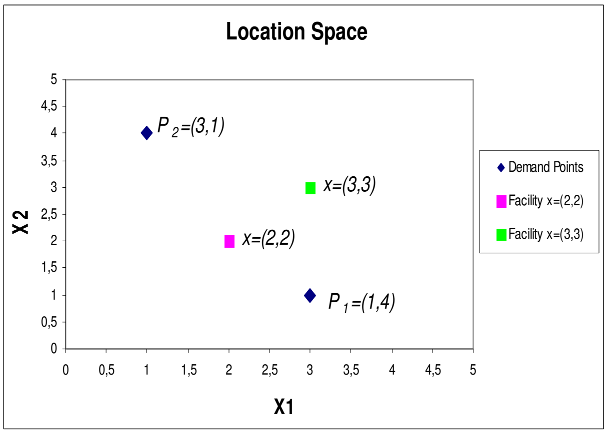

Figure 3.1: Possible location of x in the feasible set

The total travelling cost depends on the distance from the facility to each city and the size of the demand. For a chosen facility location, the distance between the facility and the cities can be calculated and remains constant. Because the distances remains constant, the total cost function only depends on the size of the demand of both cities.
In Figure 3.2, the size of the demands is presented for $1 \leq w_i \leq 4$. Consider facility location *x=(3,3)*. The green line represents the combination of w_1 and w_2 for facility location *x=(3,3)* with a total cost of 15. In symbols this is represented by TC((3,3),w)=15, i.e. TC((3,3),w)= $w_1*2+w_2*\sqrt{5}=15$.

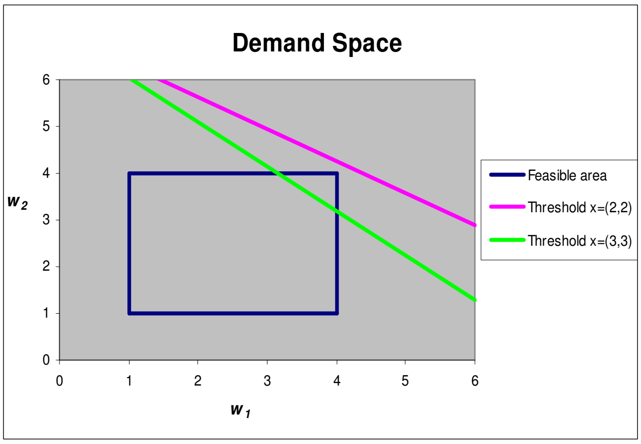

Figure 3.2: Total cost function for combinations of w_1 and w_2 by facility locations x=(2,2) and x=(3,3)

For an other facility location, the distance between the facility location and the demand points is different. Therefore the total cost function is different for every possible facility location.
For facility location *x=(2,2)*, the total cost function is: TC((2,2),w)= $w_1*\sqrt{2}+w_2*\sqrt{5}=15$.

3.3 Robustness concepts

The new goal of the company is to locate a facility, not with minimum cost, but with a maximum robust solution. The robustness goal is to find a location for which the total cost will be below a threshold *T=15* cost units despite uncontrollable variation of the demand size (w_i). A threshold is in this case an upper limit or a maximum budget of the company which represents the total costs the company allows the facility to make to supply the demand to both demand points. The threshold value is represented by the symbol *T.* The minimizing total cost Objective Function (3.1) is replaced by the maximum robustness Objective Function (3.4) where $\rho_g(x)$ represents robustness.
Suffices are used to distinguish between the different concepts of robustness. Suffix ***g*** represents the **g**eneral robustness model.

$$\underset{x}{Max}\{\rho_g(x)\} \qquad (3.4)$$

This robustness model is not complete. The goal of maximizing the robustness value is stated, but how robustness is defined and measured depends on the concept and definition of robustness. In the next subsections five different concepts of robustness are introduced and the consequences of each concept on the robustness model are discussed. The five robustness concepts are:

1. The Yes or No performance robustness concept Section 3.3.1
2. The Probabilistic robustness concept Section 3.3.2
3. The Deviation robustness concept Section 3.3.3
4. The Safety First robustness concept Section 3.3.4
5. The Maximum Regret robustness concept Section 3.3.5

3.3.1 The Yes or No performance robustness concept

The Yes or No performance robustness is a concept which treats robustness as a yes or no parameter. Yes for a solution *x* which performs under all circumstances, and no for a solution which does not work always (Ben-Tal and Nemirovski, 1998 and Hendrix 2008).

To illustrate the Yes or No performance robustness concept, the small numerical example of Section 3.2 is used. For this example, facility location *x* is robust if for all possible values of w_1 and w_2, the total cost of the system is below threshold *T*. The robustness parameter $\rho(x)$ is 1 for a facility location *x* for which the total cost function for all feasible combinations of *w* is below the threshold. The robustness parameter is 0 for any facility location *x*, for which this is not the case.

The robustness model for the Yes or No performance robustness concept is described by following Equation (3.5). Suffix ***y*** represents here the **Y**es or No robustness concept.

$$\rho_y(x) = \begin{cases} 0 & if \quad not \ for \ all \ w \ , \ TC(x,w) \le T \\ 1 & if \quad for \ all \ w \ , \ TC(x,w) \le T \end{cases} \qquad (3.5)$$

In Figure 3.2, the difference between yes and no performance is illustrated. Let w_1 and w_2 vary between 1 and 4. For facility locations *x=(2,2)* and *x=(3,3)*, the total cost function is analysed. The pink line represents budget line TC((2,2), *w*)=15, the green line represents budget line TC((3,3),*w*)=15.

Solution *x=(2,2)* is robust, as for all combinations of the ranges of w_1 and w_2 the total cost is below the threshold of 15 cost units. For facility location *x=(3,3)* this is not the case. Although the total costs for almost the whole feasible range of combinations of w_1 and w_2 is below the threshold, the highest values of w_1 and w_2 do exceed the budget. Therefore facility location *x=(3,3)* is not robust.

In the Yes or No performance concept there is no distinction between two performing and two non performing facility locations. A facility location where only one possible combination of w_1 and w_2 exceeds the threshold is not performing and is assigned a robustness value of 0. A facility location with only combinations of w_1 and w_2 that exceeds the threshold is also not performing and gets a robustness value of 0.

3.3.2 The Probabilistic robustness concept

The probabilistic robustness concept models the level of robustness of different solutions. The probabilistic robustness concept determines the chance that a chosen facility location *x* is performing as required. Therefore a probability distribution of the uncontrollable variables is required (Olieman, 2008).

To illustrate the probabilistic robustness concept, the small numerical example of Section 3.2 is used. Consider, w_1 and w_2 to be uniformly distributed between 1 and 4. Due to this uniform distribution, the chance of every combination of w_1 and w_2 between 1 and 4 is equal. Robustness of facility location *x* is now defined as the probability that the total cost under influence of w_1 and w_2 does not exceed the budget (threshold).
The robustness model for the probabilistic robustness concept is described by Equation (3.6). Suffix ***p*** represents here the **p**robabilistic robustness concept.

$$\rho_p(x) = P\{TC(x,w) \leq T\} \tag{3.6}$$

For facility location *x=(2,2)* (see Figure 3.2), the robustness is 100%. This is logical as the Yes or No performance method already showed that facility location *x=(2,2)* performs under all circumstances. For facility location *x=(3,3)* this is not the case. Here the robustness should be proportional to the surface of the feasible area which is under the budget line. The proportion of the surface area which is above the threshold value is approximately 12.5 %. Therefore the robustness of facility location *x=(3,3)* is 87.5%.

3.3.3 The Deviation robustness concept

The deviation concept uses a completely different definition of robustness. The deviation concept does not need the ranges or the probability distribution of the demand. Only a nominal value *v* of the demand is used which is an expected or average value for demand. Assuming that the chosen facility location *x* performs well for these nominal values *w=v*, the minimum deviation of the uncontrollable variables is measured for which location *x* does not perform as required. Maximum robustness for the deviation concept is obtained by maximizing this minimum deviation.
The robustness model for the deviation concept, is described by Equation (3.7). Suffix ***d*** represents here the **d**eviation robustness concept.

$$\rho_d(x) = \min_w \|w - v\| \text{ for which } TC(x,w) \geq T \tag{3.7}$$

To illustrate the deviation robustness concept, the small numerical example of Section 3.2 is used. Instead of the ranges or the probability distribution for the uncertain demand, only the nominal demand values of w_1 and w_2 have to be known. In the example, the nominal values for both uncontrollable variables are considered to be 2.5. Figure 3.3 represents the deviation robustness for facility location *x=(2,2)* and *x=(3,3)*.

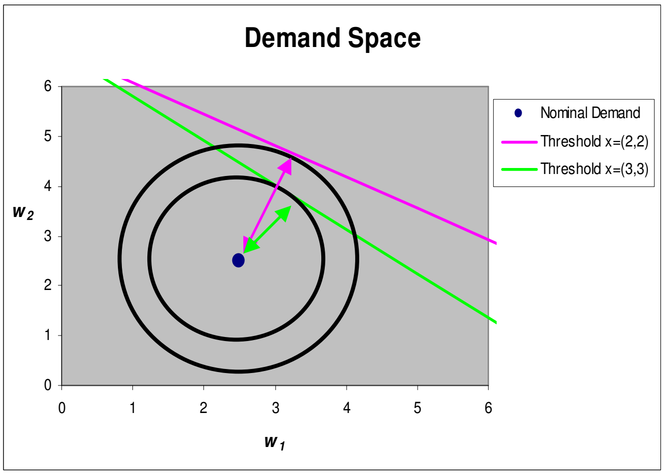

Figure 3.3: Deviation robustness for x=(2,2) and x=(3,3)

The deviation robustness of facility location *x*, is the minimum distance between the nominal demand vector *v* and vector *w* for which the total cost exceeds the threshold of 15 cost units. The blue dot in Figure 3.3 represents the nominal demand, the pink and the green line are the threshold lines for respectively facility location *x=(2,2)* and *x=(3,3)*. The minimum Euclidean distances between the nominal demand and the threshold lines are represented by the arrows, where the pink arrow is the minimum deviation for facility location *x=(2,2)* and the green arrow is the minimum deviation for facility location *x=(3,3)*.

The pink threshold line for facility location *x=(2,2)* is represented by TC((2,2),*w*)=15=(√2)*w_1+(√5)*w_2. For facility location *x=(2,2)*, the minimum distance between the nominal *v* and the threshold *T* is calculated by:

$$p_d(2,2) = \frac{T - d_1(x) * v_1 + d_2(x) * v_2}{\sqrt{d_1(x)^2 + d_2(x)^2}} = \frac{15 - \sqrt{2} * 2.5 + \sqrt{5} * 2.5}{\sqrt{2+5}} = 2.22$$

The green threshold line for facility location *x=(3,3)* is represented by TC((3,3),*w*)=15=2*w_1+(√5)*w_2. For facility location *x=(3,3)* of the example, the minimum distance between the nominal *v* and the threshold *T* is calculated by:

$$p_d(3,3) = \frac{T - d_1(x) * v_1 + d_2(x) * v_2}{\sqrt{d_1(x)^2 + d_2(x)^2}} = \frac{15 - 2 * 2{,}5 + \sqrt{5} * 2{,}5}{\sqrt{2^2 + 5}} = 1{,}47$$

For the deviation robustness concept, facility location *x=(2,2)* is more robust than facility location *x=(3,3)*, which corresponds to the probabilistic robustness concept.

Deviation robustness is determined by the minimum distance between nominal vector *v* and vector *w,* resulting in a total cost equal to the threshold *T*. This minimum distance is represented by ||*w-v*|| which is the mathematical representation of a norm. This distance norm is a function which assigns a strictly positive length or size to vector *w - v*. The distance norm and its function can be interpreted in several ways. Depending on the problem situation a certain distance norm is more appropriate.

The following three distance norms are considered to be the most relevant (Hendrix, 1998):

a) the 1-norm $||w-v||_1$
b) the 2-norm $||w-v||_2$
c) the infinite-norm $||w-v||_\infty$

To illustrate the three deviation robustness concept in the Sections 3.3.3.1 to 3.3.3.3, the small numerical example of Section 3.2 is used.

3.3.3.1 The 1-norm

The one-component distance, or 1-norm distance, focuses on the sum of deviations for each element $|w_i - v_i|$ at the time. The 1-norm deviation robustness (Equation (3.8)) is the minimum sum of all positive deviations between $|w_i – v_i|$, where the threshold T is reached. Equation (3.9) formulates the 1-norm deviation, where $||w-v||_1$ is the norm and |.| is a positive deviation between w_i and v_i for all i.

$$\rho_{d1}(x) = \min\|w - v\|_1 \text{ for which } TC(x, w) \geq T \tag{3.8}$$

$$\|w - v\|_1 = \sum_i^n |w_i - v_i| \tag{3.9}$$

Figure 3.4 considers the uncertainty and uncontrollability of two demand points. The figure resembles the 1-norm distance between nominal demand *v=(2.5,2.5)* and the threshold (represented by the pink line). The green arrow is the minimum 1-norm deviation between v and the threshold T. The dark red diamond resembles all points w on 1-norm distance D_1 from the nominal v. For all points on this diamond, the sum of all positive deviations between $w - v$ is equal to $||w-v||_1$.

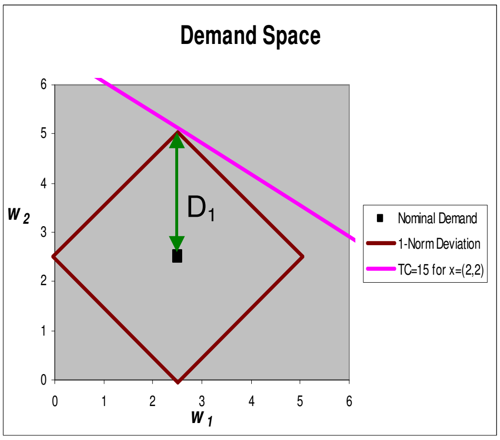

Figure 3.4 The 1-norm minimum deviation robustness

For this two dimensional case, the minimum deviation distance is represented by summing the positive difference between the horizontal and the vertical value. The rectangular distances discussed in Section 2.4 are calculated in the same way. The rectangular distance is also a 1-norm distance.

3.3.3.2 The 2-norm

The 2-norm distance focuses on deviation in all elements $w_i - v_i$ at the same time. The 2-norm deviation robustness (Equation (3.10)) is the minimum of the square root of the sum of all quadratic deviations $(w_i - v_i)^2$, where the threshold T is reached. Equation (3.11) formulates the 2-norm deviation, where $||w-v||_2$ is the 2-norm.

$$\rho_{d2}(x) = \min\|w - v\|_2 \text{ for which } TC(x, w) \geq T \tag{3.10}$$

$$\|w - v\|_2 = \sqrt{\sum_i^n (w_i - v_i)^2} \tag{3.11}$$

Figure 3.5 resembles the 2-norm distance between nominal demand *v=(2.5,2.5)* and the threshold (the pink line). The green arrow between nominal demand *v* and threshold *T* is the minimum 2-norm distance $||w-v||_2$. The dark red circle represents all points *w* which are 2-norm D_2 away from the nominal *v*.

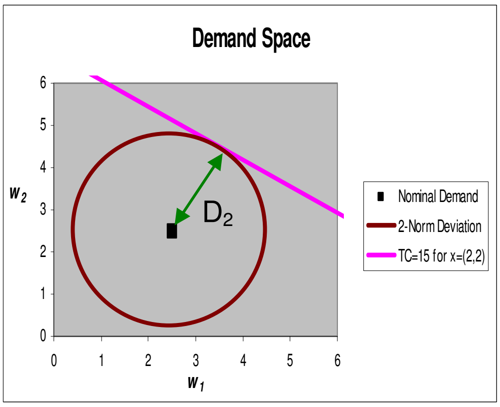

Figure 3.5 The 2-norm minimum deviation robustness

This deviation is calculated on the Euclidian way by taking the square root of the sum of all quadratic deviations w_i - v_i.

3.3.3.3 The infinite-norm

The infinite-norm, or Chebyshev distance, focuses on deviation of all elements $|w_i - v_i|$ at the same time. The infinite-norm deviation robustness (Equation (3.12)) is the minimum of the largest deviation $|w_i - v_i|$ for any *i*, where the threshold *T* is reached. Equation (3.13) formulates the infinite-norm deviation, where $||w-v||_\infty$ is the largest difference between w_i and nominal demand v_i for any *i*.

$$\rho_{d\infty}(x) = \min\|w-v\|_\infty \text{ for which } \quad TC(x,w) \geq T \tag{3.12}$$

$$\|w-v\|_\infty = \max_i |w_i - v_i| \tag{3.13}$$

Figure 3.6 depicts the infinite-norm minimum distance between nominal demand *v=(2.5,2.5)* and the threshold (the pink line). The green arrow is the infinite-norm distance between nominal *v* and threshold *T*. The square resembles all points *w* on infinite-norm distance D_∞ from the nominal *v*.

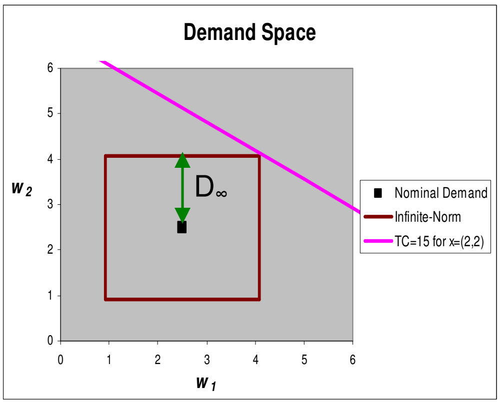

Figure 3.6 The infinite-norm minimum deviation robustness

In warehouse logistics the Chebyshev norm is used to model travel time for order picking, in the case that order picking machines can move in the horizontal and vertical direction at the same time while using both motors at their speed capacity (Langevin and Riopel, 2005).

3.3.4 The Safety First robustness concept

The safety first concept is a conservative notion of robustness on worst case scenario results. The worst case scenario reflects the worst objective function value for all possible combinations of the uncontrollable variables. Robustness is inversely related to the worst case scenario. The lower the objective value of the worst case scenario, the higher the robustness. Therefore the most robust facility location is a location, where the highest possible total cost is as low as possible. The goal for the safety first concept is thus to minimize the worst case scenario outcome for the new facility location *x*.

The robustness model for the safety first concept, is described by Equation (3.14). Suffix ***s*** represents here the **s**afety first robustness concept. The **–** sign before the max deals with the inverse relation with robustness.

$$\rho_s(x) = -\max_{w}\{TC(x,w)\} \qquad (3.14)$$

To illustrate the safety first robustness concept, the small numerical example of Section 3.2 is used. Taking the ranges $1 \le w_i \le 4$, for both demand points *i*, the worst case outcome is the maximum demand *w=(4,4),* as the total cost in the numerical example is proportional to the size of demand. In Figure 3.7 the dark red arrow points out the worst case weight scenario for the numerical example.

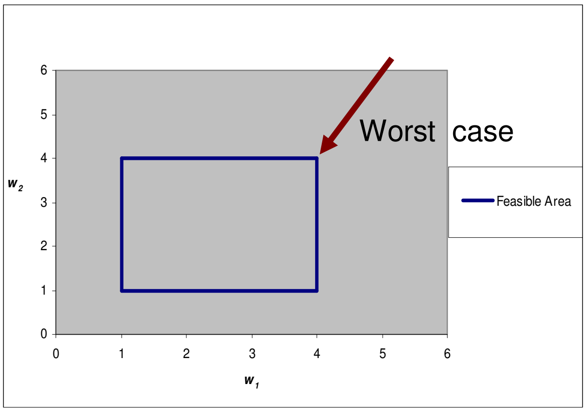

Figure 3.7: Safety First Robustness for the small numerical example

The total cost is determined by the sum of the weighted distances. For facility locations *x=(2,2)* and *x=(3,3)* the worst case total costs are calculated.

$$TC_{((2,2),(4,4))} = w_1 * d_1(x) + w_2 * d_2(x) = 4*\sqrt{2} + 4*\sqrt{5} = 14{,}60$$
$$TC_{((3,3),(4,4))} = w_1 * d_1(x) + w_2 * d_2(x) = 4*2 + 4*\sqrt{5} = 16{,}94$$

As the worst case scenario for facility location *x=(2,2)* is lower than that of location *x=(3,3)*, facility location *x=(2,2)* is more robust than facility location *x=(3,3)*, which corresponds with the probabilistic and the deviation robustness concept.

