

Delft University of Technology Faculty of Electrical Engineering, Mathematics and Computer Science Delft Institute of Applied Mathematics

Facility Location Models in Emergency Medical Service: Robustness and Approximations

A thesis submitted to the Delft Institute of Applied Mathematics in partial fulfilment of the requirements

for the degree

MASTER OF SCIENCE in APPLIED MATHEMATICS

by

R.B.O. Kerkkamp

Delft, The Netherlands May 2014

Copyright © 2014 by R.B.O. Kerkkamp. All rights reserved.



MSc THESIS APPLIED MATHEMATICS

"Facility Location Models in Emergency Medical Service: Robustness and Approximations"

R.B.O. Kerkkamp

Delft University of Technology

Daily supervisor Prof.dr. R.D. van der Mei Responsible professor

Prof.dr.ir. K.I. Aardal

Other thesis committee members

Dr. S. Bhulai C.J. Jagtenberg, MSc Prof.dr.ir. A.W. Heemink P.L. van den Berg, MSc

May 2014

Delft, The Netherlands

Abstract

In emergency medical service (EMS) the use of optimisation models and Operations Research techniques is becoming more common. EMS providers incorporate facility location models and simulation software packages into decision support tools, allowing the extensive evaluation of 'what-if' scenarios. We give a literature survey of facility location models applied to EMS and analyse the properties of one EMS model in particular, namely the Maximal Covering Location problem (MCLP).

We analyse the sensitivity of the MCLP to changes in the parameters and design approaches to construct insensitive solutions. Furthermore, we prove performance guarantees for two heuristic solution methods for the MCLP: the Greedy Search and the Swap Local Search. All solution methods are numerically evaluated using generated instances and realistic instances based on The Netherlands.

Our main research contributions are as follows. First, we apply Robust Optimisation to EMS optimisation models. We derive and analyse a Robust Counterpart formulation for a general linear constraint under the assumption of a certain polytopal uncertainty structure. Second, we present a constructive proof of the tight performance guarantee for the Swap Local Search. The proof explicitly derives the family of worst-case MCLP instances, which have a certain symmetry. Finally, we perform a thorough computational study for the described methods.

This research is performed to conclude the Master in Applied Mathematics at the Delft University of Technology in The Netherlands. It is a cooperative project with CWI in Amsterdam, as part of the REPRO research project on ambulance logistics. CWI is the national research institute for mathematics and computer science in The Netherlands. For more information on the REPRO project, see repro.project.cwi.nl.

Contents

Li	st of	Figures	9
Li	st of	Tables	11
Li	st of	Algorithms	13
Li	st of	Symbols	15
1	Inti	roduction	17
	1.1	Emergency Medical Service	18
	1.2	Mathematical Modelling	19
	1.3	Research Objectives and Structure	20
2	Fac	ility Location Models	21
	2.1	Introduction	22
	2.2	Deterministic EMS Models	23
	2.3	Probabilistic EMS Models	33
	2.4	Multiple Vehicle Types	44
	2.5	Multiple Time Periods	45
	2.6	Conclusion	47
3	Sen	sitivity Analysis for the MCLP	49
	3.1	Introduction	50
	3.2	Robust Optimisation	51
	3.3	Post-Optimisation Neighbourhood Analysis	63
	3.4	Numerical Results	66
	3.5	Two Dutch Regional Ambulance Services in Detail	94
	3.6	Conclusion	106
4	ADI	proximations for the MCLP	107
	4.1	Introduction	108
	4.2	Submodular Functions	110
	4.3	Greedy Search for the MCLP	114
	4.4	Swap Local Search for the MCLP	125
	4.5	Numerical Results	146
	4.6	Two Dutch Regional Ambulance Services in Detail	158
	4.7	Conclusion	164

5	Conclusions and Future Work	165
	5.1 Conclusions	166
	5.2 Future Work	168
Α	Complexity Theory	177
	A.1 Introduction	178
	A.2 Complexity Classes P and NP	179
в	Implementation	181
	B.1 Introduction	182
	B.2 A Selection of Algorithms	183
С	Additional Tables	193
	C.1 Travel Time Thresholds	194
	C.2 Greedy Search Results	196
	C.3 Swap Local Search Results	201
Bi	bliography	215

List of Figures

1.1.1	The Regional Ambulance Services of The Netherlands.	18
3.3.1	Types of ordered performance curves (OPCs).	65
3.4.1	The 24 Regional Ambulance Services of The Netherlands.	67
3.4.2	Maximum coverage attainable with the shown number of bases.	71
3.4.4	Maximum coverage attainable with the shown travel time threshold	79
3.4.6	Ordered performance curves of the 1-Swap neighbourhood	85
3.4.8	Ordered performance curves of the 2-Swap neighbourhood	87
3.4.10	Relative perturbation in the number of inhabitants for parameter c	91
3.5.1	Gooi- en Vechtstreek (RAV14) of The Netherlands.	94
3.5.2	The 4- and 6-digit postal code areas of Gooi- en Vechtstreek (RAV14).	95
3.5.3	The number of inhabitants of each 4-digit postal code area of Gooi- en Vecht-	
	streek (RAV14)	96
3.5.4	Optimal MCLP base locations for RAV14.	97
3.5.5	Alternative optimal MCLP base locations for RAV14	97
3.5.6	Current and two-stage MCLP base locations for RAV14.	98
3.5.7	Optimal MCLP base locations for RAV14 with different travel time thresholds.	99
3.5.8	Limburg-Noord (RAV23) of The Netherlands.	100
3.5.9	The 4- and 6-digit postal code areas of Limburg-Noord (RAV23).	101
3.5.10	The number of inhabitants of each 4-digit postal code area of Limburg-Noord	
	$(RAV23). \ldots \ldots$	102
3.5.11	Optimal MCLP base locations for RAV23.	103
3.5.12	Current and two-stage MCLP base locations for RAV23.	104
3.5.13	Robust Counterpart MCLP base locations for RAV23.	105
4.3.1	Worst-case examples for 1-Greedy Search.	120
4.3.2	Example of an Uncapacitated Location problem instance transformation to an	
	MCLP instance.	122
4.3.3	Worst-case example for Reverse 1-Greedy Search.	124
4.4.1	Example of the first MCLP instance transformation ϕ_1	130
4.4.2	Example of the second MCLP instance transformation ϕ_2	132
4.4.3	Example of an MCLP instance in $\phi_2(\phi_1(\Omega))$	132
4.4.4	Example of permuted MCLP instances.	136
4.4.5	Example of the fourth MCLP instance transformation ϕ_4	138
4.4.6	Worst-case examples for Swap Local Search.	145
4.6.1	RAV14 and RAV23 of The Netherlands.	158
4.6.2	Heuristic base locations for RAV14	160

4.6.3	Heuristic base locations for RAV23.	162
5.2.1	Example of an MCLP instance transformation to a <i>p</i> -Median problem instance.	176

List of Tables

3.4.1	Specifications of all Regional Ambulance Services (RAV) regions.	68
3.4.2	Minimum number of bases to attain full coverage.	70
3.4.3	Maximum coverage attainable with the shown number of bases.	73
3.4.4	Minimum travel time threshold to attain full coverage.	78
3.4.5	Maximum coverage attainable with the shown travel time threshold	81
3.4.6	Number of optimal solutions in the neighbourhood.	84
3.4.7	Number of alternative optimal solutions with the shown number of bases	89
3.4.8	Robust coverage of the MCLP and Robust Counterpart MCLP (RC-MCLP).	92
3.4.9	Non-robust coverage of the MCLP and Robust Counterpart MCLP (RC-MCLP).	93
4.5.1	Theoretical bounds for the relative optimality gap of ρ -Greedy Search for the	
	RAV regions.	148
4.5.2	Relative optimality gap of ρ -Greedy Search for the RAVs	149
4.5.3	Theoretical bounds for the relative optimality gap of ρ -Swap Local Search for	
	the RAV regions.	151
4.5.4	Relative optimality gap of ρ -Swap Local Search for the RAV regions	152
4.5.5	Number of ρ -Swap Local Search iterations for the RAV regions	153
4.5.6	Number of isolated demand points for the generated Euclidean instance classes.	155
4.5.7	Performance of the heuristic solution methods	157
B.2.1	Example of a Greedy Search iteration.	184
B.2.2	Example of a Swap Local Search iteration.	187
C.1.1	Average travel time thresholds for the generated Euclidean instance classes.	195
C.2.1	Average relative optimality gaps of Greedy Search for the generated Euclidean	
	instances	197
C.3.1	Average relative optimality gaps and iterations of Swap Local Search after	
	Greedy Search for the generated Euclidean instances.	202

List of Algorithms

4.3.1	Greedy Search for Non-Decreasing Submodular Functions	114
4.3.2	Reverse Greedy Search for Non-Decreasing Submodular Functions	123
4.4.1	Swap Local Search for Non-Decreasing Submodular Functions	125
B.2.1	Calculate Coverage for the MCLP	183
B.2.2	Greedy Search for the MCLP	185
B.2.3	Greedy Search Iteration for the MCLP	186
B.2.4	Swap Local Search for the MCLP	188
B.2.5	Swap Local Search Iteration for the MCLP	189
B.2.6	Determine Travel Time Thresholds for the MCLP	191

List of Symbols

Set	Definition
$\mathbb B$	Binary numbers $\{0,1\}$.
\mathbb{N}	Non-negative integer numbers $\{0, 1, 2, \ldots\}$.
\mathbb{Z}	Integer numbers $\{\ldots, -2, -1, 0, 1, 2, \ldots\}$.
\mathbb{R}	Real numbers $(-\infty, \infty)$.