3.3.5 The Maximum Regret robustness concept

A variant on the safety first concept is to minimize the maximum regret. The maximum regret concept is also a worst case scenario analysis but not applied to the absolute function values but to the regret values. The regret value, or opportunity loss, is the difference between the total costs of placing the new facility on location *x* (or on the optimal facility location *x*(w)=z(w)*), given a realization of the demand *w*. Maximum regret is the maximum of this difference for all possible demand weights.
Robustness is inversely related to the maximum regret. The smaller the maximum regret, the higher the robustness, The goal of the maximum regret concept is to minimize this maximum difference in function value for facility *x*, and for the optimal facility location *z* for all possible demand weights.

The robustness model for the maximum regret concept, is described by Equation (3.15). Suffix ***r*** represents here the maximum **r**egret concept. The **–** sign before the max deals with the inverse relation with robustness.

$$\rho_r(x) = -\max_w \{TC(x,w) - \min_z TC(z,w)\} \qquad (3.15)$$

To illustrate the maximum regret robustness concept, the small numerical example of Section 3.2 is used. For this numerical example, the maximum regret is simple as *z* is easy to determine. Figure 3.8 illustrates how the optimum location *z* depends on the size of both demands.

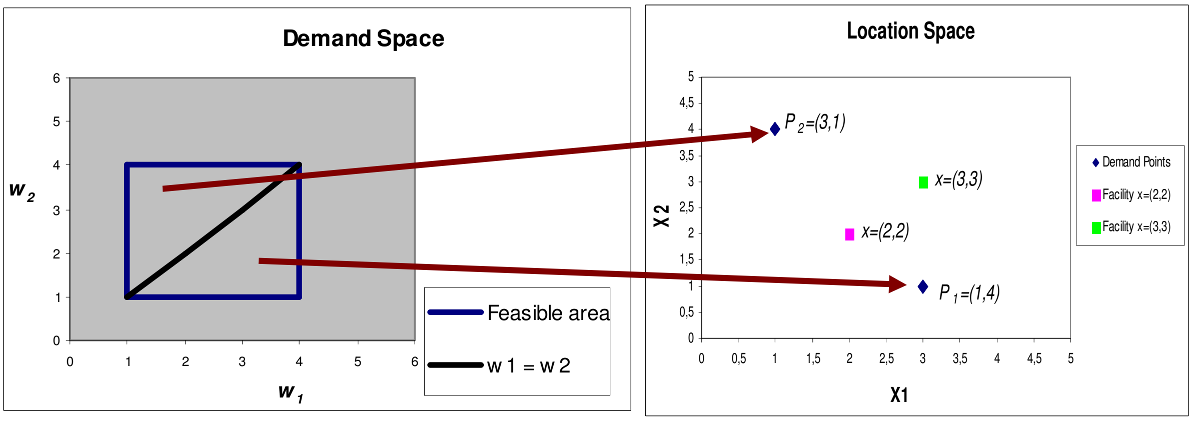

Figure 3.8: Maximum Regret for the small numerical example of Section 3.2

If $w_1>w_2$, the optimal facility location *z* is p_1, with $TC(z,w)= w_1*\sqrt{13}$. If $w_1<w_2$, the optimal facility location *z* is p_2, with $TC(z,w)= w_2*\sqrt{13}$. If $w_1=w_2$, the optimal facility location *z* is located on the line segment λ*(1,3)+(1-λ)*(4,1) for *λ* ranges between 0 and 1, with $TC(z,w)= w_1*\sqrt{13}$. This means that for the example, the total cost of the optimal facility location *z* is given by one of the following formulas.

$$TC(z,w) = w_1 * \sqrt{13} \qquad \text{or}$$
$$TC(z,w) = w_2 * \sqrt{13} \qquad \text{or}$$
$$TC(z,w) = w_1 * \sqrt{13}$$

The robustness is the inverse of the maximum difference between *TC(x,w)* and *TC(z,w)*, for all *w*.

$$TC((2,2),w) = w_1 * d_1(x) + w_2 * d_2(x) = w_1 * \sqrt{2} + w_2 * \sqrt{5}$$
$$TC((3,3),w) = w_1 * d1(x) + w_2 * d_2(x) = w_1 * 2 + w_2 * \sqrt{5}$$

Here the maximum difference can be found for the largest values of *w*. As *w* ranges between 1 and 4, *w=(4,4)* determines the robustness. The optimal facility location for these weight is on all locations of the line segment λ*(1,3)+(1-λ)*(4,1). The total cost of this line segment is TC(z)=4*√13=14.42.

For facility location *x=(2,2)* the total cost for *w=(4,4)* is TC((2,2),(4,4))=4*√2+4*√5=14.60 cost units. The opportunity loss for location *x=(2,2)* is 14.60-14.42=0,18 cost units, therefore the robustness of facility location *x=(2,2)* is -0.18.

For facility location *x=(3,3)* the total cost for *w=(4,4)* is TC((3,3),(4,4))=4*2+4*√5=16.94 cost units. The opportunity loss for location *x=(3,3)* is 16.94-14.42=2.52 cost units, therefore the robustness of facility location *x=(3,3)* is -2.52, which is less robust than facility location *x=(2,2)*.

In the next chapter the robustness models which are found in literature are illustrated by use of the small numerical example of Section 3.2. The models will be introduced by comparison to one of the five robustness concepts.

4. Studied Robustness Models in Literature

This chapter contains the robustness facility location models found in literature. For each robustness concept, first the general robustness model is formulated in a continuous field. The formulated model represents the robustness concept for the small numerical example case of Section 3.2. For the models found in literature is illustrated where they differ from the formulated general robustness models.

Section 4.1 discusses robustness models using the Yes or No performance robustness concept, Section 4.2 the probabilistic robustness concept, Section 4.3 the deviation concept, Section 4.4 the safety first concept and Section 4.5 the minimize maximum regret concept.

4.1 Yes or No Performance Robustness models

The standard Yes or No performance robustness concept of Section 3.3.1 is presented as continuous model described by Equations (4.1) to (4.4) in Section 4.1.1. No relevant models have been found in literature to illustrate in this thesis.

4.1.1 The standard Yes or No Performance robustness model

The standard Yes or No performance robustness model for the uncapacitated facility location problem of Section 3.1 is build up as follows:

<u>Indices:</u>

i	index of demand points	*i=1,2,…,n*

<u>Variables:</u>

x	location variable for the new facility

<u>Data:</u>

p_i	location of demand point *i*
T	Threshold value
V	feasible area for the facility location *x*
w_i	uncontrollable demand of demand point *i*, with set of realisation of *W*

<u>Objective function:</u>

$$\underset{x}{Max}\{\rho_y(x)\} \tag{4.1}$$

Where: $$\rho_y(x)=\begin{cases}0 & if \;\; not \;\; for \; all \; w\in W \;\; TC(x,w)\le T \\ 1 & if \;\; for \; all \; w\in W \;\; TC(x,w)\le T\end{cases} \tag{4.2}$$

$$TC(x,w)=\sum_{i=1}^{n} w_i d_i(x) \tag{4.3}$$

$$d_i(x)=\sqrt{((x_1-p_{i1})^2+(x_2-p_{i2})^2)} \tag{4.4}$$

<u>Subject to:</u>

$$x\in V$$

4.2 Probabilistic concept Robustness models

The standard probabilistic robustness concept of Section 3.3.2 is presented as model described by Equations (4.5) to (4.8) in Section 4.2.1.
Section 4.2.2 deals with a threshold satisfying location model of Drezner, et al., 2002.
In Section 4.2.3, the probabilistic location problem with discrete demand weights of Berman and Wang, 2004, is illustrated, followed in Section 4.2.4 by the probabilistic 1-maximum covering problem with discrete demand weights, also from Berman and Wang, 2008. In Section 4.2.5, two probabilistic models for the 1-center problem in the plane with independent random weights of Pelegrin, Fernandez and Toth, 2008, are discussed.

4.2.1 The standard Probabilistic robustness model

The standard probabilistic robustness model for the uncapacitated facility location problem of Section 3.1 is build up as follows:

Indices:

i	index of demand points	$i=1,2,\dots,n$

Variables:

x	location variable for the new facility

Data:

p_i	location of demand point i
T	Threshold value
V	feasible area for the facility location x
w_i	uncontrollable demand of demand point i, with a distribution over support set W

Objective function:

$$\underset{x}{Max}\{\rho_p(x)\} \tag{4.5}$$

Where $$\rho_p(x) = P\{TC(x,w) \le T\} \tag{4.6}$$

$$TC(x,w) = \sum_{i=1}^{n} w_i d_i(x) \tag{4.7}$$

$$d_i(x) = \sqrt{((x_1 - p_{i1})^2 + (x_2 - p_{i2})^2)} \tag{4.8}$$

Subject to:

$$x \in V$$

4.2.2 A threshold satisfying competitive location model (2002)

In ‘A threshold-satisfying competitive location model’ of Tammy Drezner, Zvi Drezner and Shogo Shioge (2002), robustness is defined as minimizing the variance of cost or profit to reduce uncertainty. This is modelled in their paper by minimizing the probability that revenues fall short of a given threshold which is the minimum revenue needed for survival.

Drezner et al. (2002), consider survival as the main consideration for a new entrant to the market. Therefore the objective of their model is to minimize the probability that revenues fall short of a given threshold, instead of maximizing profit or minimizing cost. The threshold is a minimum market share which is needed for survival of the new entrant. This means that their model searches for the optimum location of a new facility, where the chance of survival is maximized. After survival is secured, the objective may change to the more common one of maximizing profit or minimizing costs (Drezner et al., 2002).

The model tries to minimize the probability that the market share is below the threshold. Robustness is the inverse of this objective function. Robustness is the probability that the market share is above the threshold. Instead of minimizing the probability that the market share is below the threshold it is also possible to maximize the robustness. To adapt Objective Function (4.6) for the Drezner et al. model, it can be replaced by Objective Function (4.9).

$$\rho_p(x) = P\{Z > (T - \mu(x)) / \sigma(x)\} \qquad (4.9)$$

In this function, μ represents the mean of the market share and σ is its standard deviation.

The mean of the market share is based on the competitive Huff model (see Section 2.6.1).

$$\mu(x) = \sum_{i=1}^{n} w_i * \frac{A * d_i^{-\lambda}}{A * d_i^{-\lambda} + \sum_{j=1}^{k} \alpha_j * \partial_{ij}^{-\lambda}} \qquad (4.10)$$

, where A is the attractiveness of the new facility x, α_j is the attractiveness of competitor facility j, $d_i(x)$ is the distance between demand point i and the new facility and δ_{ij} is the distance between demand point i and competitor facility j. Parameter λ represents the distance decay.
Distance decay reflects a decay in importance over distance. If the distance decay is 2, a demand point at distance 2 is reflected at being 2^2 far away, at distance 4. A distance decay of 1 means that there is no decay in distance. In this model the distance decay is negative. If the distance decay is higher (more negative) an object is closer.

An extra element in this probabilistic model is that demand of demand point i is assumed to be correlated to the demand of every other demand point. This assumption is based on the fact, that in times of economic growth or in times of a financial crisis all demand points are affected and their demand can likely be affected in a similar way. The standard deviation of the market share captured is influenced by this correlation.
The standard deviation of the market share is

$$\sigma(x) = \sqrt{\sum_{i=1}^{n} \sum_{m=1}^{n} r_{im} * \sigma_i * \sigma_m * M_i(x) * M_m(x)} \qquad (4.11)$$

, in which σ_i is the standard deviation of the demand of demand point i, σ_m is the standard deviation of the demand of demand point m, r_{im} is the correlation coefficient of demand point i with demand point m and M_i is the market share captured by facility location x from demand point i. The captured market share is given by Equation (4.12).

$$M_i(x) = \frac{A * d_i^{-\lambda}}{A * d_i^{-\lambda} + \sum_{j=1}^{k} \alpha_j * \partial_{ij}^{-\lambda}} \qquad (4.12)$$

The small numerical example of Section 3.2 is used to illustrate how this model works. For simplicity of the illustration, the correlation between the demand points is neglected. The captured market share is in that case resembled by Equation (4.10).
Consider w_1 and w_2 to be uniformly distributed between 1 and 4. One competitor is located at location (3,3). The goal is to capture a market share of 3 units of demand. For facility location *x=(2,2)* the mean captured market share *($\mu_{(2,2)}$)* is calculated using Equation (4.10). The attractiveness of the new facility and the competitor are considered to be 1, and the distance decay is assumed to be neglectible for the small example.

Robustness is the probability that the captured market share is above the threshold. To calculate the robustness, the mean and the variance have to be calculated.

The mean is represented by the expected value of a probability distribution. The expected value of the uniformly distributed weights is the average of the range which is 2.5 for both weigths. The mean market share for facility *x=(2,2)* is

$$\mu_{(2,2)} = w_1 * \frac{1*d_1(x)}{1*d_1(x)+1*d_1(c)} + w_2 * \frac{1*d_2(x)}{1*d_2(x)+1*d_2(c)} = w_1 * \frac{\sqrt{2}}{\sqrt{2}+2} + w_2 * \frac{\sqrt{5}}{\sqrt{5}+\sqrt{5}}$$

$$\mu_{(2,2)} = w_1 * 0.4142 + w_2 * 0.5 = 2.5 * 0.4142 + 2.5 * 0.5 = 2.29$$

The variance of a uniform distribution is is given by $\sigma_1^2=\sigma_2^2=1/12*3^2=3/4$ (Claassen et al., 2007), such that the standard deviation of the market share for facility *x=(2,2)* is

$$\sigma_{(2,2)} = \sqrt{3/4 * \left(\frac{\sqrt{2}}{\sqrt{2}+2}\right)^2 + 3/4 * \left(\frac{\sqrt{5}}{\sqrt{5}+\sqrt{5}}\right)^2} = \sqrt{3/4 * 0.4142^2 + 3/4 * 0.5^2} = 0.562$$

The robustness is the probability that the captured market share is above the threshold.

$$\rho_{p(2,2)} = P\{Z > (T - \mu_{(2,2)}) / \sigma_{(2,2)}\} = P\{Z > (3-2.29)/0.562\} = P\{Z > 1.26\} = 10.2\%$$

4.2.3 Probabilistic location problems with discrete demand weights (2004)

In ´probabilistic location problems with discrete demand weights´ of Oded Berman and Jiamin Wang, four probabilistic robustness location problems are considered. In these models the uncontrollable demand is represented by independent discrete random variables. Two of the models are for desired facilities (the 1-center and the 1-median problem) and two models are for undesired facilities (the 1-antimedian and the 1-anticenter problem).

The way of dealing with robustness is exactly the same for all these models. The goal is to maximize the probability that the standard objective function value is above or below a threshold. The standard objective function is here for example the total weighted distance for the 1-median problem (Berman and Wang, 2004).

The small numerical example of Section 3.2 is used to illustrate how this model works. The way of modelling is similar to the probabilistic robustness concept which is illustrated in Section 3.3.2. The probabilistic robustness concept is identically measured, but the uncertain demand is not treated the same.

In the probabilistic robustness concept of Section 3.3.2, demand is uniformly distributed between 1 and 4. This means that all combinations of w_1 and w_2 between 1 and 4 occur with the same probability. In the article of Berman and Wang, the demand is represented by random discrete demand weights. Discrete means here that the value of *w* do not vary continuously between 1 and 4, but only take pre-selected values.

In Figure 4.1, the discrete random weights are represented by red dots. The selection of these possible discrete demands is random. This means that the chance of the individual discrete weights is equal for all discrete weights (1/16th).

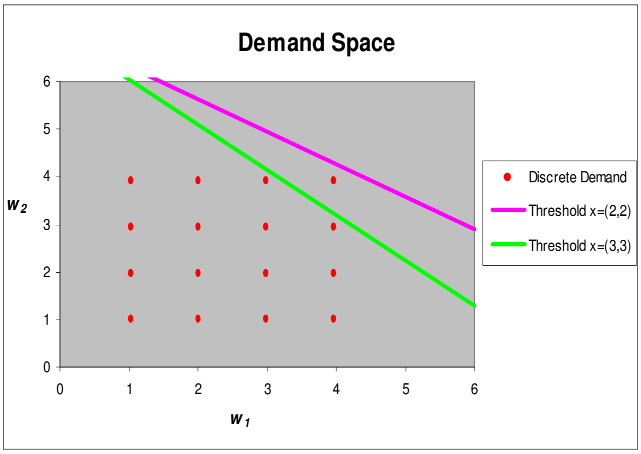

Figure 4.1: Probabilistic robustness for the example with discrete random weights

Facility location *x=(2,2)* is 100% robust, as all possible combinations of demand result in a total cost lower than the threshold of 15 cost units.

For facility location *x=(3,3)*, one of the 16 combinations has a higher cost than the threshold. For *w=(4,4)*, the total costs are higher than 15. As all 16 combinations of demand have the same probability, the probability robustness for facility location *x=(3,3)* is (15/16=) 93.8%.

4.2.4 Probabilistic 1-maximum covering problem with discrete demand weights (2008)

Another article of Oded Berman and Jiamin Wang about a probabilistic robustness model is ´The probabilistic 1-maximal covering problem on a network with discrete demand weights´. As in the previous section, this model deals with the uncontrollable demand as independent discrete random variables.

The goal of the model is to find a facility location *x*, with a maximum probability that the total covered demand is greater than or equal to a pre-defined threshold value. The total demand of a demand point is covered if the demand point lies within a maximum service distance *R* (Berman and Wang, 2008). The total cost Function (4.7) is replaced by total cost Function (4.13) with an extra binary variable y_i. The variable y_i is 1 if demand point *i* lies within the cover radius *R* of facility location *x*, and 0 otherwise. Equation (4.14) is added to define variable y_i.

$$TC(x,w) = \sum_{i=1}^{n} y_i w_i \tag{4.13}$$

Where: $$y_i = \begin{cases} 0 & if \quad d_i(x) > R \\ 1 & if \quad d_i(x) \le R \end{cases} \quad \textit{for all } i \tag{4.14}$$

The small numerical example of Section 3.2 is used to illustrate this model. Assumed is that the cover radius *R* of the new facility is 2.5, represented by the red circle in Figure 4.2. For facility *x=(3,3)*, only demand point p_1 is covered (y_1=0 and y_2=1) and therefore only the demand of p_1 is attracted.

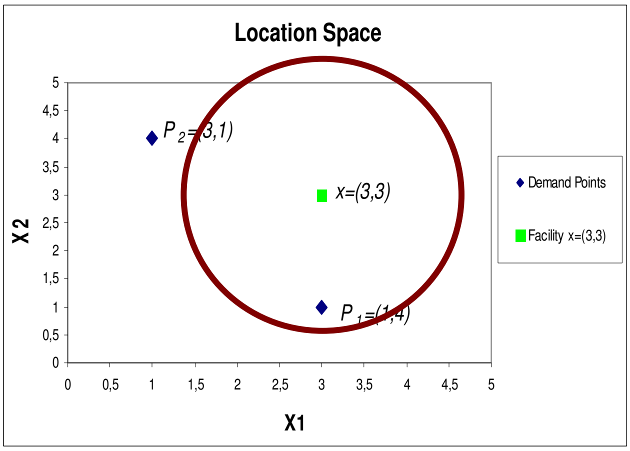

Figure 4.2: Cover radius for facility location x=(3,3)

The robustness threshold value is 3.5. This means that robustness is the probability that a new facility location attracts at least 3.5 units of demand. The threshold is represented in Figure 4.3 by the pink line and the discrete random weights by the red dots.

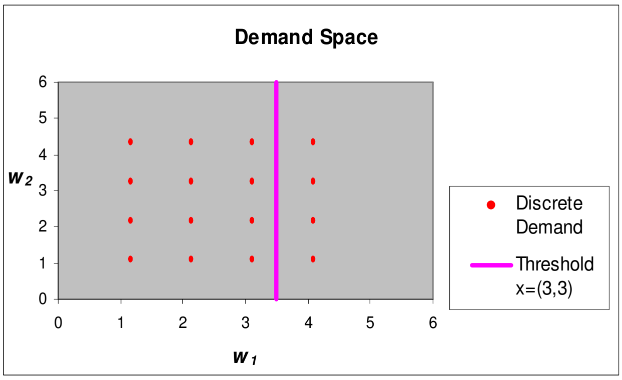

Figure 4.3: Robustness threshold for facility location x=(3,3)

The probability that the attracted demand of facility *x=(3,3)* is at least 3.5 is in 4 of the 16 cases of the discrete random weights. As these weights are random, $\rho_p(3,3)$ = 4/16 = 25 %.