Notation	Definition			
\mathcal{O}	Big-O notation, describing the limiting behaviour of the growth			
	rates of functions. We say $f \in \mathcal{O}(g)$ if $ f $ is asymptotically			
	bounded from above by g up to a constant factor: $ f(n) \leq cg(n)$			
	for all $n > n_0$ for some constants n_0 and $c \in \mathbb{R}_{>0}$.			

Set	Elements	Definition
\mathcal{I} \mathcal{J}	$i \atop j$	Ambulance bases. Demand points.

Variable	Index	Definition
$x_i \in \mathbb{B}$	$i \in \mathcal{I}$	One if ambulance base i is opened, zero otherwise.
$y_i \in \mathbb{N}$	$i \in \mathcal{I}$	Number of emergency vehicles assigned to base i .
$z_j \in \mathbb{B}$	$j \in \mathcal{J}$	One if demand point j is covered, zero otherwise.

Parameter	Index	Definition
$a_{ij} \in \mathbb{B}$	$i \in \mathcal{I}, j \in \mathcal{J}$	One if base i can cover point j , zero otherwise.
$d_j \in \mathbb{R}_{\geq 0}$	$j\in \mathcal{J}$	Demand in point j .
$\lambda_j \in \mathbb{R}_{\geq 0}$	$j\in \mathcal{J}$	Call arrival rate in point j .
$p \in \mathbb{N}$	-	Maximum number of opened bases.
$q \in \mathbb{N}$	-	Maximum number of emergency vehicles.
$q_i \in \mathbb{N}$	$i \in \mathcal{I}$	Maximum number of vehicles assigned to base i .

Chapter 1

Introduction

Contents

1.1	Emergency Medical Service	18
1.2	Mathematical Modelling	19
1.3	Research Objectives and Structure	20

1.1 Emergency Medical Service

In case of an out-of-hospital medical emergency it is of vital importance that paramedics arrive on site swiftly and that the patient(s) can be transported to hospital if needed. This requires an ambulance and its crew of paramedics to be stationed nearby. It has never been possible to station ambulances at each road intersection, primarily because this would lead to unacceptably high costs. Therefore, we have a limited number of ambulances and crews, and should use them in the most efficient way.

Emergency medical service (EMS) is the organisation and coordination of (out-of-hospital) acute medical care and transportation in designated regions. EMS typically consists of call centres, dispatchers, ambulances, and paramedics. For example, the EMS in The Netherlands is divided into 24 so-called Regional Ambulance Services, each with their own coordination centre and designated service region (see Figure 1.1.1). To give an indication of the size of the Dutch EMS: there are in total around 700 ambulances stationed at approximately 200 ambulance bases (see for example Boers et al. (2010)).

There is a differentiation in the severity, and thus the urgency, of emergency medical calls: A1 calls correspond to life-threatening situations and A2 calls to less severe situations. Furthermore, ambulances provide planned (non-urgent) transportation of patients between hospitals (or other locations), which are called B trips. There are approximately 500 000 A1 calls, 250 000 A2 calls, and 350 000 B trips each year in The Netherlands.

Ambulances aim to arrive on site within certain response time standards: within 15 minutes for A1 calls and within 30 minutes for A2 calls. Note that the response time includes the time to handle the call, to dispatch the ambulance, and the travel time. Consequently, an ambulance is said to 'cover' a region if it can reach the area within a certain response time threshold.

Given the available ambulances and the response time standards, emergency medical services are faced with the challenge to optimise their performance. This often translates to optimisation problems common in Operations Research, e.g., maximising the number of areas covered by ambulances.



Figure 1.1.1: The Regional Ambulance Services of The Netherlands distinguished by colour. (Modified from source: www.zorgatlas.nl.)

1.2 Mathematical Modelling

Operations Research in the emergency medical service is aimed at finding a balance between service performance and costs, where the allowed decisions are limited by physical and societal constraints. Fundamental properties of the EMS are:

- the current state,
- the allowed state changing actions,
- the objective.

In today's Information Age, the current state of the EMS can be estimated with great accuracy. Emergency service call centres keep track of incoming calls (location, severity, and call duration), ambulances have GPS tracking equipment (driving and response times), and hospitals keep records of the medical outcome (survival rates). The current locations and capacity of ambulance bases are also known.

The state of the EMS can be altered by actions such as repositioning ambulances, changing dispatch rules, and reducing ambulance response times by expanding the infrastructure. The state changing actions can be divided between possible and allowed actions. For instance, it is (physically) possible to close all ambulance bases. Service performance would decline, resulting in a life-threatening state. However, such states are not accepted by society, leading to societal constraints on the performance of the EMS. For example, in The Netherlands a societal constraint is that the ambulance response time to a call should be less than 15 minutes.

Last but not least, is the objective, a vital component in Operations Research. Given the current state and allowed actions, what do we want to improve? Common objectives are: reducing costs, minimising response times, and maximising patient survivability. In practice, the objective is often a combination of goals, resulting in multi-objective optimisation.

When advising on changes to the emergency medical service, it is important to justify the suggested actions. Mathematical models can contribute to the required justification. In particular, Mixed Integer Programming optimisation models and simulations are useful for decision support. The EMS can be abstracted and formulated as an optimisation model: the current state and allowed actions translate to decision variables and constraints, and the objective is minimised or maximised. Various changes to the EMS can be quickly evaluated, resulting in suggested actions to be taken. Simulations can be used as an additional validation of the results.

The formulation of the EMS optimisation model is not a trivial task. It requires abstractions, assumptions, and simplifications. A complex model is realistic, but often unmanageable. Each model should be tailored to the desired EMS features and the objective. Consequently, the results should always be seen in the context of the used model.

1.3 Research Objectives and Structure

Our research focusses on facility location in the emergency medical service, i.e., the placement of ambulance bases and the allocation of ambulances to the bases. In particular, we are interested in Mixed Integer Programming optimisation models. Choosing or formulating a suitable model is essential for decision support. Therefore, we give a literature survey of EMS facility location models in Chapter 2, ranging from simplistic to more advanced models.

For any optimisation model it is useful to perform a sensitivity analysis on the parameters. It is often the case that the required resources rapidly increase near the boundary of the constraints. For example, increasing coverage from 94% to 95% can significantly and disproportionally increase costs. One can question if such an investment is really necessary. Furthermore, the parameters are usually estimated from data and can contain small errors. Preferably, the final solution of the optimisation model should be insensitive to such errors.

Emergency medical service optimisation problems can become impractical if they are difficult to solve to (near-)optimality. Moreover, commercial software with advanced solution methods is not always available. Therefore, the performance of solution methods (both exact and heuristic) should be considered. For example, do simple heuristics have good performance or are advanced solution methods required? If we use heuristics, can we give performance guarantees?

We will consider a basic model of our survey in detail, called the Maximal Covering Location problem. In Chapter 3 we perform a sensitivity analysis on the model parameters and design methods to construct solutions that are insensitive to changes in the parameters. One method uses a technique called Robust Optimisation, which constructs solutions that remain feasible under parameter uncertainty.