4.2.5 The 1-center problem in the plane with independent random weights (2008)

In the article ´the 1-center problem in the plane with independent random weights´ of Blas Pelegrin, Jose Fernandez and Boglarka Toth, the 1-center problem is considered with weighted distances. The distance between a facility *x* and demand point *i*, is constant but the time it takes to travel this distance is uncontrollable and not constant due to possible traffic jams or unforeseen redirections. Therefore the distances are weighted, with weights representing the travel time. These weights are supposed to be independent random variables with arbitrary probability distributions. Therefore the model of Pelegrin, Fernandez and Toth takes randomly generated values for the weight of the distance instead of fixed values (Pelegrin et al., 2008).

In the paper, two probabilistic measurements for robustness are used. In the first part of the article, robustness is the probability that the maximum weighted distance from the facility to all demand points does not exceed a given threshold. The most robust location is a facility location where the probability is as high as possible. To model this robustness, Objective Function (4.6) can be rewritten as Objective Function (4.15).

$$\rho_p(x) = P\{M \leq T\} \tag{4.15}$$

Where $M = \max_i \{w_i \, d_i(x)\}$ (4.16)

The small numerical example of Section 3.2 is used to illustrate this model. Consider a threshold value of 5 for the small example case.
Robustness is the probability that the maximum weighted distance is smaller or equal to 5. The green line in Figure 4.4 represents the threshold level. For facility location *x=(3,3)*, weight location *w=(√5,2.5)* has a weighted distance of 5 to both demand points. Robustness is the probability that the weighted maximum distance is below the threshold. As the weights are random between 1 and 4, the robustness is equal to the relative feasible area under the threshold line: $\rho_p(3,3)$ = 20.6 %.

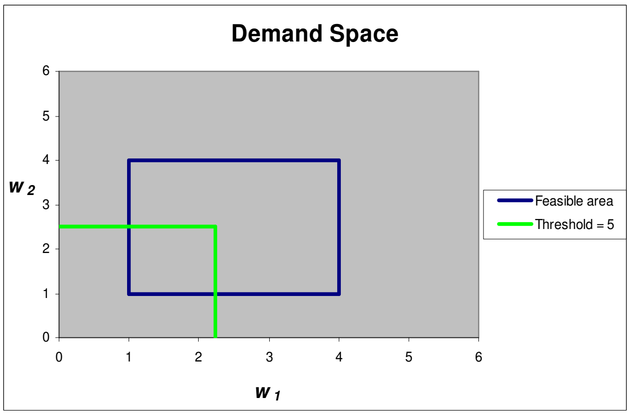

Figure 4.4: Example case for the first 1-centre problem

In the second part of the article, the goal is to minimize the threshold T. If the threshold is lowered, the probability that *M≤T* decreases.
A parameter *c* is introduced. Parameter *c* represents the minimum probability covering which is required. This means that *c* is a chance constraint on the robustness to restrict the probability of covering to at least the value of *c*. The objective in this second article is not to maximize the probability, but to minimize the threshold (or quantile of demand) for which the robustness is higher than parameter *c*. This model is formulated by Objective Function (4.5) and Equations (4.17-4.19).

$$\rho_p = -T \tag{10.17}$$

$$P\{M \leq T\} \geq c \tag{10.18}$$

Where $M = \max_i \{w_i \, d_i(x)\}$ (10.19)

The small numerical example of Section 3.2 is used to illustrate this model with a probability constraint, which should be satisfied.
The coverage probability ratio is considered 35%, which means that the probability that the maximum weighted distance is below the threshold must be at least 35%. In Figure 4.5, the lowest threshold value for facility location *x=(3,3)*, where the cover ratio is 35%, is represented by the green line. The corresponding minimum threshold is 5.875.

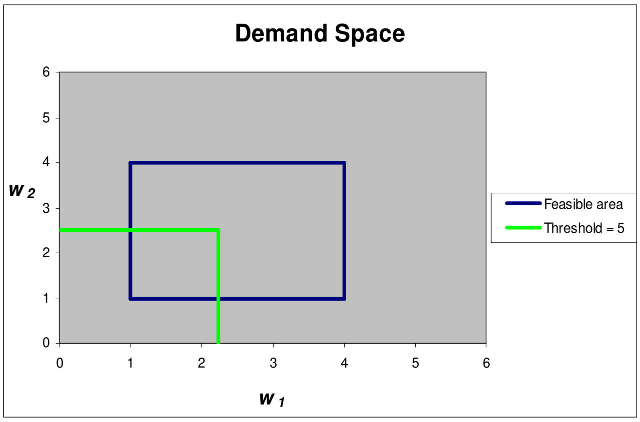

Figure 4.5: Example case for the second 1-centre problem

The goal of the model is to find a facility location *x*, where the threshold value is as low as possible with a cover ration of 35% (or higher).

4.3 Deviation concept Robustness models

The standard deviation robustness concept of Section 3.3.3 is presented as model described by Equations (4.20) to (4.23) in Section 4.3.1. Section 4.3.2 deals with a robust facility location model of Carrizosa and Nickel, 2003.

4.3.1 The standard Deviation robustness model

The standard deviation robustness model for the uncapacitated facility location problem of Section 3.1 is build up as follows:

<u>Indices:</u>

i	index of demand points	*i=1,2,…,n*

<u>Variables:</u>

x	location variable for the new facility
w	demand deviating from nominal value v

<u>Data:</u>

p_i	location of demand point *i*
T	Threshold value
V	feasible area for the facility location *x*
v_i	nominal demand value of city *i*

<u>Objective function:</u>

$$\underset{x}{Max}\{\rho_d(x)\} \tag{4.20}$$

Where $\rho_d(x) = \min_w \|w - v\|$ for which $TC(x,w) \geq T$ (4.21)

$$TC(x,w) = \sum_{i=1}^{n} w_i d_i(x) \tag{4.22}$$

$$d_i(x) = \sqrt{((x_1 - p_{i1})^2 + (x_2 - p_{i2})^2)} \tag{4.23}$$

<u>Subject to:</u>

$x \in V$

4.3.2 Robust facility location (2003)

In the article ´robust facility location´ of Emilio Carrizosa and Stefan Nickel, the standard Weber problem is considered with unknown demand. The Weber problem is a facility location problem with the objective to minimize the total distance between the facility and all demand points. An estimator of the demand is given for which the possible deviation may be non-negligible. A threshold value (or budget) is introduced which represents the highest admissible cost under which the new facility is still profitable. Robustness is here the minimum deviation of the estimator for which the total costs exceeds the budget. If the threshold is easy to exceed, robustness is low. Optimal robustness is the location where the minimum deviation that is needed for w to exceed the threshold is at a maximum (Carrizosa and Nickel, 2003).

This model is precisely represented by the standard deviation robustness model described by Equations (4.20) to (4.23). The model is already illustrated in Section 3.3.3 by use of the small numerical example of Section 3.2.

4.4 Safety first concept Robustness models

The standard deviation robustness concept of Section 3.3.4 is presented as model described by Equations (4.24) to (4.27) in Section 4.4.1. Section 4.4.2 describes the p-median problem in a changing network of Serra and Marianov, 1998. Section 4.4.3 deals with the conditional median as a robust solution concept of Ogryczak, 2009.

4.4.1 The standard Safety first robustness model

The standard safety first robustness model for the uncapacitated facility location problem of Section 3.1 is build up as follows:

Indices:

i	index of demand points	$i=1,2,\ldots,n$

Variables:

x	location variable for the new facility

Data:

p_i	location of demand point i
T	Threshold value
V	feasible area for the facility location x
w_i	uncontrollable demand of demand point i, with possible outcome set W

Objective function:

$$\underset{x}{Max}\{\rho_s(x)\} \tag{4.24}$$

Where $$\rho_s(x) = -\max_{w \in W}\{TC(x,w)\} \tag{4.25}$$

$$TC(x,w) = \sum_{i=1}^{n} w_i d_i(x) \tag{4.26}$$

$$d_i(x) = \sqrt{((x_1 - p_{i1})^2 + (x_2 - p_{i2})^2)} \tag{4.27}$$

Subject to:

$$x \in V$$

4.4.2 The p-median problem in a changing network (1998)

In the paper ´the p-median problem in a changing network: the case of Barcelona´ of Daniel Serra and Vladimir Marianov, a robustness model is formulated to deal with uncertainty in demand, travel time or travel distance. This article presents a discrete location model formulation to address the p-median problem under uncertainty. The model is applied to the location of fire stations in Barcelona. (Serra and Marianov, 1998)

The model of Serra and Marianov minimizes the total cost across all possible scenarios of demand. The cost represents here the travel time from the fire station to a possible place of fire. Robustness is here reverse to the maximum total travel time from fire station to a fire place i for all possible values of w.

This model is precisely represented by the standard safety first robustness model described by Equations (4.24) to (4.27). The model is already illustrated in Section 3.3.4 by use of the small numerical example of Section 3.2.

4.4.3 Conditional median as a robust solution concept (2009)

In the article ´conditional median as a robust solution concept for uncapacitated location problems´ of Wlodzimierz Ogryczak, the conditional median is used to measure robustness. The conditional median uses the concept quantile or portion of demand. In this case the worst portion of demand is considered. If the quantile is 50%, the conditional median considers the 50% worst (largest distance) demand points. The conditional median takes the average distance from *x* to these 50% worst demand points as function value. The robustness concept used is safety first, which means that the worst outcome of this function value for all possible *w* is to be minimized (Ogryczak, 2008 and Ogryczak, 2009).

In the model, a quantile is represented by α. Objective Function (4.25) is replaced by Function (4.28), where Equation (4.29) defines the parameter y_i. This parameter represents if demand point *i* belongs to the quartile farthest demand points $(y_i=1)$ or not $(y_i=0)$.

$$\rho_s(x) = -\max_{w \in W} \left\{ \sum_{i=1}^{n} y_i w_i d_i(x) \right\} \tag{4.28}$$

Where: $$y_i = \begin{cases} 0 & if \quad d_i(x) \notin \alpha \\ 1 & if \quad d_i(x) \in \alpha \end{cases} \qquad \textit{for all } i \tag{4.29}$$

The small numerical example of Section 3.2 is used to illustrate this model. Consider w_1 and w_2 ranges between 1 and 4, and a quantile of 50%.

Robustness is inverse to the maximum weighted distance to the 50% worst demand. For facility location *x=(2,2)*, demand point 2 is further away than demand point 1. If *w=(4,4)*, the 50% quantile of worst case demand is the total demand of demand point 2, as it is the furthest demand point. In the case that *w=(3,4)*, the 50% quantile of worst case demand is 3.5 of demand point 2. If *w=(4,3)*, the 50% quantile demand is the total of demand point 2, and 0.5 demand units of demand point 1. Robustness is the worst case demand, which in this case is at the highest value for maximum weights. As demand point 2 is the furthest demand point, the safety first robustness for facility location *x=(2,2)* is:

$$\rho_s(2{,}2) = -\max_{w} \{w_2\} * d_2(2{,}2) = 4 * \sqrt{5} = 8.94$$

For facility location *x=(3,3)*, demand point 2 is also the furthest demand point, and also on distance √5. The safety first robustness of facility location *x=(3,3)* is:

$$\rho_s(3{,}3) = -\max_{w} \{w_2\} * d_2(3{,}3) = 4 * \sqrt{5} = 8.94$$

For this specific case, by the given ranges and quantile, the robustness of both locations is the same.

4.5 Minimize maximum regret Robustness models

The standard deviation robustness concept of Section 3.3.5 is presented as model described by Equations (4.30) to (4.33) in Section 4.5.1. Section 4.5.2 deals with the minmax regret p-centre location on a network with demand uncertainty of Averbakh and Berman, 1998. Section 4.5.3 contains the facility location problems wit uncertainty on the plane of Averbakh and Bereg, 2005. Section 4.5.4 presents the multi-criteria minisum location problem of Fernandez, Nickel, Puerto and Rodriguez-Chia, 2001.

4.5.1 The standard Maximum Regret robustness model

The standard maximum regret robustness model for the uncapacitated facility location problem of Section 3.1 is build up as follows:

Indices:

i index of demand points *i=1,2,…,n*

Variables:

x location variable for the new facility

Data:

p_i location of demand point *i*
T Threshold value
V feasible area for the facility location *x*
w_i uncontrollable demand of demand point *i*, with outcome set *W*
z optimum facility location for a specific *w*

Objective function:

$$Max\{\rho_r(x)\} \qquad (4.30)$$

Where $$\rho_r(x) = -\max_{w \in W}\{TC(x,w) - \min_z TC(z,w)\} \qquad (4.31)$$

$$TC(x,w) = \sum_{i=1}^{n} w_i d_i(x) \qquad (4.32)$$

$$d_i(x) = \sqrt{((x_1 - p_{i1})^2 + (x_2 - p_{i2})^2)} \qquad (4.33)$$

Subject to:

$$x, z \in V$$

4.5.2 Minmax regret p-centre location on a network with demand uncertainty (1998)

In the article ´minmax regret p-centre location on a network with demand uncertainty´, Igor Averbakh and Oded Berman, demand is uncertain and no assumptions about the probability distribution of demand can be made. Only the upper and lower bounds of the demand are known. The aim of the model is to find the minimal maximum regret solution, which is to minimize the worst case opportunity loss (Averbakh and Berman, 1998).
The feasible area of this model is a network. In formula it is represented by the maximum regret robustness model described by Equations (4.30) to (4.33). That model is already illustrated in Section 3.3.5 by use of the small numerical example of Section 3.2.

4.5.3 Facility location problems with uncertainty on the plane (2005)

In ´facility location problems with uncertainty on the plane´ of Igor Averbakh and Sergei Bereg, the 1-median and 1-center problem on a plane are considered in a minimize maximum regret concept. Their focus is on the introduction of algorithms for solving these problems (Averbakh and Bereg, 2005).
This model is precisely represented by the standard maximum regret robustness model described by Equations (4.30) to (4.33). The model is already illustrated in Section 3.3.5 by use of the small numerical example of Section 3.2.

4.5.4 The multi-criteria minisum location problem (2001)

In the article ´Robustness in the pareto-solutions for the multi-criteria minisum location problem´ of Francisco Fernandez, Justo Puerto, Stefan Nickel and Antonio Rodriguez-Chia, the maximum regret concept is used in a bi-criteria objective function. A large company is considered to have decision makers for several departments. The interest of a new facility location, may differ for each department. The article of Fernandez et al., 2001, considers two decision makers which consider their own set of weights. The bi-criteria model of the article minimizes the maximum regret of both weight sets for the placement of the new facility (Fernandez et al, 2001).

The bi-criteria regret location problem uses for each decision maker, one weight set: $w^1 \in W^1$ for decision maker 1 and $w^2 \in W^2$ for decision maker 2. Equation (4.31) is replaced by Equation (4.34). The negative signs are used to model the inverse relation between the maximum regret and robustness. For both decision makers, the optimal facility location is not per definition the same, z^1 represents the optimum location for decision maker 1 for a certain weight and z^2 for decision maker 2.

$$\rho_r(x) = -\max_{w^1 \in W^1}\left\{TC(x,w^1) - \min_{z^1} TC(z^1,w^1)\right\} - \max_{w^2 \in W^2}\left\{TC(x,w^2) - \min_{z^2} TC(z^2,w^2)\right\} \quad (4.34)$$

The small numerical example of Section 3.2 is used to illustrate this model. For this example, two weight ranges are considered. Decision maker 1 considers w^1 with demand ranges $1 \leq w_1 \leq 2.5$ and $2.5 \leq w_2 \leq 4$. Decision maker 2 considers w^2 with demand ranges $2.5 \leq w_1 \leq 4$ and $1 \leq w_2 \leq 2.5$. The left graph of Figure 4.6 represents feasible area 1 for w^1 and feasible area 2 for w^2.
To calculate the maximum regret for both decision makers for facility location *x=(3,3)*, the optimal location for both decision makers for their weight set is needed. As in feasible area 1, the demand of demand point 2 is always larger than the demand of demand point 1, the optimum location for decision maker 1 for all possible w^1 is on location z^1*=(3,1)*. As in feasible area 2, the demand of demand point 1 is always larger than the demand of demand point 2, the optimum location for decision maker 2 for all possible w^2 is on location z^2*=(1,4)*.

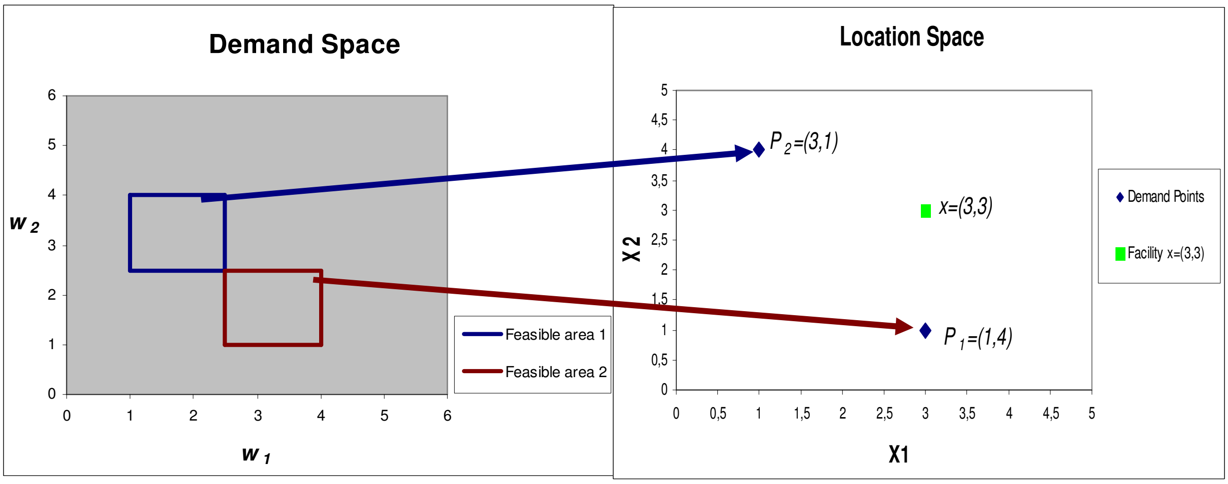

Figure 4.6: Bi-criteria maximum regret for the small numerical example

For this small example, the maximum regret occurs at maximum weight w^1*=(2.5,4)* and w^2*=(4,2.5)*.

$$\max_{w^1 \in W^1}\left\{TC(x,w^1) - \min_{z^1} TC(z^1,w^1)\right\} = 2.5 * d_1(x) + 4 * d_2(x) - 2.5 * d_1(z^1) + 4 * d_2(z^1) = 4.93$$

$$\max_{w^2 \in W^2}\left\{TC(x,w^2) - \min_{z^2} TC(z^2,w^2)\right\} = 4 * d_1(x) + 2{,}5 * d_2(x) - 4 * d_1(z^2) + 2{,}5 * d_2(z^2) = 4.58$$

The max regret robustness of *x=(3,3)* is:

$$\rho_r(3{,}3) = -4.93 - 4.58 = -9.51$$

5. A competitive deviation robustness model

In this chapter we apply the standard deviation robustness concept (Section 3.3.3) for a static competitive location field. Huff based location parameters (Section 2.6.1) are used in combination with the standard deviation model (Section 4.3.1) to create a competitive deviation robustness model.

Section 5.1 introduces the elements of the new model by explaining the above referred concepts. The new competitive deviation robustness model is formulated in Section 5.2.

5.1 Elements of the new model

In most cases, demand data for solving facility location problems are not known, only estimated. The ranges and probability distributions of the demand data are not known either. Even after intensive study the proposed ranges and or probability distributions are still arbitrary as the influence of all circumstances are hard to model and to predict.
The deviation robustness concept models this structure adequately as it does not need ranges nor probability distributions to model robustness for a facility location problem. The deviation robustness concept only needs a nominal value which represents the best guess possible for the demand. If the ranges or probability distribution are known and reliable, the nominal value is easily adapted for these values. For ranges, the nominal is the middle value and for the probability distribution the expected value (Snyder, 2006).