In Chapter 4 we discuss two heuristic solution methods: the Greedy Search and the Swap Local Search. We give theoretical performance guarantees for these methods, using the framework of submodular functions. In particular, we present a tight guarantee for the Swap Local Search. The corresponding proof is a constructive proof of a family of worst-case instances. Besides worst-case performance, we also consider the empirical performance for realistic instances. Details on the implementation are given in Appendix B. Finally, we summarise our findings and suggest topics for future research in Chapter 5.

We assume that the reader is familiar with the basic concepts of mathematical optimisation, e.g., Linear Programming, duality, Mixed Integer Programming, and heuristics (see for example Papadimitriou and Steiglitz (1998)). Other optimisation techniques and terminology are introduced accordingly, with references for further reading. In particular, an informal introduction to complexity theory is given in Appendix A.

Chapter 2

Facility Location Models

Contents

2.1	Intr	oduction	22
2.2	Det	erministic EMS Models	23
	2.2.1	Coverage Neglect	25
	2.2.2	Backup Coverage	26
	2.2.3	General Coverage Constraints	28
	2.2.4	Survival and Decay Models	31
2.3	Pro	babilistic EMS Models	33
	2.3.1	Hypercube Queueing Correction	34
	2.3.2	Stochastic Response Times and Preferred Bases	35
	2.3.3	Reliability Constraints	38
	2.3.4	Loss Models	41
	2.3.5	Stochastic Programming Models	41
2.4	Mul	tiple Vehicle Types	44
2.5	Mul	tiple Time Periods	45
2.6	Con	clusion	47

2.1 Introduction

Facility location has been an active research topic in Operations Research, originally investigated for military purposes, but has found many purposes in other application areas. One of these areas is the field of emergency medical service (EMS), where facility location models have been applied since the 1970's. The challenge of facility location in EMS is to determine where to place ambulance bases (facilities) such that a desired service level is attained. For instance, response times to emergency calls should be minimal.

There are three main types of service goals (objectives) used in EMS:

- maximise the number of regions reachable within predefined maximum response times,
- minimise the average response time to emergency calls,
- minimise the maximum response time to emergency calls.

Service goals are often combined with budget constraints. The three types correspond to coverage models, *p*-median models, and *p*-centre models, respectively. Coverage models are the most common type in the literature. Furthermore, most models discretise the real world to finite sets and graphs. Two fundamental sets are the set of all possible locations for ambulance bases and the set of demand points, i.e., the sources of emergency calls.

We will focus on the (discretised) EMS coverage models, which we classify into deterministic and probabilistic models. This classification is not binding: when viewed as abstract mathematical models, some EMS models have a similar structure even though they are classified differently. The deterministic models are treated in Section 2.2, followed by the probabilistic models in Section 2.3. Both types of models can be extended to multiple vehicles types (Section 2.4) and multiple time periods (Section 2.5). This review of EMS models is partially based on the review papers by Brotcorne et al. (2003), Farahani et al. (2012), Goldberg (2004), Li et al. (2011), and ReVelle (1989), and the theses by Looije (2013) and Van Buuren (2010).

The models discussed in the next sections will often use the same type of parameters and variables. We have attempted to unify the notation to make it consistent among all shown models. The following notation is shared by all models and used in all chapters. All possible ambulance bases are denoted by the (finite) set \mathcal{I} and all demand points by the (finite) set \mathcal{J} . Whether we open a certain base $i \in \mathcal{I}$ is indicated by the decision variable $x_i \in \mathbb{B}$. That is, $x_i = 1$ if and only if base *i* is opened. Often, at most $p \in \mathbb{N}$ bases can be opened in total. Each base can cover (service) a subset of demand points, therefore, we need variables $z_j \in \mathbb{B}$ to keep track of whether point $j \in \mathcal{J}$ is serviced: $z_j = 1$ if and only if point *j* is covered.

Unless stated otherwise, the models assume time independent deterministic demand or stationary arrivals of calls for each demand point. Coverage properties (parameters) are assumed to be known and time independent deterministic, and ambulances are identical.

2.2 Deterministic EMS Models

Two early EMS models are the Location Set Covering model (LSCM) by Toregas et al. (1971) and the Maximal Covering Location problem (MCLP) by Church and ReVelle (1974). Both models assume that the requirements for covering a demand point is deterministic and satisfies an 'all-or-nothing' relation. That is, either a point is covered or uncovered, and we know exactly which one is the case. As a result, we can define coefficients $a_{ij} \in \mathbb{B}$ for $i \in \mathcal{I}$ and $j \in \mathcal{J}$ such that

$$a_{ij} = \begin{cases} 1 & \text{if point } j \text{ can be covered by base } i \\ 0 & \text{otherwise} \end{cases}$$

The Location Set Covering model (LSCM) is given in Model 2.2.1. As mentioned above, the decision variable x_i denotes whether we open (1) or close (0) base $i \in \mathcal{I}$. The objective of the LSCM is to minimise the number of opened bases whilst covering all demand points. The LSCM allows us to estimate the maximum number of required bases (subject to the assumptions). Note that the LSCM is infeasible if any point cannot be covered.



Model 2.2.1: Location Set Covering model (LSCM).

The mathematical formulation of the Maximal Covering Location problem (MCLP) has two equivalent versions, see Model 2.2.2. The parameter $p \in \mathbb{N}$ is the maximum number of allowed opened bases and the parameters $d_j \in \mathbb{R}_{\geq 0}$ indicate the demand (the weight) of point $j \in \mathcal{J}$. For EMS models it is natural and common to assume that these weights are non-negative. The MCLP assumes that demand is known and fixed. The decision variable z_j indicates whether demand point $j \in \mathcal{J}$ is covered (1) or not (0). All other parameters and variables are the same as in the LSCM.

The objective is to either maximise covered demand or minimise uncovered demand, given that we can open up to p bases. Both formulations are valid, but we will use the maximisation of covered demand. The alternative is more similar to the LSCM. As the demand is non-negative, it is optimal to open exactly p bases. By adding an insignificantly small penalty to opening a base, we can verify if p bases are indeed necessary. For this reason, we have chosen for the formulations with inequalities.

Maximise			Minimise
$\sum_{j\in\mathcal{J}}a$	$b_j z_j$		$\sum_{j \in \mathcal{J}} d_j (1 - z_j)$
subject to			subject to
$\sum_{i \in \mathcal{I}} x_i \le p,$			$\sum_{i\in\mathcal{I}}x_i\leq p,$
$\sum_{i \in \mathcal{I}} a_{ij} x_i \ge z_j$	$\forall j \in \mathcal{J},$	(2.2.2)	$\sum_{i \in \mathcal{I}} a_{ij} x_i + (1 - z_j) \ge 1 \qquad \forall j \in \mathcal{J},$
$x_i \in \mathbb{B}$	$\forall i \in \mathcal{I},$		$x_i \in \mathbb{B} \qquad \forall i \in \mathcal{I},$
$z_j \in \mathbb{B}$	$\forall j \in \mathcal{J}.$		$z_j \in \mathbb{B} \forall j \in \mathcal{J}.$

Model 2.2.2: Maximal Covering Location problem (MCLP).

A model that is strongly connected to the LSCM is the Hierarchical Objective Set Covering model (HOSC) by Daskin and Stern (1981), see Model 2.2.3. It is similar to the LSCM, but rewards multiple coverage of demand points (captured by the variables u_j). The parameter $\gamma \in \mathbb{N}$ is a positive weight. For instance, if γ is big enough, the objective is to first minimise the number of opened bases, followed by increasing multiple coverage.

Minimise	
$\gamma \sum_{i \in \mathcal{I}} x_i$:	$-\sum_{j\in\mathcal{J}}u_j$
subject to	
$\sum_{i\in\mathcal{I}}a_{ij}x_i-u_j\geq 1$	$\forall j \in \mathcal{J},$
$u_j \in \mathbb{N}$	$\forall j \in \mathcal{J},$
$x_i \in \mathbb{B}$	$\forall i \in \mathcal{I}.$

Model 2.2.3: Hierarchical Objective Set Covering model (HOSC).

The Goal-oriented Location Covering model (GLCM) by Storbeck (1982) is related to the three mentioned EMS models. It approaches facility location from a Goal Programming perspective. For weight parameter $\gamma \in \mathbb{N}$, the GLCM is shown in Model 2.2.4. The decision variables w_j^+ and w_j^- indicate the overshoot and undershoot in coverage, respectively.