During this research, only one article has been found in literature, which uses the deviation robustness concept. In this article, ¨robust facility location¨ of Emilio Carrizosa and Stefan Nickel, robustness is the minimum deviation between the nominal vector for demand and vector *w*. Vector *w* represents a demand for which the total cost are above a predefined threshold, for example the budget of a company.

A competitive deviation robustness model is not found in literature, and is therefore developed and presented in the next section. The aim of this new deviation model is to take static competition into account.
As defined in Section 2.6.1, static competition is a competitive market situation were the competing facilities are already located in the feasible area. After the placement of the new facility, it is assumed that there will be no reaction from competitors or other change in the competitive market situation.
In Section 2.6.1. the Huff model is presented as Equation (2.4). The Huff model, calculates the market share captured by the new facility. This captured market share depends on the relative distance between new facility *x* and the demand points, and the competitors and the demand points.

In the new competitive deviation robustness model, robustness is related to the captured market share. The goal is to attract a minimum market share *T*. If the captured market share is lower than *T*, the new facility is not profitable. Robustness is here the ability of a facility location to capture the minimum market share, despite variations in demand.

5.2 The competitive deviation robustness model

The new Huff based competitive deviation robustness model is formulated as follows:

Indices:

i	index of demand points	*i=1,2,…,n*
j	index of competitors	*j=1,2,…,m*

Variables:

x	location variable for the new facility
w	demand deviating from nominal value v

Data:

A	Attractiveness of the new facility
α_j	Attractiveness of competitor *j*
p_i	location of demand point *i*
δ_{ij}	distance between demand point *i* and competitor *j*
T	Threshold value
V	feasible area for the facility location *x*
v_i	nominal demand value of city *i*

Objective function:

$$\underset{x}{Max}\{\rho_d(x)\} \quad (5.1)$$

Where $\rho_d(x) = \min_w \|w - v\|$ for which $M(x,w) \le T$ (5.2)

$$M(x,w) = \sum_{i=1}^{n} w_i * \frac{A * d_i(x)^{-\lambda}}{A * d_i(x)^{-\lambda} + \sum_{j=1}^{k} \alpha_j * \partial_{ij}^{-\lambda}} \quad (5.3)$$

$$d_i(x) = \sqrt{((x_1 - p_{i1})^2 + (x_2 - p_{i2})^2)} \quad (5.4)$$

Subject to:

$$x \in V$$

Robustness is the minimum deviation between nominal demand vector *v* and demand vector *w* for which the captured market share is below the threshold *T*, see Figure 5.1.

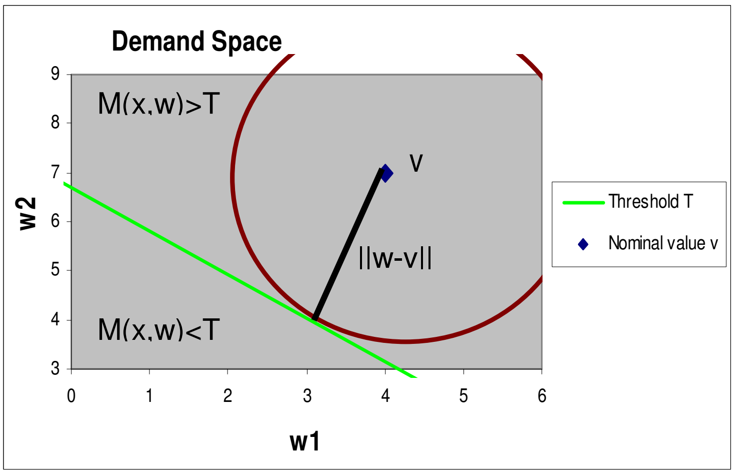

Figure 5.1: Deviation between the nominal demand and the market threshold

Facility location x, is only robust if the obtained market share under the nominal demand conditions is higher than the predefined threshold.

6. Algorithms for solving robust location problems

In order to find the optimum of a robust facility location model, four stochastic optimization algorithms are considered. An optimization algorithm is a description of steps, which results in finding an approximation of an optimum point. (Hendrix and Toth, 2009)
The following algorithms are introduced: Controlled Random Search, Genetic Algorithm, Simulated Annealing and Multi Start. The algorithms are presented in pseudo code, which is not the exact computer code but a readable representation. The exact code is used in the Matlab files, which are attached in the Appendices B.3 to B.6 for reproducibility purposes. These Matlab files include the settings of the algorithms which are discussed in Chapter 8.
The basics of every algorithm are explained and the relevant parameter settings are discussed. In Chapter 8, the potential of these algorithms will be investigated by use of two test cases.

6.1 Controlled Random Search

Controlled Random Search (CRS) is a simple population algorithm developed by W. Price in 1978 for global optimization. The CRS algorithm is a direct procedure (no gradients are involved) and is applicable to constrained and unconstrained optimization problems. The algorithm starts by generating a random uniform population in the feasible area. From the population, parent points are selected which are used to generate a new point (a trial point) in the area. If this trial point is in the feasible set, and if this point has a better function value than the worst point of the population, then the worst point of the population is replaced by the new point (Price, 1978 and Price, 1983).

For a maximization problem, Controlled Random Search works as follows:

ALGORITHM 6.1: Controlled Random Search Algorithm

Step 1: Generate a randomly uniform population *A* of size *N* on the feasible area *V.*
Step 2: Evaluate all points in population *A.*

Iteration process

Step 3: The lowest function value in *A* is denoted as *f(M)* with corresponding point *M*.
Step 4: Select randomly parent points for recombination from *A.*
Step 5: A new trial point (*T*) is generated from these parent points.
 *If *T* is in *V*, proceed to *step 6*,
 *If not, start a new iteration
Step 6: Evaluate *f(T).*
 *If *f(T)* is better then *f(M)*, proceed to *step 7*.
 *If not, start a new iteration
Step 7: Replace *M* and *f(M)* by *T* and *f(T)* in *A*.
 *If a stopping criterion is reached, proceed to *step 8*.
 *If not, start a new iteration
Step 8: The maximum function value in *A* is the optimum *f(O*)* with corresponding point *O**.

(Price, 1978 and Price, 1983)

In literature, many strategies are used for the generation of trial points (step 5 of the algorithm). The most common point generation is a linear combination of two or more parent points which are randomly drawn from array *A* (Kaelo and Ali, 2006 and Hendrix and Toth, 2009).

Two stopping criteria are very common for CRS in literature.

1. A fixed maximum number of iterations or function evaluations is often used as a standard criterion to compare different optimization methods or algorithm settings.
2. For optimization purposes, if the difference between the minimum and the maximum function value in *A* is below a predefined accuracy value alpha, the algorithm stops and the best solution found approximates the optimum.

(Hendrix et al., 2001 and Kaelo and Ali, 2006)

The exact algorithm code, used to test the controlled random search algorithm on all test cases, is presented in Appendix B.3.

6.2 Genetic Algorithm

The Genetic Algorithm (GA) is an optimization algorithm which tries to find the optimal solution of a problem using techniques inspired by evolutionary biology (Redondo, 2008).
Like CRS, the Genetic Algorithm is a population based algorithm which starts with a randomly chosen initial set of potential facility locations (the initial population) and calculates their function value (called fitness for GA). From the initial population, a set of parent points is selected which will be used to generate the trial points (offspring or children). This selection can be completely random or fitness based, where individuals with a better fitness value have more chance to be selected.
After the selection, the set of parents are used to produce offspring. This recombination can be inspired by many processes that are mimicked from evolutionary biology. Well known and often used recombination techniques in GA´s involve mutation, crossovers or plain inheritance.
After the generation of the trial points, the old population is replaced by the new offspring. This new population is the new initial population and this process can repeat itself till one of the stopping criteria is met. The individual with the best fitness value at the end approximates the optimum facility location (Barricelli, 1957 and Ortigosa, 2008).

In literature, many different variations on the Genetic Algorithm exist, which mostly differ in the way of parent selection or inheritance. The Genetic Algorithm which is used in this report makes use of elitist selection. The elitist selection strategy allows that some of the best individuals from the current generation, move unaltered to the next generation, see Algorithm 6.2.

For a maximization problem, the Genetic Algorithm works as follows:

ALGORITHM 6.2: Genetic Algorithm

Step 1: Generate a randomly uniform population *A* of size *N* on the feasible area *V*.
Step 2: Evaluate all individuals in population *A*.

Iteration process

Step 3: Select parents from population *A* based on their fitness
Step 4: Individuals in the current population with the best fitness are marked as elite.
Step 5: Produce children (*T*) from the parents via mutation (with *P=m*), crossover (with *P=c*) or plain inheritance (with *P=1-m-c*). Only feasible offspring is allowed.
Step 6: Replace population *A* by the elite and children to form a new generation.
Step 7: Evaluate the new population points and store them with their function values in *A*.
*If a stopping criterion is reached proceed to *step 8*,
*If not, start a new iteration
Step 8: The maximum value in *A* approximates the optimum *f(O*)* with corresponding point *O**.
(Matlab, 2008)

In CRS, a combination of population points is used in a straightforward way, leading to linear combinations of the parents. In GA the recombination of trial points is called inheritance. This inheritance can be straight forward as in CRS which mimics plain inheritance of the evolutionary biology. Besides this plain inheritance, GA uses other recombination techniques mimicked from evolutionary biology like mutations and crossovers. These mutations and crossovers have the advantage for the algorithm to possibly escape from local optima towards better optima as the generated trial points don't have to be linear combinations of the parent points. (Ortigosa, 2008 and Redondo, 2008)

The two most used stopping criteria of the genetic algorithm are the same as those of CRS.

1. A fixed maximum number of iterations or of function evaluations is often used as a standard criterion to compare different optimization methods or algorithm settings.
2. If the difference between the minimum and the maximum function value in *A* is below a predefined level alpha, the algorithm stops and the best solution found approximates the optimum.

(Hendrix et al., 2001 and Ortigosa, 2008)

To test all test cases of Chapter 7, the Genetic Algorithm of the optimtool of Matlab is used. In Appendix B.4 the used Matlab files included for all performed tests with the Genetic Algorithm.

6.3 Simulated Annealing

Simulated Annealing is a probabilistic optimization technique based on principles of thermodynamics used in metallurgy. In metallurgy, annealing is a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The crystals of a material consists of atoms which in their optimal uniform formation (with the lowest internal energy) make the material more solid and harder to break.

Annealing starts with heating the material to a high temperature. At this high temperature, atoms contain more energy which makes them capable to move from their initial positions (a place with a relative high internal energy) and wander trough space towards a better suited location (a location with lower internal energy). By slowly cooling the material, the energy of the atoms decreases which diminishes their ability to wander trough space towards better suited locations. If the temperature is low enough, the energy levels of the atoms is too low to move at all and the annealing process is stopped. *(Note: energy level and internal energy are two different attributes!)*

The Simulated Annealing algorithm mimics the annealing process. The atoms are represented by randomly chosen possible solutions and the objective value stands for the internal energy. The algorithm starts at a high temperature level, where the solutions (the atoms) contain much energy to move from their location towards other locations in the feasible area. As this energy level is high, the ability to wander trough space (the feasible area) and accept a worse state is big, which gives many opportunities to find remote locations with a lower function value. By cooling slowly, the energy level of the possible solutions decreases and their ability to find a position with a lower objective value decreases to a less remote region. The process stops when the temperature is too low for the possible solutions to wander through any region at all. From this moment on, the simulated annealing process resembles the hill climbing process. The final lowest solution found resembles the optimum, for a minimizing total cost problem (Metropolis et al., 1953).

ALGORITHM 6.3: Simulated Annealing

Step 1: Generate one randomly uniform starting point *M* on the feasible area *V*.
Step 2: Evaluate *f(M).*

Iteration process For every temperature *T* from high to low*,* perform a fixed number *n* of iterations

Step 3: Generate a randomly feasible neighbour point *L* of *M*.
Step 4: Evaluate *f(L).*
*If *f(M)* is better then *f(L)*, proceed to *step 5.*
*If not, proceed to *step 6*
Step 5: Generate a random uniform point *R,* between 0 and 1.
*If $R < \exp^{(-[L-M] / T)}$ go to *step 6*
*If not, start a *new iteration*
Step 6: New point *M* replaces point *L.*
*If a stopping criterion is reached, proceed to *step 7.*
*If not, start a *new iteration*
Step 7: The best *M* approximates the optimum found.

The Simulated Annealing algorithm attempts to permit small worse moves, while rejecting large ones. To succeed in this, the challenge lays in choosing the right parameter values for the cooling process. This starts by choosing an appropriate starting and end temperature, followed by the right method of temperature decrement and the number of iterations to perform at every temperature.

If the starting temperature is too high, almost all alternative solutions will be accepted and the process resembles for a long period of time a random search. If the starting temperature is too low, the ability to move further than the neighbourhood states is very low and the process will look like Hill Climbing as it will not succeed in escaping from a local optimum.

Zero is a usual end temperature. However, this can make the algorithm run a lot longer than necessary. In practise, it is not necessary to let the temperature decrease to zero because the chance of accepting a worse move at low temperatures is almost zero.

Literature states that enough iterations at each temperature should be allowed, to let the system stabilise at every temperature. Literature also states that to achieve this, the number of iterations at each temperature can be exponential to the problem size. Therefore a compromise is needed between the method of temperature decrement and the number of iterations to perform at every temperature. Either a large number of iterations can be done at a few temperatures, or a small number of iterations at many temperatures, or a balance between these two (Ortigosa, 2008).

In Appendix B.5 all Matlab files are presented which used the SA optimtool settings of Matlab.

6.4 Multi Start

The Multi Start algorithm is an extension of a local search algorithm. A local search algorithm is an optimization method which goes from a starting point to a local optimum in the neighbourhood. The Multi Start algorithm generates *L* random uniform starting points in the feasible area which individually follow a local search algorithm to generate a local optimum. The best local optimum found approximates the global optimum. The higher the number of starting points, the bigger the chance to find the global optimum. As the generation of starting points is chance based, there is no guarantee that the optimum found is the global optimum (Ortigosa, 2008 and Redondo, 2008).

For the Multi Start algorithm, many local search algorithms can be used. In this report, the fmincon optimizer of Matlab is used. Fmincon uses information of the Hessian, a square matrix of second order partial derivatives. The Hessian describes the curvature of a function, and contains information about the direction in which the improvement of the objective function is the largest.
Fmincon of Matlab contains several algorithms, which all three handles the Hessians differently. The standard algorithm which fmincon uses is called ¨trust region reflective¨. By running the Multi Start files with this algorithm, a warning occurs:
Warning: Trust-region-reflective method does not currently solve this type of problem, using active-set (line search) instead.
Therefore all Multi Start Matlab files are modified to use the ¨active-set¨ algorithm as the standard algorithm. The fmincon ¨active-set¨ algorithm is used as black box in the Multi Start algorithm (Matlab, 2008).

ALGORITHM 6.4: Multi Start

Step 1: Generate *L* random uniform initial starting solutions in the feasible area *V*.
Step 2: Perform the fmincon ¨active-set¨ algorithm on all initial points (the Black Box)
Step 3: The best value of fmincon approximates the optimum f_O* with corresponding point *O**.

The Multi Start Matlab files which are used to test all test problems, are added to this report in Appendix B.6.

The next chapter introduces 3 test cases. The first two test cases are used to test and compare the algorithms of this chapter on their ability to solve robust facility location models. The third test case is used to illustrate the behaviour of the new competitive deviation model of Chapter 5.

7. Test examples

This chapter contains three cases which all work with the same feasible area and demand points described in Section 7.1.

Test case 1 introduces two p-centre problems, see Section 7.2. Test case 2 introduces two robust p-centre problems, see Section 7.3. Test case 3, is a competitive robustness test case, used for illustration, see Section 7.4.

7.1 The feasible area

The test models are situated in a continuous field *V* of size 20 by 20. In this field, 21 demand points are located *(i=1,..,21)*. The list of demand points can be found in Appendix A.1 (matrix P). Figure 7.1 gives a representation of the continuous field and the location of the demand points. The location of each demand point is denoted by p_i. P_{i1} and P_{i2} represent the first and second coordinates of each demand point *j*.

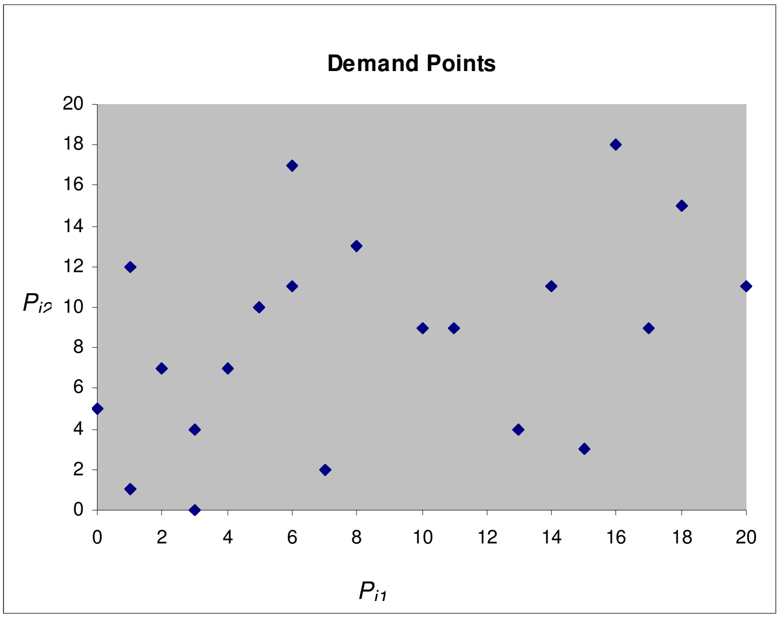

Figure 7.1: Demand points of the test case in the area V

All demand points have a certain demand weight w_i, which represents the amount of a product they can buy.

In the test cases of Section 7.2, the objective is to minimize the maximum distance between a demand point and its closest facility for the placement of *p* facilities. This test case is tested for the placement of *p=2* and *p=3* facilities. In this case the demand is considered to be constant and known. Appendix A.2 contains the weights in vector *w*.

In Section 7.3, demand is considered to be constant. A time period of 100 time units (days, weeks or months) is considered in which the demand fluctuates. Appendix A.3 contains the weight scenario matrix *W*, representing the weights for all the demand points per scenario.

In the illustration case of Section 7.4, a nominal demand is considered. Besides the 21 demand points, competitors will be located in the feasible area. The goal is to find the optimal deviation robustness location for one new facility, for obtaining a market share of at least 20%.

7.2 Test Case 1: The p-centre problems

Model described by Equations (2.11) to (2.16) of Section 2.8.2, describes the general *p*-centre problem for a discrete set of candidate points. The two *p*-centre problems of this test case aim to locate *p* facilities in a continuous field.

The first *p*-centre problem is a 2-centre problem. This means that in the feasible area of Figure 7.1, two facilities have to be placed. Facility one is represented by *x* with coordinates x_1 and x_2. Facility two is represented by *y* with coordinates y_1 and y_2.
The goal of the 2-centre problem is to locate *x* and *y* in such a way that the largest transport costs between the demand points and their closest facility is at a minimum level. The Euclidian distance between a demand point and its closest facility will be represented by where *i* represents the demand point.

$$d_i(x, y) = \min\{\|p_i - x\|, \|p_i - y\|\} \tag{7.1}$$

The objective is given by Equation (7.2):

$$\min_{x,y}\{\max_i w_i * d_i(x, y)\} \tag{7.2}$$

In this 2-centre problem, the demand w_j of each demand point *i*, is considered to be constant and no capacity constraints on the facilities or whatsoever are taken into account.

The second *p*-centre problem is a 3-centre problem. The only difference with the 2-centre problem is that besides the placement of *x* and *y* also a third facility *z*, with z_1 and z_2 as coordinates, is to be placed in the same feasible area. The goal remains to minimize the largest transport cost of all demand points to their closest facility. The formulas are therefore quite similar for the calculations of the distance and for the objective function, see Equations (7.3) and (7.4).

$$d_i(x, y, z) = \min\{\|p_i - x\|, \|p_i - y\|, \|p_i - z\|\} \tag{7.3}$$

$$\min_{x,y,z}\{\max_i w_i * d_i(x, y, z)\} \tag{7.4}$$

Also for the 3-centre problem, the demand w_j of each demand point *i*, is considered to be constant, and no capacity constraints on the facilities or what so ever are taken into account.

7.3 Test Case 2: The robust p-centre problems

The robust *p*-centre problem, deals also with the placement of *p* facilities. but in the robust case, demand is not considered to be constant. The variance in demand is represented by 100 scenarios in which the demand w_i of every demand point differs per scenario *j*. Robustness is here defined as the ability to deliver in time. Therefore a threshold value *T* (or service indicator) is introduced, which represents the maximum allowed distance between all demand points and their closest facility. If the distance of one or more demand points to its closest facility exceeds this threshold, the system fails to deliver in time. Maximum robustness is obtained by locating *p* facilities in such a way, that a minimum number of scenarios fail to deliver in time.

In the Matlab files, robustness is counted in a reverse way. For every scenario in which the chosen *p* facility locations fail to deliver on time, a penalty point is given. The minimum robustness score is therefore 100, if the chosen *p* facility locations fail for all scenarios to deliver on time. Maximum robustness is 0, wherein the chosen *p* facility locations does not fail for any scenario to deliver in time.
Two robust *p*-centre problems are tested, namely the robust *2*-centre and the robust *3*-centre problem, which deals respectively with the placement of two and three facilities in the feasible area *V*. For both cases the threshold value *T* is 50.

The objective function of the robust *2*-centre problem is represented by Equations (7.5) to (7.7):

$$\max_{x,y}\{R(x,y)\} \quad (7.5)$$

Where: $$R(x,y)=\sum_{j=1}^{m}\{F(x,y,w_j)\} \quad (7.6)$$

Where: $$F(x,y,w_j)=\begin{cases}0 & if \quad \max_i\{w_{ij} * d_i(x,y)\}\leq T\\ 1 & if \quad \max_i\{w_{ij} * d_i(x,y)\}> T\end{cases} \quad \textit{for all } j \quad (7.7)$$

As the robustness measure is a numerical counter, the objective function has stepwise changes in natural numbers from 0 to 100. This results in a plateau wise shape of the objective function. To gain more insight in this plateau shaped objective function a grid search is performed for the robust *2*-centre problem in which facility *y* is fixed at coordinates (*12,15)*, see Figure 7.2.

A grid search is an algorithm which evaluates the function value by investigating systematically all feasible locations of *x* over the grid. In this case a grid is chosen with grid size 1. This grid size is the distance between two evaluated points. In this case, the feasible area is from 0 to 20 in two dimensions, this results in ($[20+1]^2$=) 441 points evenly spread over the feasible area, from (*0,0)* to (*20,20*). The first point is (*0,0)*, the second point will be (*0,1)* and the last point is (*20,20)*. The grid search is performed in Matlab, the according M-file can be found in Appendix B.2.1 (Matlab, 2008).

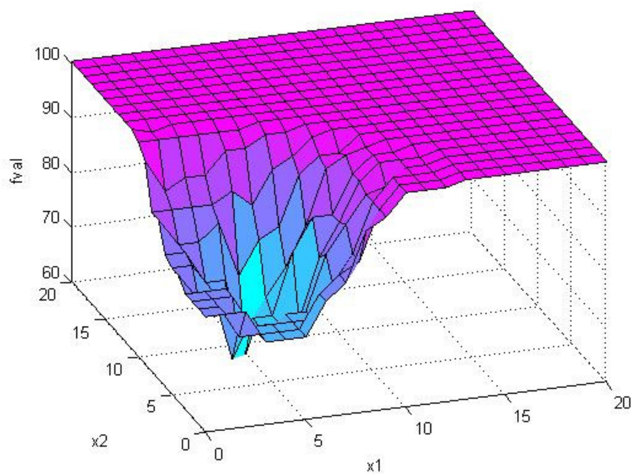

Figure 7.2: Grid search of the robust 2-centre problem with fixed facility y on (12,15)

Because there is no difference between facilities *x* and *y*, symmetric optimal solutions can occur. For example, if facility *x* and *y* are switched, both facility locations are different but the function evaluation and the practical implementation would be identical. To prevent these symmetric solutions, an adjustment is made in the Matlab work files. For all solutions, the first coordinate of *x* must be smaller than the first coordinate of *y*.

The robust 3-centre problem is quite similar to the robust 2-centre problem. The objective function of the robust 3-centre problem is represented by Equations (7.8) to (7.10):

$$\max_{x,y,z}\{R(x,y,z)\} \quad (7.8)$$

Where: $$R(x,y,z)=\sum_{j=1}^{m}\{F(x,y,z,w_j)\} \quad (7.9)$$

Where: $$F(x,y,z,w_j)=\begin{cases}0 & if \quad \max_i\{w_{ij} * d_i(x,y,z)\}\leq T\\ 1 & if \quad \max_i\{w_{ij} * d_i(x,y,z)\}> T\end{cases} \quad \textit{for all } j \quad (7.10)$$

Also for the robust 3-centre problem a grid search is performed, in this case facility *y* and *z* are fixed. The coordinates of *y* are (*5,10)*, the coordinates of *z* are (*15,15)*. The results of this grid search is shown in Figure 7.3. This grid search is performed in Matlab, the according Matlab file can be found in Appendix B.2.2.

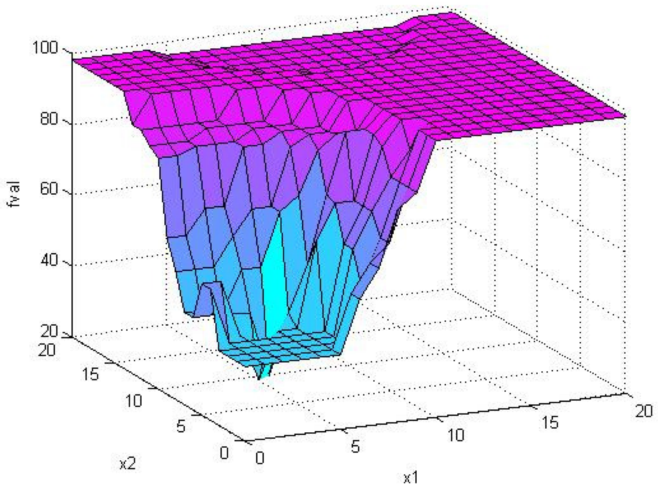

Figure 7.3: Grid search of the robust 2-centre problem with fixed facility y on (5,10) and facility z on (15,15)

To prevent symmetric solutions in the robust 3-centre problem, an extra adjustment is made in the Matlab work files. For all solutions, the first coordinate of *x* must be smaller than the first coordinate of *y*, which in turn must be smaller then the first coordinate of *z*.

7.4 Illustration Case: Robust 1-median problem with competition

The illustration case is a small adaptation of Test Case 1 of Section 7.2. The feasible area and the 21 demand points are unchanged, and the demand vector *w* is used as nominal demand vector *v*. Additional to this, competitors are located in the feasible area.
The model is tested for placing one facility in area *V*, for two competitive situations. In the first situation, two competitors are present, $c_1=(3.6,3.79)$ and $c_2=(15.41,13.96)$, the green squares in Figure 7.4. The locations of these competitors represent the optimum for the 2-centre problem of Test Case 1.
In the second situation, three competitors are present, $c_1=(2.14,2.02)$, $c_2=(10.32,8.24)$ and $c_3=(17.74,15.14)$, the pink triangles in Figure 7.4. These locations represent the optimum for the 3-centre problem of Test Case 1.

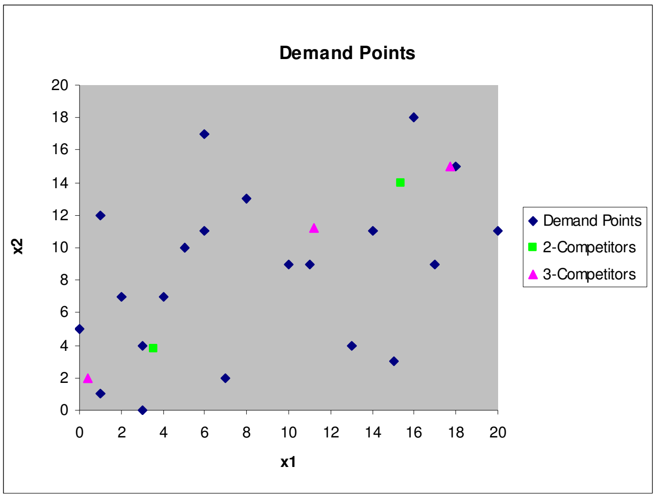

Figure 7.4: Illustration case with competitors

For both illustration problems, the goal of the facility is to obtain a market share of at least *T=20* demand units. The objective is to maximize the deviation robustness of one new facility in both competitive market situations.

This deviation robustness is the minimum distance in the 21-dimensional weight space between the nominal weight vector *v* and a weight vector *w*, for which the market share is smaller or equal than the required market share threshold. The market share is calculated according to the Huff model of Section 2.6.1. For a given *x*, the market share that *x* obtains for each demand point *i* is represented by $M_i(x)$.

$$M_i(x) = \frac{A * d_i(x)^{-\lambda}}{A * d_i(x)^{-\lambda} + \sum_{j=1}^{k} \alpha_j * \partial_{ij}^{-\lambda}} \tag{7.11}$$

The robustness of the model for a new facility location *x* is

$$\rho_d(x) = \min_{w \in W} \|w - v\| \quad \text{with} \quad \sum_{i=1}^{n} w_i M_i(x) \le T \tag{7.12}$$

The optimum of Equation (7.12) can be shown to be

$$\rho_d(x) = \frac{S}{\|M(x)\|} \qquad \text{with} \qquad S = \sum_{i=1}^{n} v_i * M_i(x) - T \tag{7.13}$$

, with an appropriate norm.

S is here the slack market share, which is the difference between the threshold market share and the market share captured for the nominal demand.

As presented in Section 3.3.3, the distance between *w*, and the nominal *v*, can be calculated with several norms. To calculate norm ||*v-w*|| for this illustration model, norm ||*M(x)*|| is needed. These two norms are not the same, but opposite form each other. To calculate the 1-norm of *v-w*, *S* must be divided by the infinite-norm of *M(x)*, and vice versa. For the Euclidian norm, the inverse is identical. In other words, to calculate the 2-norm of *v-w*, *S* must be divided by the 2-norm of *M(x)*. In Equations (7.14) to (7.16) the 3-norms of robustness are presented.

The minimum 1-norm distance between *w* and *v*, is calculated by dividing *S* by the infinite-norm of *M(x)*. Equation (7.14) calculates the 1-norm of Equation (7.13),

$$\rho_{d1}(x) = \|v - w\|_1 = \frac{S}{\|M(x)\|_\infty} = \frac{S}{\max_i |M_i(x)|} \tag{7.14}$$

The minimum 2-norm distance between *w* and *v*, is calculated by dividing *S* by the 2-norm of *M(x)*. Equation (7.15) calculates the 2-norm of Equation (7.13).

$$\rho_{d2}(x) = \|v - w\|_2 = \frac{S}{\|M(x)\|_2} = \frac{S}{\sqrt{\sum_{i=1}^{n} (M_i(x))^2}} \tag{7.15}$$

The infinite-norm distance between *w* and *v*, is calculated by dividing *S* by the 1-norm of *M(x)*. Equation (7.16) calculates the infinite-norm of Equation (7.13).

$$\rho_{d\infty}(x) = \|v - w\|_\infty = \frac{S}{\|M(x)\|_1} = \frac{S}{\sum_{i=1}^{n} |M_i(x)|} \tag{7.16}$$

8. Stochastic analysis at a robust location test case

In this chapter, the efficiency and effectively of the algorithms of Chapter 6 are analysed via two test cases on certain test criteria. The goal of this chapter is to select the best algorithm of Chapter 6 to find the optimal solution for robust and non robust facility location models.

In Section 8.1, the criteria for comparing the four algorithms of Chapter 6 are discussed. In Section 8.2 all algorithms are tested on Test Case 1 and Test Case 2. Test Case 1 is presented in Section 7.2 and Test Case 2 in Section 7.3. All algorithms are tested using their standard settings. These standard settings are the setting which are obtained from the original Matlab file which is used and edited for this test case. Only one setting is modified, which will result in an equal number of function evaluations for comparison reasons. In Section 8.3, the four algorithms are compared on their test results.

For all tests Matlab version 7.6.0 is used, release 2008(a). All tests are performed on a personal computer with an Intel Core 2 Duo processor and 1024 MB memory (Hewlet Packerd, 2007).

8.1 Test criteria, effectiveness and efficiency

The most important criteria of the algorithm is its effectiveness. Effectiveness is the ability of the algorithm to find a good solution. As the goal of all cases is minimization (minimizing the maximum distance to the closest facility or minimizing the number of scenarios that fail to remain below the threshold), effectiveness is the ability of an algorithm to find a solution as low as possible.
As the four algorithms are stochastic, the outcome of one test is not necessarily representative for the ability of the algorithms. Therefore, all cases will be run 100 times for every algorithm. The average of the function values of these 100 runs and the variance are more representative for the ability of each algorithm. The lower the average of these 100 runs, the better the algorithm performs. The lower the variance, the better the average represents the ability of the algorithm.

Besides effectiveness, the efficiency of an algorithm is an important criterion. Efficiency is the effort an algorithm has to take to come to its optimum. This efficiency can be measured in convergence speed, in memory requirements or in number of function evaluations. In this case the efficiency of the algorithm is not the most important issue and therefore is chosen to make efficiency a circumstance or restriction in stead of a test criterion. For all test cases and all algorithms the average number of function evaluations of the 100 runs may not exceed 2700 function evaluations.

8.2 Algorithm settings and results for Test Cases 1 and 2

In this section, Controlled Random Search, Genetic Algorithm, Simulated Annealing and Multi Start are tested on Test Case 1 and 2 of Chapter 7. Every algorithm is run 100 times with approximately 2600 function evaluations on average. The anti-symmetric measures described in Section 7.3 are taken for all algorithms and for all tested problems.
In Section 8.2.1 to 8.2.4. the chosen parameter settings are elaborated, and the outcome locations are graphically presented per algorithm. In Section 8.3 the final results are given in a table on which the conclusion is based.
All Matlab work files that are used to process these results are added in the Appendices B.3 to B.6, the test results in Appendices C.1 to C.4.

8.2.1 Controlled Random Search settings and results

The Controlled Random Search method is the first algorithm and the corresponding Matlab files can be found in Appendix B.3, the tables with results in Appendix C.1.
For the Controlled Random Search algorithm, a Matlab file of E.M.T. Hendrix is used (Hendrix and Toth, 2009). This Matlab file is adapted to solve the problems and the anti-symmetric configuration is implemented.

The Controlled Random Search algorithm is performed with a starting population *M* of 50 individuals. The stopping criteria accuracy alpha is set on 0.05 which means that, if the function evaluation of the best and the worst individual of the population differs less than 0.05 the algorithm stops.

To limit the number of function evaluation to an average close to 2700 function evaluations per run, a limit is set on 2700 function evaluations in all control random search Matlab files. If the function values of the total population is converged within alpha, the number of function evaluations will be lower.

Figure 8.1 shows the results of the CRS algorithm on the *p*-centre problems. The left graph presents the results of the 2-centre problem. These results are obtained with exactly 2700 function evaluations as the population did not convergence before it. The optimal solution was a maximum distance of 32.9303 by the facility locations *x=(2.61,4.19)* and *y=(15.36,13.93).*
The right graph presents the results of the 3-centre problem. On average, the number of function evaluations was 2240.91 per run and the optimal solution was a maximum distance of 27.6430 for facility locations, *x=(4.09,2.22)*, *y=(7.61,8.27)* and *z=(17.73,15.01)*.

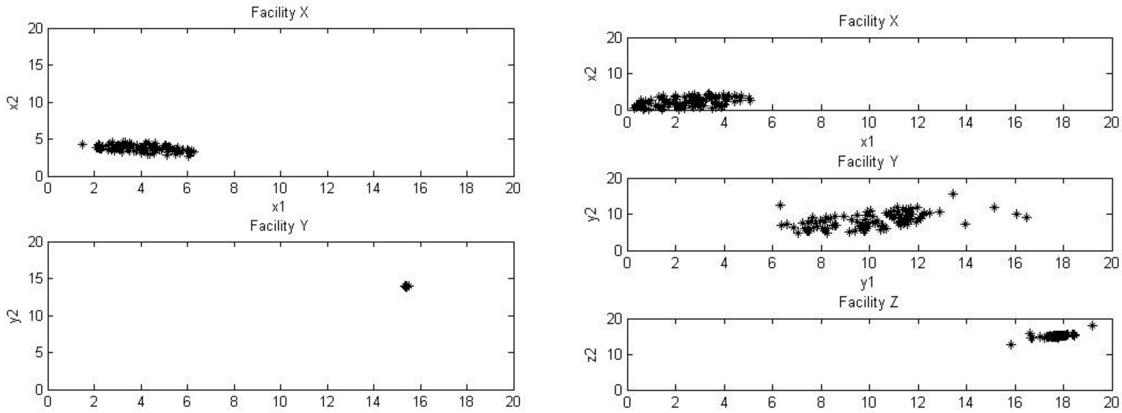

Figure 8.1: Controlled Random Search facility location outcomes for the p-centre problems

Figure 8.2 shows the results of the CRS algorithm on the robust *p*-centre problems. The left graph presents the results of the robust *2*-centre problem. The average function evaluations were 980 per run and the optimal robustness found was 2, which means that the optimum location of *x* and *y* resulted in a failure for two scenarios. This optimum was found in 36 of the 100 runs. In all 64 other runs the optimum was 3. The average position of facility *x* by the 36 times that the optimum was 2 was (*5.05,4.11)*, the average position of *y* was (*14.75,13.25)*.

The right graph presents the results of the robust *3*-centre problem. The average function evaluations were 814 per run and the optimal robustness was here 0. By locating three facilities the controlled random search algorithm found locations for x, y and z where none of the 100 scenarios failed. Of the 100 runs, this was the case for 51 runs. For the other 49 runs, 1 one of the scenarios failed for the optimum location. The average position of facility *x* by the 51 times that the optimum was 0 was (*3.22,5.63)*, the average position of *y* was (*9,85,14.69)* and for *z (16.24,10.74)*.

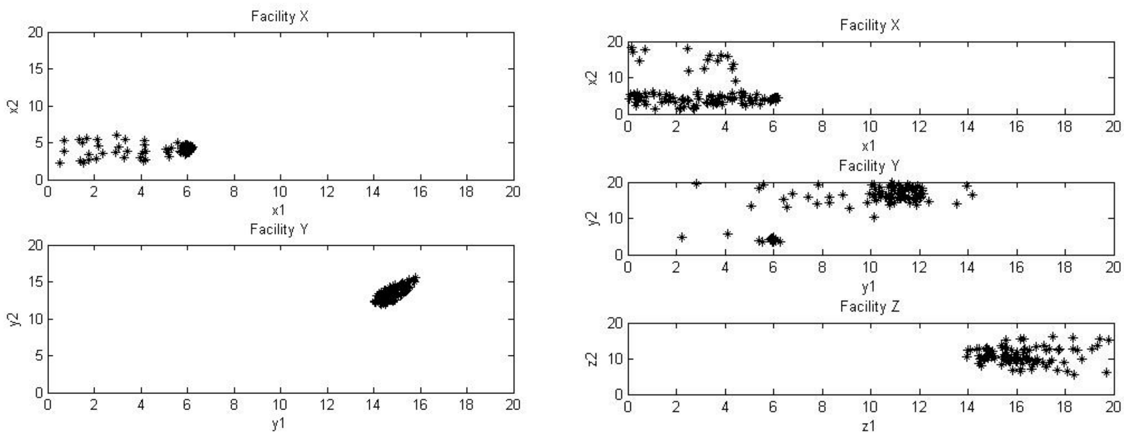

Figure 8.2: Controlled Random Search facility location outputs for the robust p-centre problems

8.2.2 Genetic Algorithm standard settings

The Matlab files of the Genetic Algorithm can be found in Appendix B.4, the tables with results in Appendix C.2.
For the Genetic Algorithm Matlab file, the automatic generated GA Matlab files are used of the optimtool-box of Matlab. By using the optimtool, you can select GA and adapt the settings to your problem. After adapting the settings you can auto-generate a Matlab file.