The LSCM, MCLP, HOSC and GLCM show four ways to model the underlying real multicriteria objective (minimising costs and maximising coverage). The LSCM enforces coverage of all demand points and minimises costs, whereas the MCLP maximises coverage for a given number of bases (a given budget). The HOSC and GLCM models incorporate coverage overshoot and undershoot in the objective. All four models can be used as a starting point for more complex models, but mainly the LSCM and MCLP are used as such in the literature. In the next sections we will discuss several extensions to these models.

Minimise

$$\begin{aligned}
-\sum_{j\in\mathcal{J}} w_j^+ + \gamma \sum_{j\in\mathcal{J}} d_j w_j^- \\
\text{subject to} \\
\sum_{i\in\mathcal{I}} a_{ij} x_i - w_j^+ + w_j^- &= 1 \qquad \forall j\in\mathcal{J}, \\
\sum_{i\in\mathcal{I}} x_i \leq p, \\
w_j^+ \in \mathbb{N} \qquad \forall j\in\mathcal{J}, \\
w_j^- \in \mathbb{B} \qquad \forall j\in\mathcal{J}, \\
x_i \in \mathbb{B} \qquad \forall i\in\mathcal{I}.
\end{aligned}$$

Model 2.2.4: Goal-oriented Location Covering model (GLCM).

2.2.1 Coverage Neglect

One major shortcoming of the MCLP is that arbitrarily long response times are possible for uncovered demand points. This is due to the 'all-or-nothing' (binary) coverage. As a result, demand points that are difficult to cover are completely neglected. In the literature there are several models that try to counteract this negative result by introducing alternative or additional coverage constraints.

One common approach is to have two types of coverage, for instance, response times within 15 minutes and within 30 minutes. The coverage constraint is that each demand point must be reached within 30 minutes and the objective is to maximise the (weighted) number of demand points covered within 15 minutes. The mathematical formulation is as follows. We define

$$a_{ij}^{c_{15}} = \begin{cases} 1 & \text{if point } j \text{ can be reached by base } i \text{ within 15 minutes} \\ 0 & \text{otherwise} \end{cases},$$
$$a_{ij}^{c_{30}} = \begin{cases} 1 & \text{if point } j \text{ can be reached by base } i \text{ within 30 minutes} \\ 0 & \text{otherwise} \end{cases}.$$

The more strict coverage constraints can be incorporated in the MCLP by replacing Equation (2.2.2) with the following constraints:

$$\sum_{i \in \mathcal{I}} a_{ij}^{c_{15}} x_i \ge z_j \qquad \forall j \in \mathcal{J},$$
$$\sum_{i \in \mathcal{I}} a_{ij}^{c_{30}} x_i \ge 1 \qquad \forall j \in \mathcal{J}.$$

The resulting model is the Maximal Covering Location problem with Mandatory Closeness Constraints by Church and ReVelle (1974). This model can be viewed as a combination of the LSCM and MCLP. Points that cannot be reached within time (15 minutes) are not completely neglected, as these must still be reached within 30 minutes. Alternatively, we can enforce a minimum average coverage of a certain area. If at least a fraction of $\alpha \in [0, 1]$ of the total demand must be covered, we can add the constraint

$$\sum_{j \in \mathcal{J}} d_j z_j \ge \alpha \sum_{j \in \mathcal{J}} d_j.$$

For the MCLP this particular constraint makes little sense, but if we divide all demand points into regions we can enforce average coverage per region. When such constraints are added to a model, neglect of a region is prevented.

Finally, there is a whole different class of EMS models that drop the 'all-or-nothing' assumption. These will be treated in Section 2.2.4.

2.2.2 Backup Coverage

In reality, when a region (a demand point) is covered by a base $i \in \mathcal{I}$, it does not imply that an ambulance at base *i* is available when an emergency accident occurs. All ambulances at the base could be busy handling other calls. There are probabilistic ways to model these phenomena, which will be treated in Section 2.3. A deterministic approach is to add backup coverage into the model.

An early backup coverage model is the Backup Coverage problem (BACOP) by Hogan and ReVelle (1986). It is the similar to the MCLP, but requires that each point is covered by at least one base. The objective is to maximise the (weighted) number of points that is covered twice. This model is referred to as BACOP1. Alternatively, we do not enforce coverage of all points, but modify the objective. The modified objective is a weighted sum of once and twice covered demand. This variation is referred to as BACOP2. For weight $\gamma \in [0, 1]$, BACOP2 is shown in Model 2.2.5.

Maximise

$$\begin{split} \gamma \sum_{j \in \mathcal{J}} d_j w_j + (1 - \gamma) \sum_{j \in \mathcal{J}} d_j z_j \\ \text{subject to} \\ \\ \sum_{i \in \mathcal{I}} x_i \leq p, \\ \sum_{i \in \mathcal{I}} a_{ij} x_i \geq w_j + z_j \qquad \forall j \in \mathcal{J}, \\ \sum_{i \in \mathcal{I}} a_{ij} x_i \geq w_j + z_j \qquad \forall j \in \mathcal{J}, \\ x_i \in \mathbb{B} \qquad \forall i \in \mathcal{I}, \\ w_j, z_j \in \mathbb{B} \qquad \forall j \in \mathcal{J}. \end{split}$$

Model 2.2.5: Backup Coverage problem 2 (BACOP2).

The decision variable $w_j \in \mathbb{B}$ is equal to 1 if point j is covered at least once and zero otherwise. Note that the constraint $z_j \leq w_j$ makes sure that z_j is equal to 1 only if point j is covered at least twice. A different backup coverage model is the Backup Double Coverage model of Başar et al. (2009). In their model a demand point is covered when at least two opened bases are nearby, where the second base can be further away. A similar but more general approach is the Multiple Coverage LSCM presented in Batta and Mannur (1990), which will be treated below. Church and Gerrard (2003) also consider a generalisation of LSCM where multi-level coverage is required.

As the name indicates, the Multiple Coverage LSCM by Batta and Mannur (1990) is based on the LSCM (see Model 2.2.6). The decision variables y_i are the number of (identical) ambulances to assign at base $i \in \mathcal{I}$, implicitly opening base i when $y_i > 0$. The coverage constraints are as follows. For each demand point $j \in \mathcal{J}$ we have an upper bound on the number of vehicles needed to cover this point, denoted by $\beta_j \in \mathbb{N}$. Order the required vehicles from 1 to β_j , where vehicle 1 is the nearest and vehicle β_j the farthest vehicle from point j. For covering point j, the first vehicle must be within a certain time limit, e.g., 15 minutes. Similarly, the k-th vehicle has a certain time limit, where $k \in \{1, \ldots, \beta_j\}$. Which bases satisfy these time limits is encoded into $a_{ij}^k \in \mathbb{B}$ for $i \in \mathcal{I}$. Note that the coverage constraints assume that the time limits are non-decreasing in k.



Model 2.2.6: Multiple Coverage LSCM.

Additional constraints on the number of available vehicles are required for models based on MCLP. Common constraints are:

$$\sum_{i \in \mathcal{I}} y_i \le q,$$
$$y_i \le q_i x_i \qquad \qquad \forall i \in \mathcal{I}.$$

The parameter $q \in \mathbb{N}$ denotes the total number of available ambulances and the parameters $q_i \in \mathbb{N}$ the allowed maximum number of vehicles assigned to base $i \in \mathcal{I}$. Of course, if a base is closed, no vehicles can be assigned to it.

A model that combines several of the mentioned model extensions is the Double Standard model (DSM) by Gendreau et al. (1997), see Model 2.2.7. The objective is to maximise demand covered by two vehicles within 15 minutes, such that a fraction of $\alpha \in [0, 1]$ of the total demand is covered at least by one vehicle within 15 minutes. Furthermore, all demand points must be covered within 30 minutes. At most $p \in \mathbb{N}$ bases can be opened and each base $i \in \mathcal{I}$ can accommodate at most $q_i \in \mathbb{N}$ ambulances. The total number of ambulances is limited to $q \in \mathbb{N}$.

Maximise

$$\sum_{j \in \mathcal{J}} d_j z_j$$

subject to

$$\begin{split} \sum_{i \in \mathcal{I}} x_i &\leq p, \\ \sum_{i \in \mathcal{I}} y_i &\leq q, \\ y_i &\leq q_i x_i & \forall i \in \mathcal{I}, \\ \sum_{i \in \mathcal{I}} a_{ij}^{c_{15}} y_i &\geq w_j + z_j & \forall j \in \mathcal{J}, \\ \sum_{i \in \mathcal{I}} a_{ij}^{c_{30}} y_i &\geq 1 & \forall j \in \mathcal{J}, \\ \sum_{i \in \mathcal{I}} d_j w_j &\geq \alpha \sum_{j \in \mathcal{J}} d_j, \\ x_i &\in \mathbb{B} & \forall i \in \mathcal{I}, \\ y_i &\in \mathbb{N} & \forall i \in \mathcal{I}, \\ w_j, z_j &\in \mathbb{B} & \forall j \in \mathcal{J}. \end{split}$$

Model 2.2.7: Double Standard model (DSM).

A variation on the DSM is the DSM with Limited Coverage Capacity by Doerner et al. (2005). It modifies two coverage constraints into soft constraints by penalising the objective for violations of these constraints (similar to Lagrangian relaxation). Furthermore, the model balances the assignment of demand to the bases with a balance parameter $\tau \in \mathbb{R}_{>0}$:

$$\frac{d_j}{\sum_{i\in\mathcal{I}}a_{ij}y_i}\leq\tau\qquad\qquad\forall j\in\mathcal{J}.$$

In fact, this constraint is also relaxed and deviations are penalised in the objective.

We have discussed several coverage constraint from the literature. Many can be placed within a general framework for coverage, which we will describe in the next section.

2.2.3 General Coverage Constraints

In order to cover demand point $j \in \mathcal{J}$ certain conditions have to be satisfied, such as proximity to opened bases. We assume that these conditions are defined in terms of elementary properties of the ambulance bases \mathcal{I} . In addition, these elementary properties should be independent of the decisions made in the location problem, hence they can be determined a priori in a preprocessing phase. With these assumptions we can derive a general formulation for a broad class of coverage constraints. An example of a common elementary property is whether emergency vehicles from base $i \in \mathcal{I}$ can reach point j within a certain amount of time, e.g., within 15 minutes. As before, we define $a_{ij} \in \mathbb{B}$ as follows:

$$a_{ij} = \begin{cases} 1 & \text{if point } j \text{ can be reached by base } i \text{ within 15 minutes} \\ 0 & \text{otherwise} \end{cases}$$
(2.2.3)

Looije (2013) considers the elementary property whether a demand point can be reached within time followed by a trip to hospital, where the total time cannot exceed a predefined threshold. Another common elementary property is whether a base has certain medical facilities. In general, we have a set \mathcal{R} of elementary properties and for each $r \in \mathcal{R}$ we can define $a_{ij}^r \in \mathbb{B}$ with $i \in \mathcal{I}$ and $j \in \mathcal{J}$ such that

$$a_{ij}^r = \begin{cases} 1 & \text{if property } r \text{ holds for base } i \text{ and point } j \\ 0 & \text{otherwise} \end{cases}.$$

From these elementary properties we can construct a set C of elementary conditions for covering point $j \in \mathcal{J}$, which are of the form

$$\sum_{i\in\mathcal{I}} a_{ij}^{r(c)} x_i \ge b_j^c,\tag{2.2.4}$$

where $b_j^c \in \mathbb{N}_{\geq 1}$ and $r(c) \in \mathcal{R}$ is the corresponding elementary property of elementary condition $c \in \mathcal{C}$. For example, we may require that at least two opened bases are within 15 minutes of point j. With a_{ij} defined as in Equation (2.2.3), the condition can be formulated as

$$\sum_{i\in\mathcal{I}}a_{ij}x_i\geq 2.$$

For each elementary condition c based on property r(c) we introduce a binary variable $w_j^c \in \mathbb{B}$. This variable is equal to one if and only if the elementary condition (2.2.4) is satisfied:

$$\sum_{i \in \mathcal{I}} a_{ij}^{r(c)} x_i \ge b_j^c w_j^c,$$

$$\sum_{i \in \mathcal{I}} a_{ij}^{r(c)} x_i \le (b_j^c - 1) + M w_j^c,$$

$$w_i^c \in \mathbb{B}.$$
(2.2.5)

Equation (2.2.5) is required to force $w_j^c = 1$ if Equation (2.2.4) holds. Here, $M \in \mathbb{R}_{\geq 0}$ is a suitably large constant (the 'big-M'). It is often sufficient to take M = p or M = p + q. Do note that using $(b_j^c - 1)$ is valid, since the left-hand side of Equation (2.2.5) is integral.

To handle complements of elementary conditions (i.e., to model 'smaller than'-inequalities), we introduce the complementary variable $w_i^{\overline{c}} \in \mathbb{B}$ for each w_i^c , with the additional condition that

$$w_i^c + w_i^{\overline{c}} = 1 \qquad \qquad \forall c \in \mathcal{C}, j \in \mathcal{J}.$$

Hence, we can define the set $\overline{\mathcal{C}} = \{\overline{c} : c \in \mathcal{C}\}$ of complementary conditions. For a coherent notation we need to introduce \dot{c} , which means that it is either a standard condition ($\dot{c} = c \in \mathcal{C}$) or a complementary condition ($\dot{c} = \overline{c} \in \overline{\mathcal{C}}$).

More complex conditions for coverage can be constructed using logical formulas with w_j^c as a literal, e.g., consider the abstract logical condition based on certain elementary conditions c_1 , c_2 , c_3 , and c_4 :

$$w_j^{c_1} \vee (w_j^{c_2} \wedge (\neg w_j^{c_3} \vee w_j^{c_4})).$$
(2.2.6)

As is known, all logical formulas can be translated to equisatisfiable formulas in conjunctive normal form by introducing dummy literals. For example, by introducing dummy literal ζ we can rewrite (2.2.6) to

$$(w_j^{c_1} \lor \zeta) \land (\neg \zeta \lor w_j^{c_2}) \land (\neg \zeta \lor \neg w_j^{c_3} \lor w_j^{c_4}).$$

Thus, we can assume that each coverage condition is in conjunctive normal form, taking into account that the set of elementary conditions also contains dummies. Each dummy condition $c \in C$ is associated with a dummy property $r(c) \in \mathcal{R}$. Of course, we also add the complementary variable as described above. Therefore, each condition is a conjunction of clauses, where each clause is a finite disjunction of literals.

For demand point $j \in \mathcal{J}$ we denote the overall coverage condition by the set $\mathcal{F}_j \subseteq 2^{\mathcal{C} \cup \overline{\mathcal{C}}}$ with clauses as elements, where $2^{\mathcal{C} \cup \overline{\mathcal{C}}}$ denotes the power set of $\mathcal{C} \cup \overline{\mathcal{C}}$. Each clause $f \in \mathcal{F}_j$ consists of the disjunction of (complementary) elementary conditions (encoded in w_j^c and $w_j^{\overline{c}}$). Recall that an elementary condition $\dot{c} \in f$ can be either standard ($\dot{c} = c \in \mathcal{C}$) or complementary ($\dot{c} = \overline{c} \in \overline{\mathcal{C}}$). For given $x_i \in \mathbb{B}$ ($i \in \mathcal{I}$), the coverage of point $j \in \mathcal{J}$ can now be determined as follows. First, check each elementary condition:

$$\begin{split} &\sum_{i \in \mathcal{I}} a_{ij}^{r(c)} x_i \geq b_j^c w_j^c \qquad \qquad \forall c \in \mathcal{C}, \\ &\sum_{i \in \mathcal{I}} a_{ij}^{r(c)} x_i \leq (b_j^c - 1) + M w_j^c \qquad \qquad \forall c \in \mathcal{C}, \\ & w_j^c + w_j^{\overline{c}} = 1 \qquad \qquad \forall c \in \mathcal{C}, \\ & w_j^c, w_j^{\overline{c}} \in \mathbb{B} \qquad \qquad \forall c \in \mathcal{C}. \end{split}$$

Notice that we check complementary elementary conditions by evaluating their standard counterpart. Next, determine if all clauses of the coverage condition are satisfied, that is, if point j is covered:

$$\sum_{\substack{\dot{c} \in f}} w_j^{\dot{c}} \ge z_j \qquad \qquad \forall f \in \mathcal{F}_j,$$
$$z_j \in \mathbb{B}.$$

These coverage constraints can be generalised further by incorporating the variable $y_i \in \mathbb{N}$ with $i \in \mathcal{I}$, where y_i equals the number of emergency vehicles placed at base i (as usual). We can construct similar coverage conditions as those based on x_i and include them into the model. To differentiate between elementary properties based on the ambulance base and those based on the number of vehicles, we use $a_{ij}^{r_x(c)}$ and $a_{ij}^{r_y(c)}$, respectively. Thus, the new constraints are of the form:

$$\sum_{i \in \mathcal{I}} \left(a_{ij}^{r_x(c)} x_i + a_{ij}^{r_y(c)} y_i \right) \ge b_j^c w_j^c \qquad \forall c \in \mathcal{C}, j \in \mathcal{J}$$