The population size *M* of the Genetic Algorithm is 50. The algorithm is stopped after 51 iterations which results for all problems in a total of approximately 2600 function evaluations ([51 iterations + 1 starting population=] 52*M).

Figure 8.3 shows the results of the Genetic Algorithm on the *p*-centre problems. The left graph presents the results of the *2*-centre problem. The optimal solution of the 100 runs was a maximum distance of 32.9278 by the facility locations of *x=(2.41,4.44)* and *y=(15.36,13.93)*.
The right graph presents the results of the *3*-centre problem. The optimal solution of the 100 runs was a maximum distance of 27.6420 by the facility locations of *x=(3.38,0.71)*, *y=(10.53,6.63)* and *z=(17.71,15.00)*.

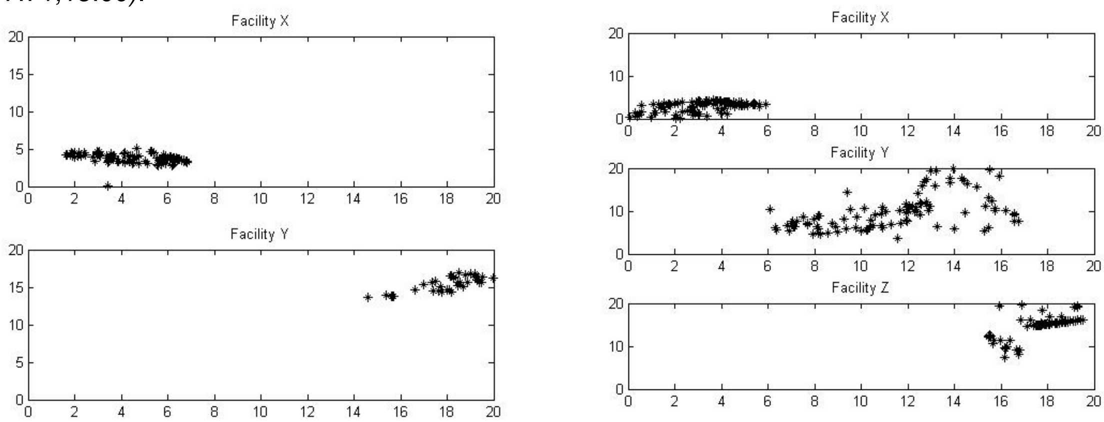

Figure 8.3: Genetic Algorithm facility location outcomes for the p-centre problems

Figure 8.4 shows the results of the Genetic Algorithm on the robust *p*-centre problems. The left graph presents the results of the robust 2-centre problem. The optimal robustness found was 2, which means that the optimum location of *x* and *y* resulted in a failure for two scenarios. This optimum was found in 10 of the 100 runs. In the other 90 runs the optimum was 3 or much higher with an optimum of 51 for the worst run.
The right graph presents the results of the robust 3-centre problem. The optimal robustness found for this problem is 0. By locating three facilities the Genetic Algorithm found locations for *x*, *y* and *z* where none of the 100 scenarios failed. Of the 100 runs, this was the case for 28 runs. For four runs this was 2 and for the other 68 runs, 1 scenario failed for the optimum location.

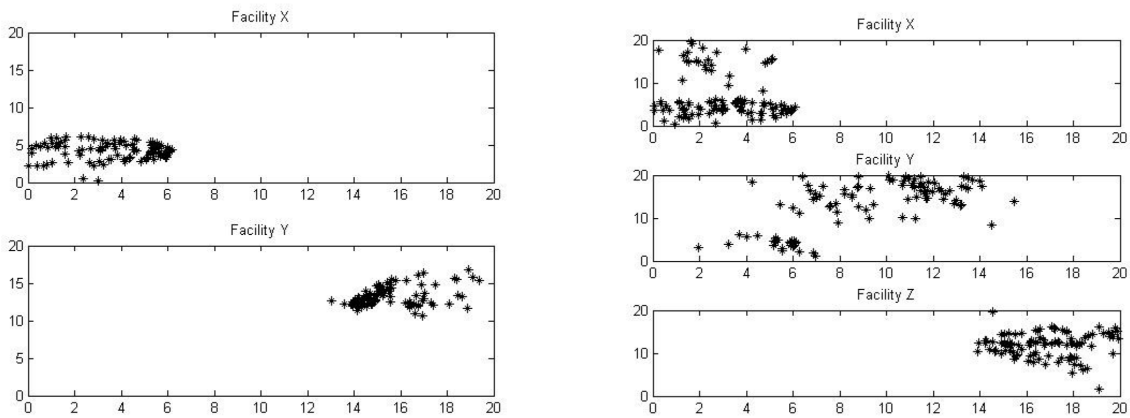

Figure 8.4: Genetic Algorithm facility location outputs for the robust p-centre problems

8.2.3 Simulated Annealing standard settings

The Matlab files of the Simulated Annealing algorithm can be found in Appendix B.5, the tables with results in Appendix C.3.

For the Simulated Annealing Matlab file, the automatic generated SA Matlab files are used of the optimtool-box of Matlab. By using the optimtool, you can select SA and adapt the settings to your problem. After adapting the settings you can auto-generate a Matlab file. The maximum limit on the number of function evaluation is set on 2700.

Figure 8.5 shows the starting points and the results of the Simulated Annealing algorithm for the 2-centre problem. For each run of the Simulated Annealing algorithm a random starting point is generated. The 100 starting points of the 100 runs are shown in the left graph.
The right graph presents the outcome facility locations. The optimal solution of the 100 runs was a maximum distance of 32.9278 by the facility locations of *x=(3.55,3.82)* and *y=(15.36,13.93)*.

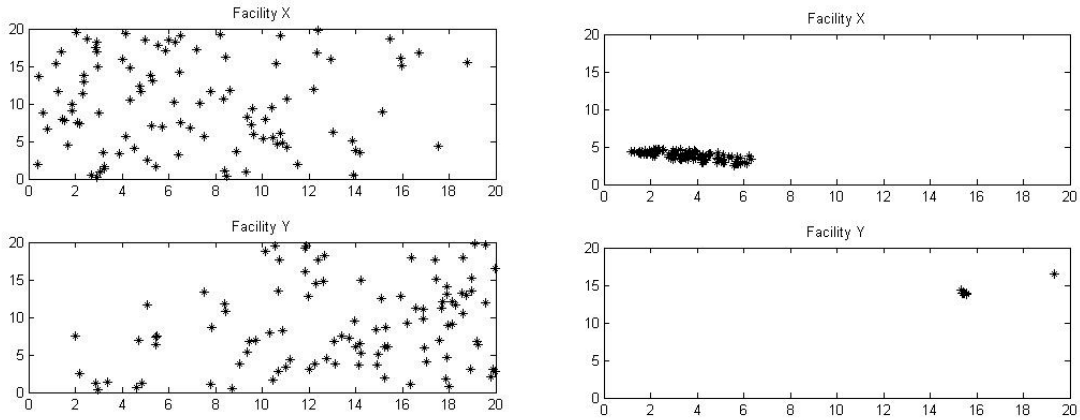

Figure 8.5: Simulated Annealing facility location starting points and outputs for the 2-centre problem

Figure 8.6 shows the starting points and the results of the Simulated Annealing algorithm for the 3-centre problem. The 100 starting points of the 100 runs are shown in the left graph.
The right graph presents the outcome facility locations. The optimal solution of the 100 runs was a maximum distance of 27.6420 by the facility locations of *x=(0.39,2.00)*, y=(*11.22,11.22)* and *z=(17.71,15.00)*.

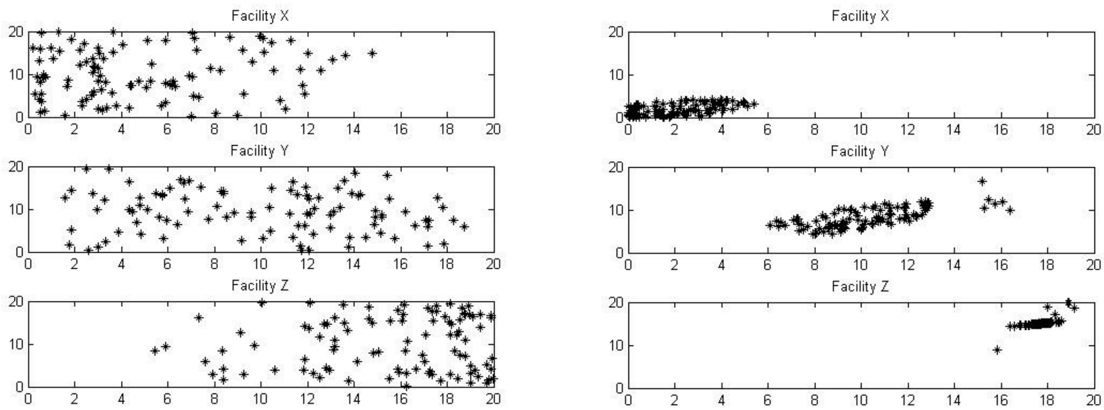

Figure 8.6: Simulated Annealing facility location starting points and outputs for the 3-centre problem

Figure 8.7 shows the starting points and the results of the Simulated Annealing algorithm for the robust 2-centre problem. The 100 starting points of the 100 runs are shown in the left graph.
The right graph presents the outcome facility locations. The optimal robustness found was 2, which means that the optimum location of *x* and *y* resulted in a failure for two scenarios. This optimum was found in 65 of the 100 runs. In the other 35 runs the optimum was 3.

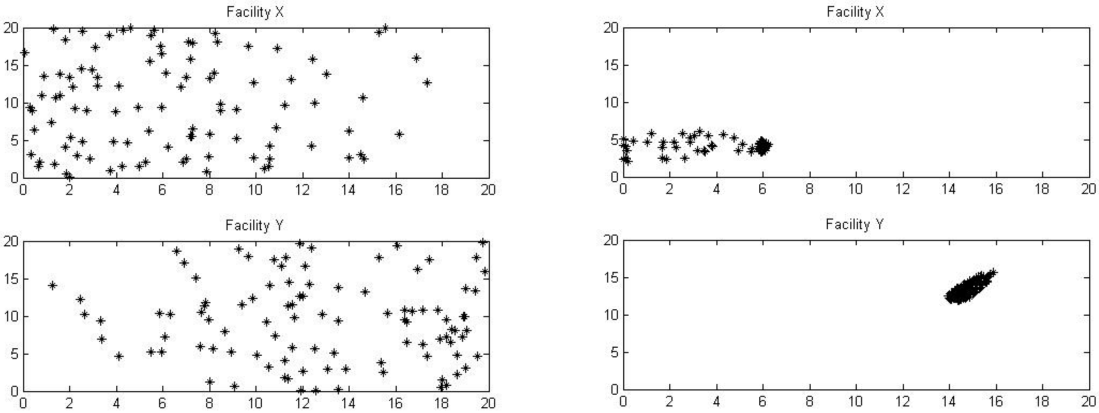

Figure 8.7: Simulated Annealing facility location starting points and outputs for the robust 2-centre problem

Figure 8.8 shows the starting points and the results of the Simulated Annealing algorithm for the robust 3-centre problem. The 100 starting points of the 100 runs are shown in the left graph.
The right graph presents the outcome facility locations. The optimal robustness found was 0, which means that the optimum location of *x*, *y* and *z* resulted in a failure for zero of the 100 scenarios. This optimum was found in 94 of the 100 runs. In the other 6 runs the optimum was 1.

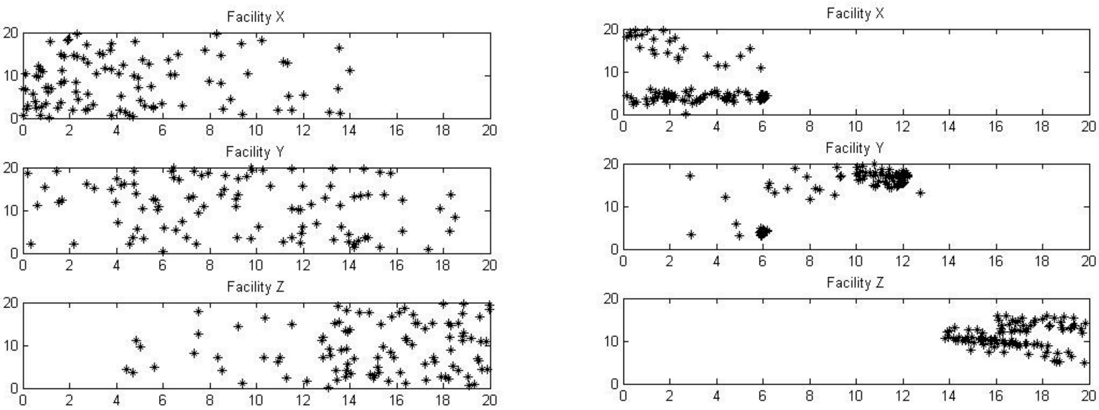

Figure 8.8: Simulated Annealing facility location starting points and outputs for the robust 3-centre problem

8.2.4 Multi Start standard settings

The Matlab files of the Multi Start algorithm can be found in Appendix B.6, the tables with results in Appendix C.4.

For the Multi Start Matlab file, the automatic generated fmincon Matlab files are used of the optimtool-box of Matlab. By using the optimtool, you can select fmincon and adapt the settings to your problem. After adapting the settings you can auto-generate a Matlab file. Fmincon is a single local optimizer. The fmincon Matlab file is edited to run with 50 starting points.

Figure 8.9 shows the results of the Multi Start algorithm for the 2-centre problem and the 3-centre problem. For the 2-centre problem, the Multi Start algorithm used on average 2728.62 function evaluations. The optimum solution of the 100 runs was a maximum distance of 32.9278 by facility location *x=(3.06,3.6)* and *y=(15.36,13.93)*.
For the 3-centre problem, the Multi Start algorithm used on average 2737.56 function evaluations. The optimum solution of the 100 runs was a maximum distance of 27.6420 by facility locations *x=(3.8,1.22)*, *y=(7.96,7.38)* and *z=(17.71,15)*.

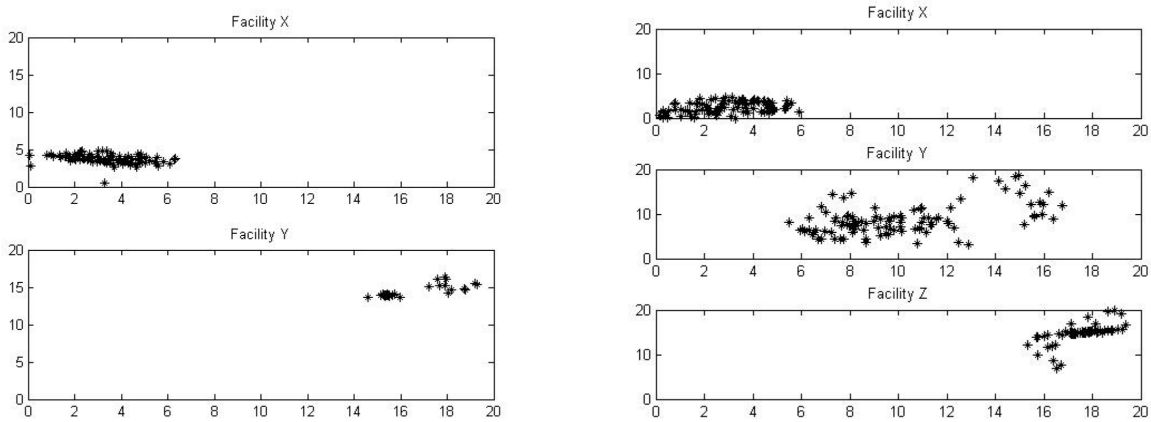

Figure 8.9: Multi Start facility location outputs for the p-centre problems

For the robust 2- and 3- centre problem the Multi Start algorithm did not perform as it got stuck on the plateau issues of the test case described in Test Case 2, Section 7.3.

8.3 Conclusions algorithm testing

In this subsection, the results of all algorithms are presented and compared to identify which algorithm is most suitable to solve and illustrate robust location models. Tables 8.1 and 8.2 contain the results of the p-centre problems of Test Case 1, Section 7.2. In both cases, the CRS algorithm is the only algorithm which did not find the exact optimum locations, although it does hardly differ. CRS has the lowest variance in the 2 centre problem. Simulated Annealing has the lowest average and the lowest maximum in both cases, and the lowest variance in the three centre problem. The Multi Start algorithm has the worst maximum and average in both cases, while the genetic algorithm has the worst variance in both cases.

Table 8.1: Results of the 2-centre problem for all algorithms

	Average value over 100 runs	**Minimum** value over 100 runs	**Maximum** value over 100 runs	**Variance** over 100 runs	Average **number** of function evaluations
CRS	32,9994	32,9293	33,2507	0,0046	2700,00
GA	33,7583	32,9278	37,3279	1,3938	2628,00
SA	32,9953	32,9278	35,1277	0,0773	2700,00
MS	33,4572	32,9278	36,1269	0,8329	2728,62

Table 8.2 Results of the 3-centre problem for all algorithms

	Average value over 100 runs	**Minimum** value over 100 runs	**Maximum** value over 100 runs	**Variance** over 100 runs	Average **number** of function evaluations
CRS	28,2999	27,6495	31,1304	0,5467	2240,91
GA	28,5663	27,6420	32,5578	1,1012	2616,50
SA	27,9363	27,6420	29,4437	0,2233	2700,00
MS	29,1321	27,6420	32,7999	2,5823	2737,56

Tables 8.3 and 8.4 contain the results of the robust p-centre problems of Test Case 2, Section 7.3. Optim-count resembles the number of times that the optimum is found. The Multi Start algorithm did not work in these robust cases due to the explained scenario plateaus. The other three algorithms did all find an optimum solution in at least one of the runs for each problem. The Simulated Annealing algorithm performed the best in finding by far the most times the optimum in both cases.
The Genetic Algorithm has the worst variance, highest maximum and least number of times finding the optima in both robust p-centre problems. Simulated annealing and Controlled Random Search found the lowest maximum in both cases. The variance of these two algorithms was equal for the robust two centre problem. For the robust 3-centre problem, the variance of simulated annealing was better.
The Controlled Random Search algorithm converged before 2700 function evaluations for both cases.

Table 8.3 Results of the robust 2-centre problem for all algorithms

	Average value over 100 runs	**Minimum** value over 100 runs	**Maximum** value over 100 runs	**Var**	Average **number** of function evaluations	Optim **Count**
CRS	2,64	2,00	3,00	0,23	941,19	36
GA	13,99	2,00	51,00	162,15	2600,00	10
SA	2,35	2,00	3,00	0,23	2240,53	65

Table 8.4 Results of the robust 3-centre problem for all algorithms

	Average value over 100 runs	**Minimum** value over 100 runs	**Maximum** value over 100 runs	**Var**	Average **number** of function evaluations	Optim **Count**
CRS	0,23	0,00	1,00	0,18	814,12	51
GA	0,76	0,00	2,00	0,27	2600,00	28
SA	0,06	0,00	1,00	0,06	2700,00	94

The Simulated Annealing algorithm gives the best minimum in all four test problems. SA gives the lowest maximum in three of the four cases. SA gives the lowest variance in three of the four cases. SA finds the most often the optimum for the robust cases.
Therefore the Simulated Annealing algorithm is the best algorithm tested on Test Case 1 and 2.

9. Illustrating the competitive deviation robustness model

In this chapter, the new developed competitive deviation robustness model of Chapter 5 is illustrated by use of the illustration case of Section 7.4. The illustration case consists of two competitive robustness cases, one with 2 competitors and one with 3 competitors. Both these cases are illustrated for all three norms, discussed in Sections 3.3.3.1 to 3.3.3.3.
In Section 9.1 the optimal facility location x, for each of the 6 illustration cases is given. The optimum is obtained by the Simulated Annealing algorithm of Section 6.3. Section 9.2 contains the illustration results of a performed grid search to analyse the shape of the objective function over the feasible area.

9.1 Simulated Annealing illustration of the optima

The Simulated Annealing algorithm, was the best tested algorithm for Test Case 1 and Test Case 2. Therefore, the Simulated Annealing algorithm is used to find the optimal facility location of both competitive situations for all three norms.
The settings of the Simulated Annealing algorithm are the standard settings of the Matlab optimtool. With a random starting point in the feasible area, and a maximum number of function evaluations of 2700. The used Matlab files can be found in Appendices D the results in Appendices E.1 to E.2.

9.1.1 Two competitor case

The Simulated Annealing Matlab files of the three norms for the two competitor case are listed in Appendix D.2.1 to D.2.3, the results in Appendix E.1. Figure 9.1 presents the starting points and the outcome for the 100 runs of the Simulated Annealing Algorithm on the three norms.

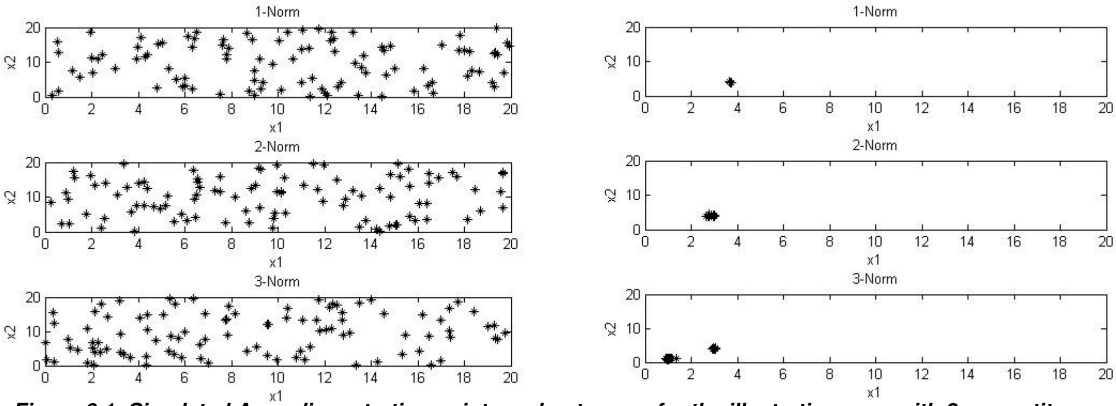

Figure 9.1: Simulated Annealing, starting points and outcomes for the illustration case with 2 competitors

Figure 9.2 presents the optimum facility locations for the three norms on the two competitor illustration case. The optimum locations are not identical but are all in the bottom left corner.

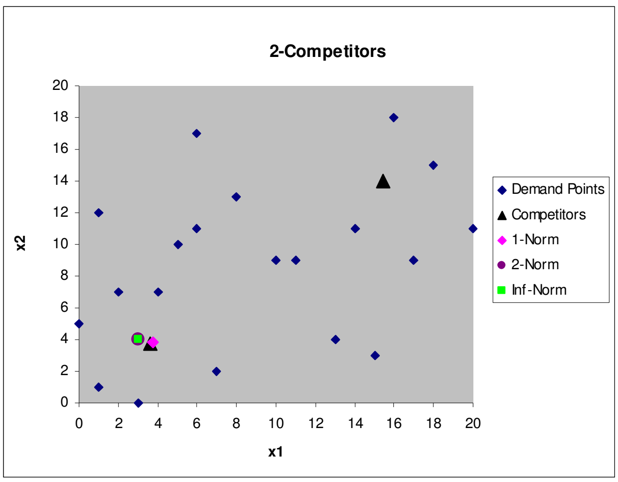

Figure 9.2: Simulated annealing optima for 2-competitors

9.1.2 Three competitor case

The Simulated Annealing Matlab files of the three norms for the three competitor case are listed in Appendix D.2.4 to D.2.6, the results in Appendix E.2. Figure 9.3 presents the starting points and the outcome for the 100 runs of the Simulated Annealing Algorithm on the three norms.

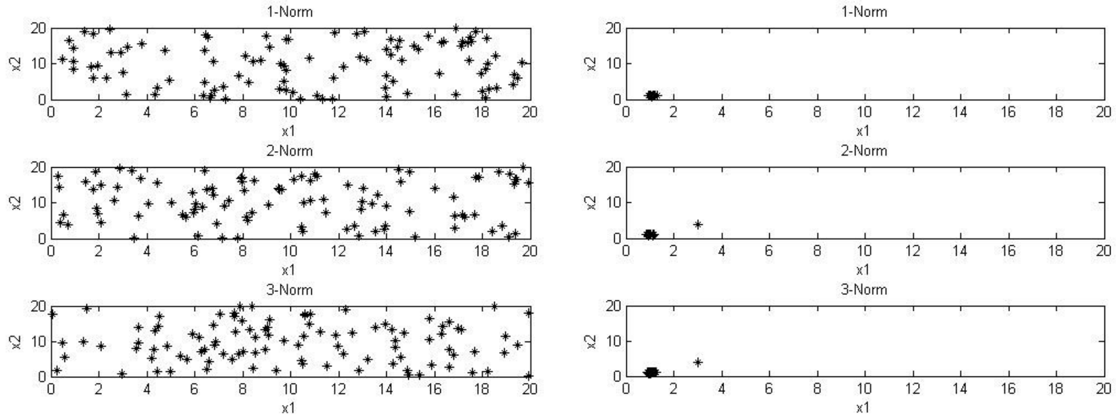

Figure 9.3: Simulated Annealing, starting points and outcomes for the illustration case with 3 competitors

For the case with three competitors, the optimum locations of all three norms are the same. For all three norms the optimum is *x=(1,1)*, see Figure 9.4.

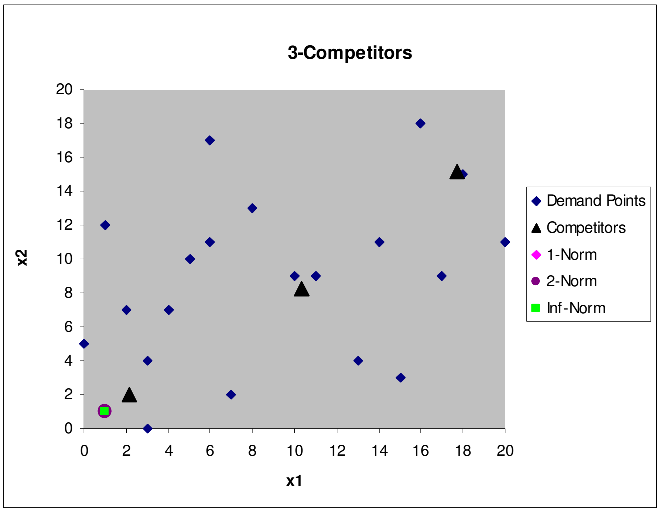

Figure 9.4: Simulated annealing optima for 3-competitors

9.1.3 Results of the Simulated Annealing illustrations

Table 9.1 presents the simulated annealing test results for all three norms in the illustration case with two competitors. The complete list of analytical results is added in Appendix E.1.

Table 9.1: Simulated annealing results for 2-competitors

	Average value over 100 runs	**Minimum** value over 100 runs	**Maximum** value over 100 runs	**Variance** over 100 runs	Average **number** function evaluations
1-norm	-18.0763	-18.0762	-17.4996	0.0038	1916.65
2-norm	-5.6379	-5.6556	-5.3217	0.0030	1852.24
∞-norm	-1.3365	-1.3677	-1.2690	0.0004	1762.13

For all three norms, it took the Simulated Annealing algorithm on average less than 2000 function evaluations to converge to a solution.

Table 9.2 presents the simulated annealing test results for all three norms in the illustration case with three competitors. The complete list of analytical results is added in Appendix E.2.

Table 9.2: Simulated annealing results for 3-competitors

	Average value over 100 runs	**Minimum** value over 100 runs	**Maximum** value over 100 runs	**Variance** over 100 runs	Average **number** function evaluations
1-norm	-3.7245	-3.7776	-2.9528	0.0280	1835.82
2-norm	-2.5744	-2.6619	-1.8244	0.0500	2005.20
∞-norm	-0.6702	-0.7505	-0.4646	0.0071	1838.96

The robustness values are quite lower in the 3-competitor case compared with the 2-competitor case. This is logical, as the demand is divided by more facilities.

It seems strange that the optimal locations for both test cases and each norm are all in the bottom left area (x_1 and x_2 are smaller than 5). In all cases, this is very close to one competitor.
In Figure 9.5 all demand points are shown with their nominal weight.

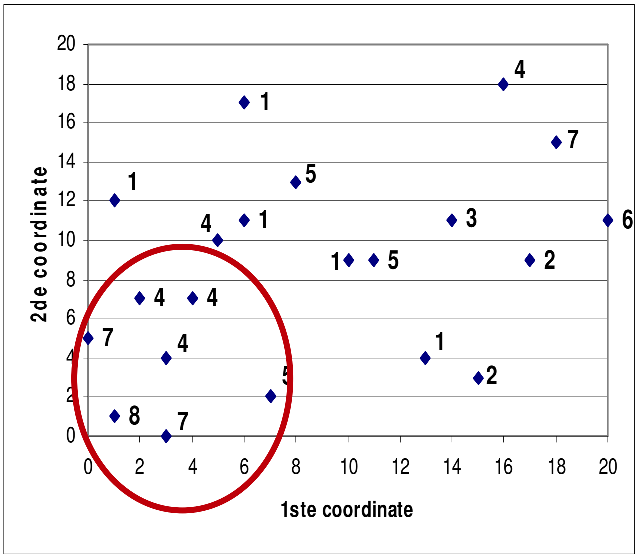

Figure 9.5: Demand point locations with their nominal weight

It turns out that from the nominal weights, almost 50% (39 of the 82 total nominal demand) is located close around the bottom left area. Therefore it is not strange that the optimal facility location for each case is in that area.

9.2 Grid Search illustration of the objective function

To visualise the value of the objective function for the whole feasible area, a grid search is performed with step size 0.5. From the objective function values, a contour graph is made with Excel 2003. The contour graph represents ranges of the function value with a colour. The corresponding ranges and colours are mentioned in the index of the figure.

In Section 9.1, the robustness values were negative. This was due to the fact that the optimtool of Matlab can only minimize. As the robustness was to be maximized, the function was made negative. Minimizing this negative function is the same than maximizing the non-negative function. The robustness values in all figures of this section are positive, as the grid search is no optimization technique and therefore the – sign has no meaning.

Figure 9.6 to 9.8, illustrates the contour graphs for placing 1-facility for the 2 competitor case, in Section 9.2.1. Figures 9.9 to 9.11 illustrates the contour graphs for the 3 competitor case.

The grid searches are performed in Matlab, the corresponding Matlab files are presented in Appendix D.3.

9.2.1 Illustration of the objective function for 2 competitors

In Figure 9.6, the grid search contour for the 1-norm deviation robustness is drawn. The blue area around *x=(4,4)* has the highest robustness values, the optimum of the simulated annealing algorithm *x*=(3.75,3.83)* lies within. The ecru, dark red and light blue areas corresponding with negative values of the objective function are non feasible locations. These areas are non feasible, as these facility locations do not obtain 20 demand units with the nominal demand vector *v*.

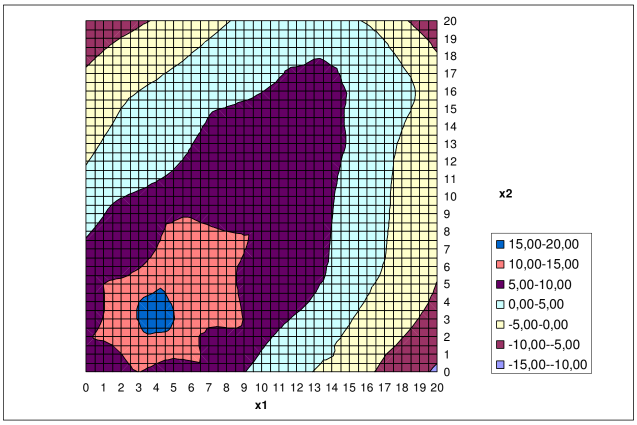

Figure 9.6: Grid Search 1-norm 2-competitors

In Figure 9.7, the grid search contour for the 2-norm deviation robustness is drawn. The blue area around *x=(4,3)* has the highest robustness values, the optimum of the simulated annealing algorithm *x*=(4,3)* lies within. In this graph, all objective values are positive, so all facility locations are feasible.

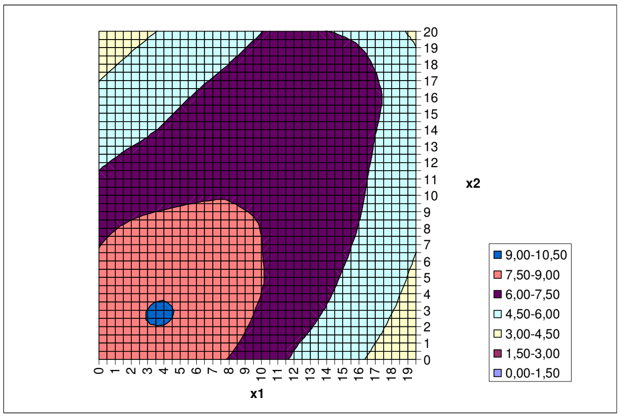

Figure 9.7: Grid Search 2-norm 2-competitors

In Figure 9.8, the grid search contour for the ∞-norm deviation robustness is drawn. The blue area around *x=(3,3)* has the highest robustness values, the optimum of the simulated annealing algorithm *x*=(4,3)* lies within. In this graph, all objective values are positive, so all facility locations are feasible.

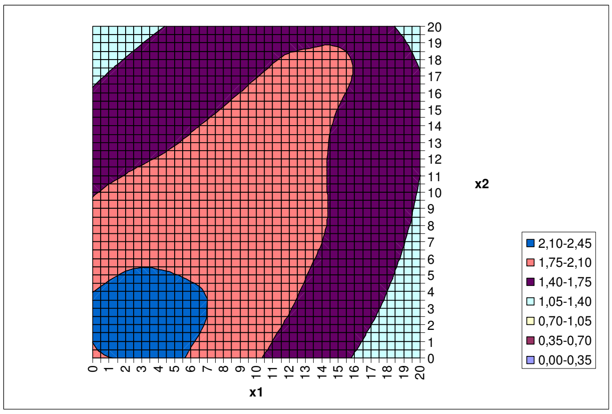

Figure 9.8: Grid Search infinite-norm 2-competitors

9.2.2 Illustration of the objective function for 3 competitors

In Figure 9.9, the grid search contour for the 1-norm deviation robustness is drawn. The blue area around *x=(4,3)* has the highest robustness values, the optimum of the simulated annealing algorithm *x*=(1,1)* lies within. All not blue areas have a negative function value and are non feasible.
Remarkable in this graph are the local red and purple spot. These are non local optima, as these locations are not feasible due to their negative robustness value.

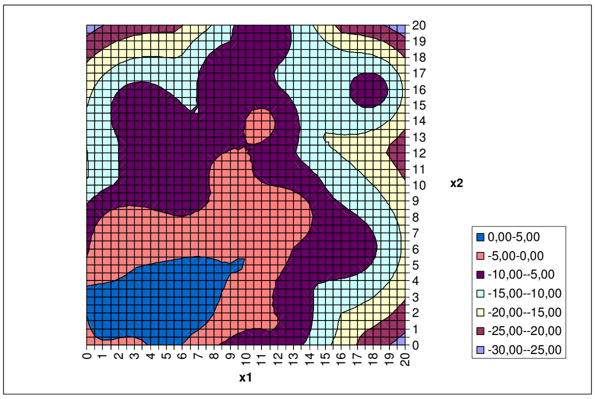

Figure 9.9: Grid Search 1-norm 3-competitors

In Figure 9.10, the grid search contour for the 2-norm deviation robustness is drawn. The blue area around *x=(3,2)* has the highest robustness values, the optimum of the simulated annealing algorithm *x*=(1,1)* lies within. Here the local deep purple spot is feasible, so there is at least a local optimum around *x=(15,15)*. The outer light purple area is not feasible, due to its negative objection value.

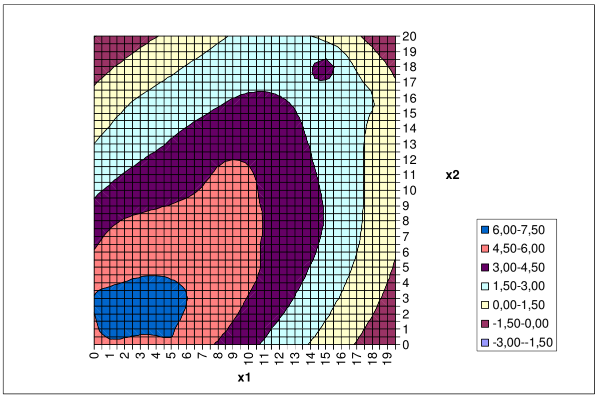

Figure 9.10: Grid Search 2-norm 3-competitors

In Figure 9.11, the grid search contour for the ∞-norm deviation robustness is drawn. The blue area around *x=(1,1)* has the highest robustness values, the optimum of the simulated annealing algorithm *x*=(1,1)* lies within. In this graph, all objective values are positive, so all facility locations are feasible.

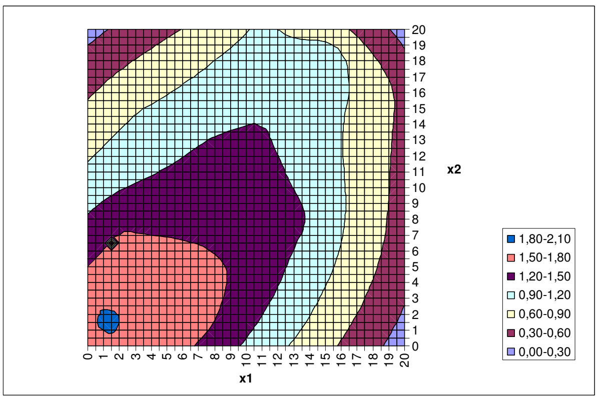

Figure 9.11: Grid Search infinite-norm 3-competitors

10. Conclusion, discussion and further research

Here the conclusions of the research are summarized, a discussion on the value and the strong and weak points are evaluated and an assessment for further research is presented.

10.1 Conclusion

The four subsections give answers and conclusions to the general and specific research questions.

10.1.1 Facility location

Facility location concerns the placement of facilities, for various objectives, by use of mathematical models and solution procedures.
By modelling issues as, feasible area, number of facilities, costs, distance measures, weights, competiveness, time, capacities, correlation, uncertainty and variability, an optimal facility location can be found which suits the objective the best.
The most common objectives for the placement of 1-facility location are: the p-centre problem, the p-median problem and the uncapacitated facility location problem.

10.1.2 Robustness Concepts

Robustness in facility location is a measure for the ability of a facility on a chosen facility location to perform under all expectable circumstances.
In literature, many definitions and descriptions of robustness exist. In this thesis, all these definitions are categorised into five robustness concepts:

1. The Yes or No performance robustness concept
2. The Probabilistic robustness concept
3. The Deviation robustness concept
4. The Safety first robustness concept
5. The Maximum regret robustness concept

The deviation robustness concept is the most interesting robustness concept as it suits the uncertainty and uncontrollability of data the best.
The deviation robustness can be measured in norms. Three norms are applicable for different real world situations, the 1-norm, the 2-norm and the infinite norm.
During this research, only one deviation robustness model is found in literature and one is developed.

10.1.3 Algorithm testing

Algorithms are tested on their efficiency and effectiveness. In this research, the focus is on the effectiveness of the algorithms and efficiency is limited as circumstance which is equally levelled for comparison reasons.
Two test cases with each two test problems are obtained to test the effectiveness of the algorithms.
Four stochastic optimization algorithms are considered:

1. Controlled Random Search
2. Genetic Algorithm
3. Simulated Annealing
4. Multi Start

The best tested algorithm is the Simulated Annealing algorithm. The Simulated Annealing algorithm found the best optimum in all cases. In three of the four cases, it also had the lowest variance, best average and best maximum. In the two robust test cases, Simulated Annealing found the optimum in the most often number of runs.

10.1.4 The competitive deviation robustness model

Only one deviation robustness model is found in literature. Only one robustness model found dealt with competition. The new model consists of three elements.

1. Competition is modelled according to the model of Huff
2. The goal of the model is to find a deviation robust location
3. Demand is uncertain and uncontrollable

To illustrate how the model works, a new test case is developed based on the two algorithm test cases. With the Simulated Annealing algorithm, the competitive deviation robustness model is solved, for all three considered norms. The new model appears to be multimodal and therefore challenging.