In conclusion, we have the general coverage constraints:

$$\begin{split} \sum_{i \in \mathcal{I}} \left(a_{ij}^{r_x(c)} x_i + a_{ij}^{r_y(c)} y_i \right) &\geq b_j^c w_j^c \qquad \forall c \in \mathcal{C}, j \in \mathcal{J}, \\ \sum_{i \in \mathcal{I}} \left(a_{ij}^{r_x(c)} x_i + a_{ij}^{r_y(c)} y_i \right) &\leq (b_j^c - 1) + M w_j^c \qquad \forall c \in \mathcal{C}, j \in \mathcal{J}, \end{split}$$

$$\begin{aligned} & w_j^c + w_j^{\overline{c}} = 1 \qquad \forall c \in \mathcal{C}, j \in \mathcal{J}, \\ & \sum_{\substack{c \in f}} w_j^{\underline{c}} \geq z_j \qquad \forall f \in \mathcal{F}_j, j \in \mathcal{J}, \\ & w_j^c, w_j^{\overline{c}} \in \mathbb{B} \qquad \forall c \in \mathcal{C}, j \in \mathcal{J}. \end{split}$$

These constraints can be used to construct initial models which can be analysed and reformulated, leading to simplifications and more computationally efficient models. For example, Equation (2.2.7) can often be omitted.

2.2.4 Survival and Decay Models

As mentioned when discussing coverage neglect, the 'all-or-nothing' coverage makes no distinction within each type of coverage (covered or uncovered). In reality, being able to respond within 5 instead of 15 minutes significantly improves the survival rates of the patients. A model that uses 'all-or-nothing' coverage cannot incorporate this distinction. As a result, so-called survival models have been developed.

Many survival models explicitly assign demand points to bases and can be viewed as modifications of the *p*-Median model. An early modification by Toregas et al. (1971) of the *p*-Median model for EMS is shown in Model 2.2.8. The decision variables u_{ij} denote the fraction of demand of point $j \in \mathcal{J}$ assigned to base $i \in \mathcal{I}$. Such an assignment has an associated cost $\gamma_{ij} \in \mathbb{R}$. All demand of a point must be allocated to opened bases by Equations (2.2.8) and (2.2.9).

$$\begin{array}{l} \text{Minimise} \\ & \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \gamma_{ij} u_{ij} \\ \text{subject to} \\ & \sum_{i \in \mathcal{I}} x_i \leq p, \\ & \sum_{i \in \mathcal{I}} a_{ij} u_{ij} = 1 \qquad \forall j \in \mathcal{J}, \qquad (2.2.8) \\ & u_{ij} \leq x_i \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, \\ & u_{ij} \in [0, 1] \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, \\ & x_i \in \mathbb{B} \qquad \forall i \in \mathcal{I}. \end{array}$$

Model 2.2.8: Modified *p*-Median Model.

An example of a survival model is the Maximum Survival Location problem by Erkut et al. (2008). It is similar to Model 2.2.8, but formulated as a maximisation problem with objective

$$\sum_{j \in \mathcal{J}} d_j \sum_{i \in \mathcal{I}} \phi_{ij} u_{ij}.$$

The covered demand is weighted by a survival function $\phi : \mathbb{R} \to [0, 1]$, that is, a decay or nonincreasing function. For example, ϕ can be based on response times, where longer response times result in lower survival values. Each assignment u_{ij} has a corresponding survival weight ϕ_{ij} . Furthermore, Equation (2.2.8) is replaced by

$$\sum_{i \in \mathcal{I}} u_{ij} = 1 \qquad \qquad \forall j \in \mathcal{J},$$

as $\phi_{ij} = 0$ if $a_{ij} = 0$. A similar approach is the Partial Coverage MCLP by O. Karasakal and E. Karasakal (2004).

The idea of incorporating a decay function in an EMS model also resulted in Cooperative models, see Berman et al. (2011). Although these models still use 'all-or-nothing' coverage, a decay is embedded in the coverage constraints. In Cooperative models an opened base emits a 'signal' ϕ with decaying intensity over distance (response time). Each demand point receives signals from all opened bases and has a certain signal intensity threshold τ : the demand point is covered only if the total signal intensity exceeds this threshold. See Model 2.2.9 for the Cooperative MCLP. For similar models, see Berman et al. (2010).



Model 2.2.9: Cooperative MCLP.

2.3 Probabilistic EMS Models

The deterministic models do not take into account the busy times (the limited availability) of ambulances. This is only valid if there are surplus ambulances available at each base. However, in reality this would lead to unreasonably high costs and therefore the total number of ambulances is limited. The class of probabilistic EMS models take into account the limited availability of ambulances by considering their busy fractions and reliability. These models assume stationary distributions for the arrival of calls and consider limiting properties of queues.

The busy fraction $\rho \in [0, 1]$ of an ambulance is the long-term fraction of time an ambulance is busy handling emergency calls, i.e., not available to respond to new calls. It is common in EMS to assume that calls arrive according to independent Poisson processes and service is done according to a first-come, first-served (FCFS) policy. No calls are lost. For instance, suppose a base has incoming calls according to a Poisson process with arrival rate $\lambda \in \mathbb{R}_{\geq 0}$. At the base $k \in \mathbb{N}$ ambulances are stationed that need on average $\mu \in \mathbb{R}_{\geq 0}$ time to handle a call, with $\lambda \mu \leq k$. Results for M/G/k queues imply that the busy fraction of such ambulances is:

$$\rho = \frac{\lambda \mu}{k}.$$

Note that a system is stable only if $\rho < 1$. There are also EMS models where the FCFS policy is relaxed. For instance, Silva and Serra (2008) consider priority levels for emergency calls.

The Maximum Expected Covering Location problem (MEXCLP) by Daskin (1983) extends the MCLP by including the ambulance busy fraction. The model assumes independently operating ambulances and a fixed busy fraction $\rho \in (0, 1)$ for all ambulances, independent of the assigned base. The MEXCLP tends to overestimate coverage due to the server independence assumption. See Model 2.3.1 for the mathematical formulation.

Maximise

$$\sum_{j \in \mathcal{J}} d_j \sum_{k=1}^q (1-\rho) \rho^{k-1} z_j^k$$

subject to

$$\begin{split} \sum_{i \in \mathcal{I}} x_i &\leq p, \\ \sum_{i \in \mathcal{I}} y_i &\leq q, \\ y_i &\leq q_i x_i \qquad \forall i \in \mathcal{I}, \\ \sum_{i \in \mathcal{I}} a_{ij} y_i &\geq \sum_{k=1}^q z_j^k \qquad \forall j \in \mathcal{J}, \\ x_i \in \mathbb{B} \qquad \forall i \in \mathcal{I}, \\ y_i \in \mathbb{N} \qquad \forall i \in \mathcal{I}, \\ z_j^k \in \mathbb{B} \qquad \forall j \in \mathcal{J}, k \in \{1, \dots, q\}. \end{split}$$

Model 2.3.1: Maximum Expected Covering Location problem (MEXCLP).

The variables x_i and y_i denote whether base $i \in \mathcal{I}$ is opened and the number of stationed vehicles at the base. Both are limited by parameters p, q and q_i . As usual, $a_{ij} \in \mathbb{B}$ indicates if base $i \in \mathcal{I}$ can cover demand point $j \in \mathcal{J}$. Associated to each demand point $j \in \mathcal{J}$ are variables $z_j^k \in \mathbb{B}$, where $z_j^k = 1$ only if at least k ambulances cover point j. As the total number of ambulances is limited to q, we have $k \in \{1, \ldots, q\}$.

The probability that all k ambulances near a point $j \in \mathcal{J}$ are busy is given by ρ^k . Hence, the expected demand coverage for point j is $d_j(1-\rho^k)$. Increasing the number of ambulances near j from k-1 to k results in a marginal gain of $d_j(1-\rho^k) - d_j(1-\rho^{k-1}) = d_j(1-\rho)\rho^{k-1}$. The objective is equal to demand multiplied by the sum of these marginal gains.

For any optimal solution it holds that $z_j^{k+1} = 1$ implies $z_j^k = 1$ for $k \in \{1, \ldots, q-1\}$. Therefore, we do not have to include the constraints $z_j^{k+1} \leq z_j^k$. Also, for $\rho \in (0, 1)$ the optimal solution satisfies the capacity constraints with equality.

In a similar way, the Maximum Expected Survival Location problem (MEXSLP) in Erkut et al. (2008) incorporates busy fractions into EMS survival models.

2.3.1 Hypercube Queueing Correction

The Hypercube model is presented by Larson (1974) and gives theoretical performance measures for a queueing system. The model assumes independent Poisson arrivals of calls and Exponential service times, independent of base assignment or call origin (demand point). Only one ambulance is required to handle a call. Furthermore, each demand point has a preferred service order of bases, e.g., based on distance.

Suppose there are in total $q \in \mathbb{N}$ ambulances stationed. The Hypercube model uses 2^q states (idle or busy ambulance) for which the steady state probabilities are determined. From these probabilities various performance measures can be determined. The Hypercube model can be incorporated into an optimisation procedure, but is limited to small number of q for tractability reasons.

Larson (1975) provides an approximation for the Hypercube model, called A-Hypercube, which is based on the fact that the Hypercube states resemble the states of $M/M/q/\infty$ queues. The steady state distribution is well-known for these queues. Consider the case that a call has to be assigned to an ambulance. Instead of using the preferred base order, a random order is used. If the selected ambulance is busy, a new random ambulance is selected without replacement. The probability that the k-th selected ambulance is the first free (idle) ambulance is given by:

$$Q(q, \rho, k-1)(1-\rho)\rho^{k-1}$$

The factor Q is called the A-Hypercube Queueing correction factor, defined as:

$$Q(q,\rho,k) = \frac{\sum_{k'=k}^{q-1} q^{k'} \rho^{k'-k} (q-k-1)! (q-k')/q! (k'-k)!}{(1-\rho) \sum_{k'=0}^{q-1} \rho^{k'} (q^{k'}/k'!) + q^q \rho^q/q!} \quad \forall k \in \{0,\dots,q-1\}.$$
(2.3.1)

These factors correct the ambulance busy fractions, which are based on independent service. Correction factors based on M/M/q/q queues are also given in Larson (1975). Note that these correction factors are approximations, as the preference order is not random in reality.

There are several extensions to the Hypercube and A-Hypercube model. Burwell et al. (1993) consider preference order ties in the Hypercube model. General service times are added to the A-Hypercube model by Jarvis (1985). Other modifications of the Hypercube model are shown in Geroliminis, Karlaftis, and Skabardonis (2006), Geroliminis, Karlaftis, Stathopoulos, et al. (2004), and Iannoni and Morabito (2007).

The Adjusted MEXCLP by Batta, Dolan, et al. (1989) incorporates the A-Hypercube correction factors into the MEXCLP. The only difference with MEXCLP is the objective. The new objective is given by

$$\sum_{j \in \mathcal{J}} d_j \sum_{k=1}^{q} Q(q, \rho, k-1)(1-\rho) \rho^{k-1} z_j^k,$$

where Q is the A-Hypercube correction factor defined in Equation (2.3.1). McLay (2009) extends the Adjusted MEXCLP with two types of servers.

2.3.2 Stochastic Response Times and Preferred Bases

Besides server dependence, uncertainty in the response times can be considered. Most of the preceding models used a binary parameter a_{ij} to indicate whether base $i \in \mathcal{I}$ covers demand point $j \in \mathcal{J}$. Instead of this binary parameter we can use the probability $p_{ij} \in [0, 1]$ that the response time is less than a certain threshold. Note that this does require remodelling of the coverage constraints (where summing over probabilities makes no sense). Using probabilities to capture uncertain response times is somewhat similar to the EMS survival models.

A non-linear model that uses stochastic response times is the MEXCLP modification by Goldberg, Dietrich, et al. (1990). This model also includes preferred bases for each demand point. See Model 2.3.2 for the formulation. For the derivation it is useful to define the bijective mapping $\pi : \{1, \ldots, p\} \times \mathcal{J} \to \mathcal{I}$ that denotes the preferred bases for each point. That is, $\pi(k, j)$ is the *k*-th preferred base of point *j*. Suppose the busy fractions ρ_i of base $i \in \mathcal{I}$ are given, where a base is busy if all its ambulances are. The probability that an ambulance from the *k*-th preferred base responds to a call from point *j* is equal to the probability that the *k*-th preferred base is available (not busy) and more preferred bases are busy:

$$(1 - \rho_{\pi(k,j)}) \prod_{k'=1}^{k-1} \rho_{\pi(k',j)}.$$

This term is multiplied by the probability $p_{\pi(k,j)j}$ that an available ambulance at base $\pi(k,j)$ will reach point j in time. However, the base preferences are decision variables in this model. Thus, the function π is unknown in advance. The binary decision variable u_{ij}^k indicates if opened base $i \in \mathcal{I}$ is the k-th preferred open base of point $j \in \mathcal{J}$. For each $k \in \{1, \ldots, p\}$ at most one base can be the k-th preferred base of point j. Furthermore, each base can have only one preference number with respect to demand point j.

This leads to the following adjustment of the probability:

$$\sum_{i \in \mathcal{I}} p_{ij} (1 - \rho_i) u_{ij}^k \prod_{k'=1}^{k-1} \sum_{i' \in \mathcal{I}} \rho_{i'} u_{i'j}^{k'}.$$

This formulation is valid, since the constraints on the decision variables u_{ij}^k imply that any feasible solution has a corresponding bijection π .

The busy fractions ρ_i are also variables and determined by non-linear equations. The model assumes that the busy fractions are independent of the state of the underlying queueing system. The stochastic response times are also ignored and average service times are used. Let λ_j be the arrival rate of calls from point j and $\mu_{ij} \in \mathbb{R}_{\geq 0}$ be the average service time of base i for a call from point j. Assume that each ambulance base has a fixed total service time $\tau \in \mathbb{R}_{\geq 0}$ available. The busy fractions must satisfy:

$$\rho_{i} = \sum_{j \in \mathcal{J}} \frac{\lambda_{j} \mu_{ij}}{\tau} \sum_{k=1}^{p} (1 - \rho_{i}) u_{ij}^{k} \prod_{k'=1}^{k-1} \sum_{i' \in \mathcal{I}} \rho_{i'} u_{i'j}^{k'},$$

which is similar to the terms in the objective.

If a predefined preference order exists (e.g., for point j base i is preferred to base i'), it can be included by adding the constraint

$$\sum_{k=1}^{p} k u_{ij}^{k} \le \sum_{k=1}^{p} k u_{i'j}^{k} + p(1 - x_{i'}).$$

Similar constraints are used in Borrás and Pastor (2002). An extension with allocation of ambulances is given in Goldberg and Paz (1991).

Maximise

$$\sum_{j \in \mathcal{J}} d_j \sum_{k=1}^p \sum_{i \in \mathcal{I}} p_{ij} (1-\rho_i) u_{ij}^k \prod_{k'=1}^{k-1} \sum_{i' \in \mathcal{I}} \rho_{i'} u_{i'j}^{k'}$$

subject to

$$\begin{split} &\sum_{i\in\mathcal{I}} x_i \leq p, \\ &\sum_{i\in\mathcal{I}} u_{ij}^k \leq 1 & \forall j\in\mathcal{J}, k\in\{1,\dots,p\}, \\ &\sum_{k=1}^p u_{ij}^k \leq 1 & \forall i\in\mathcal{I}, j\in\mathcal{J}, \\ &u_{ij}^k \leq x_i & \forall i\in\mathcal{I}, j\in\mathcal{J}, k\in\{1,\dots,p\}, \\ &\rho_i = \sum_{j\in\mathcal{J}} \frac{\lambda_j \mu_{ij}}{\tau} \sum_{k=1}^p (1-\rho_i) u_{ij}^k \prod_{k'=1}^{k-1} \sum_{i'\in\mathcal{I}} \rho_{i'} u_{i'j}^{k'} & \forall i\in\mathcal{I}, \\ &\rho_i \in [0,1] & \forall i\in\mathcal{I}, \\ &u_{ij}^k \in \mathbb{B} & \forall i\in\mathcal{I}, j\in\mathcal{J}, k\in\{1,\dots,p\}, \\ &x_i\in\mathbb{B} & \forall i\in\mathcal{I}. \end{split}$$

Model 2.3.2: Non-linear MEXCLP with Stochastic Response Times.
A similar non-linear model is described in Ingolfsson et al. (2008). A linear variation is the model in Kommer, Zuzáková, et al. (2012) where preferred ambulances instead of preferred bases are considered. The assignment of ambulances to bases is included in the model and the decision variables $u_{ij}^k \in \mathbb{B}$ indicate whether for point j the k-th preferred ambulance is located at base i. A fixed busy fraction is assumed for all ambulances, independent of the allocation to bases. Model 2.3.3 shows the mathematical model.