10.2 Discussion

The deviation concept of robustness suits the data circumstances the best as probability distributions and ranges of data are often arbitrary and therefore uncertain and uncontrollable. For the deviation robustness concept an assumption on the nominal values are needed, this assumption is probably unreliable as well.
In the case where the ranges or the probability distributions are known and reliable, the expected value can be used as a reliable nominal value.

The algorithms were tested by using the standard settings under comparable efficiency conditions. The algorithms are not tested after tweaking their parameters to obtain optimal results under the comparable conditions. The outcomes of the algorithm testing are therefore not a good representative conclusion.

The new developed competitive deviation robustness model work good for the used illustration case. The Simulated Annealing algorithm performed well in solving the robustness for all three norms, although there were multiple optima. The test case used is constructed by the author and perhaps not a good test for the solvability of the model.

10.3 Further research

By the consideration of which algorithm is the best, not everything is done to let each algorithm perform under its optimal conditions. Also not all algorithms are considered. Therefore there are improvements possible on:

1. Further tune the parameter settings of the algorithms
2. Consider other algorithms for solving robustness in facility location

Extra research on these points can lead to interesting improvements in the effectiveness and efficiency of solving bigger problems.

The developed model works well for the chosen two and three competitor test cases. The limitations on how good the model really is are not yet studied. Therefore it is interesting to do further research on:

3. Testing the model on larger test problems
4. Test the model on real world situations
5. Compare the ability of the model with other competitive robustness models

For future research it can be interesting to study possibilities to extend the model for more facility location issues:

6. From single to Multi facility
7. From single to Multi objective, for example:
    - minimize the total cost while obtaining a robust facility location for uncertain and uncontrollable demand.

The developed model shows interesting behaviour of the objective function. Therefore it seems interesting to:

8. Further study the behaviour of the model
9. Develop a specific solution procedure.

References

Avella, P., Benati, S., Canovas Martinez, L., Dalby, K., Di Girolamo, D., Dimitrijevic, B., Ghiani, G., Giannikos, I., Guttmann, N., Hultberg, T.H., Fliege, J., Marin, A., Munoz Marquez, M., Ndiaye, M.M., Nickel, S., Peeters, P., Perez Brito, D., Policastro, S., Saldanha de Gama, F.A. and Zidda, A., 1998, ´Some personal views on current state and the future of Locational Analysis´, European Journal of Operational Research 104, pp. 269-287

Averbakh, I. and Bereg, S., 2005, ´Facility location problems with uncertainty on the plane´, Discrete optimization, vol. 2, pp 3-34

Averbakh, I. and Berman, O., 1997, ´Minimax regret p-center location on a network with demand uncertainty´, Location science, vol. 5, no. 4, pp. 247-254

Barricelli, H.A., 1957, ´Symbiogenetic evolution processes realized by artificial methods´, Methodos, vol. 9, no. 35, pp. 143-182

Ben-Tal, A. and Nemirovski, A., 1998, ´Robust convex optimization´, Mathematics of Operations Research, vol. 23, no. 4, pp. 769-805

Berman, O. and Wang, J., 2004, ´Probabilistic location problems with discrete demand weights´, Networks, vol. 44, no. 1, pp. 45-57

Berman, O. and Wang, J., 2008, ´The probabilistic 1-maximal covering problem on a network with discrete demand weights´, Jorunal of the Operational Research Society, vol. 59, pp. 1398-1405

Burkard, R., Dell´Amico, M. and Martello, S., 2009, `Assignment problems` Still in Press

Carrizosa, E. and Nickel S., 2003, ‘Robust facility location’, Mathematical Methods of Operations Research, vol. 58, no. 2, pp. 331-349

Caruso, C., Colorni, A. and Aloi, L., 2003, ´Dominant´, an algorithm for the *p*-center problem, European Journal of Operational Research, vol. 149, no. 1, pp. 53-64

Claassen, G.D.H., Hendriks, Th.H.B. and Hendrix, E.M.T., 2007, ´Decision Science´, Theory and applications, Mansholt publication series – vol. 2

Drezner, T., Drezner, Z., and Shioge, S, 2002, ‘A threshold-satisfying competitive location model’, Journal of regional science, vol. 42 no. 2, pp. 287-299

Eiselt, H.A. and Laporte G., 1995, ´Objectives in location problems´, In Drezner, Z., ´Facility Location: A Survey of Applications an Methods´, Operations Research and Financial Engineering, Springer, Berlin, pp. 151-208

Fernandez, F.R., Nickel, S., Puerto, J. and Rodriguez-Chia, A.M., 2001, ´Robustness in the Pareto-Solutions for the Multi-Criteria Minisum location Problem´, Journal of Multi-Criteria decision analysis, vol. 10, pp. 191-203

Fernandez, J., Pelegrin, B., Plastria, F. and Toth, B., 2007, ´Solving a Huff-like competitive location and design model for profit maximization in the plane´, European Journal of Operational Research 179, pp. 1274-1287

Ghiani, G., Laporte, G. and Musmanno, R., 2004, Introduction to Logistics Systems Planning and Control

Hendrix, E.M.T., 1998, ´Global Optimization at Work´, PhD-Thesis, Wageningen University, Department of Mathematics

Hendrix, E.M.T., 2008, ‘On Robustness in facility location’, EURO Working Group on Locational Analysis Conference, Elche

Hendrix, E.M.T. and Toth, B.G., 2009, ´Introduction to Nonlinear and Global Optimization´, Almeria University – Computer Architecture and Electronics and Budapest University of Technology and Economics, In press

Hendrix, E.M.T., Ortigosa, P.M. and Garcia, I., 2001, ´On success rates for controlled random search´, Journal of Global Optimization 1, pp. 389-401

Hewlett Packerd, 2007, ´Technical guides and product specifications´, HP Compaq 6710B, Manuals, L.P. Development Company

Huff, D.L., 2003, ´Parameter Estimation in the Huff Model´, ArcUser, October-December, pp. 34-36

Huff, D.L., 1963, ´A Probabilistic Analysis of Shopping Center Trade Areas´, Land economics 39, pp. 81-90

Kaelo, P. and Ali, M.M., 2006, ´Some Variants of the Controlled Random Search Algorithm for Global Optimization´, Journal of optimization theory and applications, vol. 130, no. 2, pp. 253-264

Klose, A. and Drexl, A., 2005, ´Facility location models for distribution system design´, European Journal of Operational Research 162, pp. 4-29

Krarup, J., Pisinger, D. and Plastria, F., 2002, ´Discrete location problems with push-pull objectives´, Discrete Applied Mathematics 123, pp. 363-378

Langevin, A. and Riopel, D., 2005, ´Logistics Systems, design and optimization´, GERAD, pp. 95 and 96

Matlab 7.6.0, R2008a help; How the Genetic Algorithm Works

Metropolis, N., Rosenbluth, R.W., Rosenbluth, M.N., and Teller, A.E., 1953, ´Equation of State Calculations by Fast Computing Machines´, The journal of chemical physics, vol. 21, no. 6, pp. 1087-1092

Mladenovic, N., Labbe, M. and Hansen, P., 2003, ´Solving the p-Center Problem With Tabu Search and Variable Neighborhood Search´, Networks, vol. 42, no. 1, pp. 48-64

Ogryczak, W., 2008, ´Conditional Median as a Robust Solution for Locational Problems´, EURO Working Group on Locational Analysis Conference, Elche

Ogryczak, W., 2009, ´Conditional Median as a Robust Solution Concept for Uncapacitated Location Problems´, In press

Ogryczak, W. and Zawadzki, M., 2002, ´Conditional Median: A Parametric Solution Concept for Locational Problems´, Annals of Operations research 110, pp. 167-181

Olieman, N.J., 2008, ‘Methods for Robustness Programming’, PhD thesis Wageningen University

Ortigosa, P.M., 2008, ´Algoritmos de Optimizacion Global. Estrategias paralelas´, Course information

Pelegrin, B., Fernandez, J. and Toth, B., 2008, ´The 1-center problem in the plane with independent random weights´, Computers & Operations Research 35, pp. 737-749

Plastria, F., 2004, ´Location Models´, Department of Management Informatics Vrije Universiteit Brussel

Price, W.L., 1978, ´A Controlled Random Search Procedure for Global Optimization´, Toward Global Optimization 2, Edited by Dixon, L.C.W. and Szego, G.P., North-Holland Publishing Company, Amsterdam, Holland, pp. 71-84

Price, W.L., 1983, ´Global Optimization by Controlled Random Search´, Journal of optimization theory and applications, vol. 40, no. 3, pp. 333-348

Redondo, J.L., 2008, ´Solving competitive location problems via memetic algorithms. High performance computing approaches´, PhD-Thesis, University of Almeria, Department of Computers Architecture and Electronics

Saiz, M.E., Hendrix, E.M.T., Fernandez, J. and Pelegrin, B., 2008, ´On a branch-and-bound approach for a Huff-like Stackelberg location problem´, OR Spectrum, In Press

Serra, D. and Marianov, V., 1998, ´The p-median problem in a changing network: the case of Barcelona´, Location Science, vol. 6, pp. 383-394

Snyder, L.V., 2006, ´Facility location under uncertainty: a review´, Transactions, vol. 38, pp. 537-554

Stackelberg, H.F. von, 1934, ´Marktform und Gleichgewicht (Market Structure and Equilibrium)´, Julius Springer, Vienna

Vlajic, J.V., Vorst, G.A.J. van der, Hendrix, E.M.T., 2008, 'Food supply chain network robustness´, A literature review and research agenda, Working paper Mansholt graduate school, Discussion paper no. 42

Appendix A: Test Case Data

Appendix A.1: Data Demand Points Test Case 2 and 3

```
P = [3 4
10 9
6 17
15 3
4 7
6 11
11 9
2 7
1 12
5 10
7 2
8 13
17 9
13 4
14 11
0 5
1 1
3 0
20 11
18 15
16 18];
```

Appendix A.2: Data weight vector test case 2

```
w = [2 3 1 3 2 4 5 1 2 3 3 1 3 1 2 5 4 7 6 5 8];
```

Appendix A.3: Data weight matrix test case 3

```
W=[4 1 1 2 4 1 5 4 1 4 5 5 2 1 3 7 8 7 6 7 4
2 4 2 4 5 3 2 4 1 1 4 2 2 1 3 4 7 5 8 8 8
1 4 1 4 5 1 5 3 5 1 2 5 1 2 3 7 7 5 5 5 4
1 5 4 3 2 3 5 1 5 2 1 3 2 3 3 6 4 4 5 6 7
4 5 1 4 1 4 4 5 5 3 3 5 1 1 5 5 8 8 8 5 8
4 4 4 3 5 3 2 5 5 1 5 5 2 3 4 6 8 5 8 6 8
5 2 5 2 4 4 1 3 2 4 4 3 5 4 2 8 6 5 7 4 6
1 1 3 1 4 3 4 3 5 3 3 2 3 1 4 8 8 4 4 4 5
3 1 2 3 5 5 1 5 4 5 3 4 3 1 4 8 5 8 5 8 5
1 3 3 5 1 3 5 5 3 3 4 3 5 2 4 4 6 4 5 6 5
5 3 2 4 2 1 4 4 4 1 5 3 5 2 2 5 6 6 4 8 7
1 3 3 3 3 4 2 2 3 3 1 5 3 3 5 7 6 5 5 7 7
1 4 6 4 5 4 5 1 2 2 4 4 7 1 5 4 6 8 6 6 6
2 4 5 2 2 3 1 5 5 4 3 1 5 2 1 6 5 8 6 8 6
4 3 5 2 5 3 5 2 5 1 1 1 3 4 1 6 6 7 8 7 5
3 4 1 3 4 3 3 4 1 1 5 5 3 2 4 8 8 7 6 5 5
3 4 4 2 4 4 1 1 1 3 1 4 4 5 3 4 7 5 8 7 8
3 4 5 5 5 3 1 4 5 2 2 3 1 4 5 6 7 4 6 5 7
3 5 2 3 1 3 2 3 3 3 5 3 5 1 3 8 8 6 4 4 7
5 2 2 5 5 4 4 4 3 2 5 1 1 2 5 6 8 5 5 5 4
5 5 5 2 3 4 1 1 3 3 1 5 2 1 3 7 5 4 7 6 7
3 2 1 3 4 2 2 1 3 1 1 1 1 1 5 5 5 5 7 6 5
3 3 1 5 5 3 5 1 1 4 2 5 4 4 1 7 6 4 8 4 5
2 4 4 3 3 1 3 2 3 2 2 1 2 3 1 5 6 6 5 6 6
```

5	1	2	5	1	1	2	3	1	3	5	3	1	4	1	6	5	8	7	5	7
3	4	5	2	1	1	2	2	2	4	4	2	2	1	5	6	4	7	4	6	4
4	1	4	4	4	2	4	1	1	2	4	3	5	3	1	4	5	8	6	5	6
3	5	3	3	3	2	1	5	4	5	5	3	3	2	3	5	4	5	7	4	6
3	5	5	2	1	3	2	2	2	1	1	3	4	1	1	7	7	5	6	6	8
3	2	3	3	1	5	3	2	4	2	2	2	3	5	2	6	8	4	7	6	6
3	2	1	4	3	5	5	2	1	2	2	1	2	2	4	5	6	6	7	8	8
4	5	1	2	2	4	5	2	2	4	5	3	1	2	2	4	4	6	6	7	6
4	5	5	1	4	1	5	1	2	3	1	4	5	1	1	6	8	7	8	5	6
1	3	1	4	3	5	1	5	5	1	1	3	1	1	5	7	7	6	5	6	6
2	5	1	4	5	5	4	1	2	2	5	3	5	4	1	8	5	4	8	8	7
3	5	5	5	2	5	4	2	2	1	2	5	3	2	4	6	4	5	7	8	5
1	4	3	4	2	2	2	5	2	4	2	5	2	5	4	4	6	4	6	7	8
2	2	4	4	2	1	2	1	4	5	4	1	1	1	5	7	6	7	7	4	7
2	5	3	5	5	5	5	2	4	4	5	4	1	5	5	6	4	7	8	8	6
4	4	1	1	3	4	1	2	1	2	1	4	2	2	5	8	8	7	5	4	5
5	3	3	1	1	4	5	1	3	1	5	2	1	1	1	6	5	4	8	8	4
3	1	1	1	3	4	1	5	1	2	3	5	4	4	4	7	4	5	4	6	8
4	4	5	2	2	4	1	5	5	1	4	3	4	7	5	6	4	5	5	7	8
2	4	1	3	3	2	5	4	1	2	5	4	1	5	5	8	7	7	8	4	4
1	3	5	5	4	3	4	1	1	3	3	2	2	5	2	8	8	6	5	8	6
3	2	1	3	4	1	4	5	4	1	5	1	3	3	3	6	8	8	4	5	4
3	3	2	5	5	3	1	2	3	2	2	4	5	3	2	4	5	6	4	8	8
3	5	5	4	1	2	2	1	2	4	5	2	2	3	1	8	6	6	8	8	8
4	3	5	5	1	3	3	2	1	2	4	5	5	3	2	6	7	5	8	7	8
1	5	1	4	3	2	1	5	2	4	1	4	4	3	3	8	6	8	7	6	4
1	2	1	3	4	4	1	5	2	5	3	4	3	4	1	4	7	5	8	5	8
5	4	1	1	2	5	4	2	1	3	2	5	2	5	2	5	5	5	8	8	5
5	5	5	1	3	5	1	2	4	3	2	5	4	3	2	4	7	4	4	7	7
2	1	2	2	2	1	1	4	3	1	3	5	1	2	1	7	6	4	6	6	7
1	1	4	2	5	3	3	3	4	2	3	4	4	5	1	4	5	4	8	8	4
4	4	2	1	3	5	3	4	2	1	1	4	3	3	4	7	4	5	5	5	6
5	1	5	3	2	1	1	3	2	2	2	1	2	2	3	6	8	7	4	6	5
4	2	8	5	3	3	1	3	2	1	5	3	6	2	2	6	6	5	5	5	5
2	1	4	1	2	4	5	1	3	3	1	5	4	5	3	8	8	8	7	5	7
4	3	4	1	2	5	4	1	2	3	2	4	4	1	2	7	5	8	8	7	6
3	3	4	2	2	4	2	2	4	2	3	4	5	5	4	4	6	6	7	7	5
4	4	3	5	4	1	1	5	5	1	2	1	2	3	4	8	7	4	6	6	7
5	4	4	1	2	4	5	2	1	3	4	2	1	1	3	8	7	6	5	5	5
1	1	2	1	5	4	5	3	1	2	4	4	2	5	2	7	5	8	8	5	7
4	1	5	2	2	3	1	4	2	1	5	2	4	5	1	8	6	6	7	7	6
2	2	1	3	2	1	5	5	1	4	1	1	2	1	5	5	6	6	5	5	6
5	1	1	4	5	5	5	5	5	2	5	3	5	5	2	5	6	5	6	5	5
4	5	3	4	5	3	2	2	4	3	3	3	5	4	1	5	6	8	5	5	6
4	1	4	1	1	2	2	5	1	1	4	4	1	4	5	5	4	6	5	4	5
3	3	5	5	1	3	5	2	4	1	1	4	3	5	1	7	7	6	8	7	4
1	3	4	3	1	4	3	3	5	3	2	2	2	3	1	7	8	8	4	5	4
2	2	2	2	2	1	2	4	3	1	2	2	3	4	2	8	5	5	7	6	5
2	2	3	3	2	2	1	2	5	1	4	3	2	5	4	8	4	8	6	5	7
5	3	3	1	5	5	5	2	2	4	2	4	4	2	2	4	8	4	6	5	6
4	3	5	1	1	4	3	3	1	1	3	5	3	3	5	5	4	8	7	5	6
5	4	3	5	4	5	4	5	1	2	5	3	2	4	1	5	5	5	5	5	4
2	1	3	1	3	2	1	1	5	4	5	2	2	1	2	8	5	8	8	6	4
5	1	4	3	4	1	1	3	2	3	1	1	3	3	5	8	5	5	5	5	8
1	5	3	1	3	3	2	3	1	2	4	1	3	5	4	6	5	5	5	4	7
1	1	2	4	4	3	2	2	5	1	3	2	3	3	2	4	6	6	5	6	4
4	3	4	5	2	5	2	4	4	5	5	5	5	5	3	7	6	4	4	5	6
1	3	3	5	3	5	2	5	1	4	1	1	1	1	5	6	5	5	8	5	6
5	3	5	4	5	3	4	2	2	2	4	4	2	4	1	6	8	6	7	4	8
1	2	5	3	5	3	2	3	2	5	3	5	5	5	1	7	8	8	8	7	6
4	4	3	3	3	1	3	4	2	3	4	2	1	5	1	7	4	7	4	8	8

```
3  4  1  5  3  3  4  5  3  2  3  5  5  2  5  7  5  7  8  5  5
1  2  5  5  1  5  3  3  5  3  5  1  4  5  5  4  8  8  5  5  5
4  1  1  2  2  5  1  5  4  2  5  3  2  1  4  8  5  8  6  6  4
4  3  3  1  5  2  3  1  3  1  5  5  4  4  2  8  6  4  7  5  5
2  3  2  4  4  2  1  2  4  4  3  4  2  2  1  7  8  8  4  6  6
5  4  1  4  3  5  5  2  3  1  2  4  3  4  2  8  6  5  7  4  5
3  1  2  3  3  4  5  4  2  5  4  5  2  1  3  5  8  4  8  8  6
2  4  1  1  4  4  3  1  1  5  4  5  4  5  1  4  7  8  5  8  4
2  4  2  3  2  2  3  5  3  5  1  4  2  2  4  4  4  8  5  8  6
2  3  1  3  4  2  1  3  3  1  2  5  4  5  5  8  7  8  8  7  6
2  5  5  5  2  2  4  3  2  1  4  1  2  3  3  6  7  6  6  6  6
5  2  1  4  4  1  2  5  1  1  4  1  3  2  2  7  7  4  7  4  7
3  2  2  5  2  2  1  4  2  2  5  1  2  5  1  4  8  7  7  4  6
3  2  5  2  3  5  2  3  4  4  4  2  5  3  1  6  6  7  7  4  6
1  1  1  3  4  2  5  5  5  5  5  3  5  1  5  7  6  7  8  8  7];
```