Maximise

$$\sum_{j \in \mathcal{J}} d_j \sum_{k=1}^q (1-\rho) \rho^{k-1} \sum_{i \in \mathcal{I}} p_{ij} u_{ij}^k$$

subject to

$$\begin{split} &\sum_{i \in \mathcal{I}} x_i \leq p, \\ &\sum_{i \in \mathcal{I}} y_i \leq q, \\ &y_i \leq q_i x_i & \forall i \in \mathcal{I}, \\ &\sum_{i \in \mathcal{I}} u_{ij}^k \leq 1 & \forall j \in \mathcal{J}, k \in \{1, \dots, q\}, \\ &\sum_{k=1}^p u_{ij}^k \leq y_i & \forall i \in \mathcal{I}, j \in \mathcal{J}, \\ &u_{ij}^k \in \mathbb{B} & \forall i \in \mathcal{I}, j \in \mathcal{J}, k \in \{1, \dots, q\} \\ &x_i \in \mathbb{B} & \forall i \in \mathcal{I}, \\ &y_i \in \mathbb{N} & \forall i \in \mathcal{I}. \end{split}$$

Model 2.3.3: MEXCLP with Stochastic Response Times.

Goal Programming variants with stochastic response times have also been developed. For instance, the model in Alsalloum and Rand (2006) includes constraints such as

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}d_jp_{ij}u_{ij} + w_0^- = \sum_{j\in\mathcal{J}}d_j,$$
(2.3.2)

and

$$\sum_{k=1}^{q} \tau^{k} y_{i}^{k} - \sum_{j \in \mathcal{J}} d_{j} u_{ij} - w_{i}^{+} = 0 \qquad \forall i \in \mathcal{I}.$$
(2.3.3)

The decision variable $u_{ij} \in [0, 1]$ denotes the fraction of demand from point $j \in \mathcal{J}$ allocated to base $i \in \mathcal{I}$. In Equation (2.3.2), $w_0^- \in \mathbb{R}_{\geq 0}$ is equal to the uncovered demand and is penalised in the objective. Furthermore, $y_i^k \in \mathbb{B}$ indicates if there are at least k vehicles stationed at base i. The parameter $\tau^k \in \mathbb{R}_{\geq 0}$ is the additional demand a base can cover when increasing the number of assigned ambulances at a base from (k-1) to k. The variable $w_i^+ \in \mathbb{R}_{\geq 0}$ in Equation (2.3.3) keeps track of unused capacity at base i and is penalised as well.

2.3.3 Reliability Constraints

Busy fractions of ambulances can be incorporated implicitly by considering the reliability of service. Let $\alpha \in [0, 1]$ be the desired service reliability, i.e., the probability that an ambulance is available when an emergency accident occurs. Suppose we are given a fixed busy fraction ρ for all ambulances. If k vehicles cover point j, the probability that no ambulances are available is ρ^k . For the required reliability, we have assign vehicles to bases that cover point j such that

$$1 - \rho^{\sum_{i \in \mathcal{I}} a_{ij} y_i} \ge \alpha \qquad \Longleftrightarrow \qquad \sum_{i \in \mathcal{I}} a_{ij} y_i \ge \left| \frac{\log (1 - \alpha)}{\log \rho} \right|.$$

Here, the operator $\lceil \cdot \rceil$ returns the smallest integer greater than or equal to the argument. If the busy fraction is not known, the following regional estimation can be used. Let $\mathcal{J}_j \subseteq \mathcal{J}$ be the demand points sufficiently near $j \in \mathcal{J}$ (including j), depending on some metric. Estimate the busy fraction for point $j \in \mathcal{J}$ by

$$\rho_j = \frac{\mu \sum_{j' \in \mathcal{J}_j} \lambda_{j'}}{\tau \sum_{i \in \mathcal{I}} a_{ij} y_i}.$$

The numerator is the total amount of work near point j and the denominator the service capacity near point j. The parameter $\mu \in \mathbb{R}_{\geq 0}$ is the average service time for an arbitrary call, $\lambda_j \in \mathbb{R}_{\geq 0}$ the arrival rate of calls from point j and $\tau \in \mathbb{R}_{\geq 0}$ the service capacity of one ambulance. See also ReVelle and Hogan (1989b). A combination of this estimate and the reliability constraint leads to

$$\sum_{i \in \mathcal{I}} a_{ij} y_i \ge \beta_j,$$
$$1 - \left(\frac{\mu \sum_{j' \in \mathcal{J}_j} \lambda_{j'}}{\tau \beta_j}\right)^{\beta_j} \ge \alpha.$$

These constraints are defined for each demand point $j \in \mathcal{J}$. The parameter β_j can be determined numerically a priori by simply increasing its value and checking the reliability constraint.

The Maximum Availability Location problem (MALP) by ReVelle and Hogan (1989a) incorporates these reliability constraints. See Model 2.3.4, where β_j is defined either for a fixed busy fraction or estimated as shown above.

Another way to determine parameters β_j is given in Marianov and ReVelle (1994) and Marianov and ReVelle (1996). They assume that demand varies little between adjacent base regions and that travel times are insignificant with respect to service times. Thus, flows between regions cancel each other and each region is isolated. The system is modelled to have Poisson arrivals, Exponentially distributed service times, and loss of calls (no queue). Let $\rho_j = \lambda_j/\mu_j$, where μ_j is the average service time for calls from point j. For reliability α the parameter β_j is the smallest integer such that

$$\frac{\frac{1}{\beta_j!}\rho_j^{\beta_j}}{\sum_{k=0}^{\beta_j}\frac{1}{k!}\rho_j^k} \le 1 - \alpha$$

The left-hand side of this inequality is the steady-state probability that no ambulances are available near point j (the Erlang loss formula).

Maximise

$$\sum_{j \in \mathcal{J}} d_j z_j$$
subject to

$$\sum_{i \in \mathcal{I}} x_i \leq p,$$

$$\sum_{i \in \mathcal{I}} y_i \leq q,$$

$$y_i \leq q_i x_i \qquad \forall i \in \mathcal{I},$$

$$\sum_{i \in \mathcal{I}} a_{ij} y_i \geq \beta_j z_j \qquad \forall j \in \mathcal{J},$$

$$x_i \in \mathbb{B} \qquad \forall i \in \mathcal{I},$$

$$y_i \in \mathbb{N} \qquad \forall i \in \mathcal{I},$$

$$z_j \in \mathbb{B} \qquad \forall j \in \mathcal{J}.$$
(2.3.4)

Model 2.3.4: Maximum Availability Location problem (MALP).

The Extended Maximum Availability Location problem (EMALP) by Galvão et al. (2005) replaces the coverage constraint of the MALP model (Equation (2.3.4)) by a non-linear coverage constraint:

$$\left(\left(1 - \prod_{\{i \in \mathcal{I}: a_{ij} = 1\}} \rho_i^{y_i} Q\left(q, \rho, \sum_{i' \in \mathcal{I}} a_{i'j} y_{i'} - 1\right) \right) - \alpha \right) z_j \ge 0 \qquad \forall j \in \mathcal{J}.$$

This expression evaluates whether the service reliability is at least α . For each base $i \in \mathcal{I}$ there is a busy fraction ρ_i for the assigned ambulances, with average systemwide busy fraction ρ . The factors Q are the A-Hypercube queueing correction factors defined in Equation (2.3.1), with $Q(q, \rho, -1)$ defined as zero.

Instead of using busy fractions to define reliability levels, reliability can be expressed in terms of probabilities of serviced calls. An example is the Reliability Perspective model (Rel-P) by Ball and Lin (1993), see Model 2.3.5. The model assumes that the number of calls handled by each base during a certain time period is uncertain. Furthermore, it is independent of the allocation of ambulances to bases and of calls at other bases. Define the discrete cumulative distribution function F_i such that $F_i(k)$ is the probability that $k \in \mathbb{N}$ or less ambulances are sufficient to handle the calls of base $i \in \mathcal{I}$. That is, $(1 - F_i(k))$ is the probability that k+1 or more ambulances are needed.

The decision variable y_i^k indicates if exactly k vehicles are stationed at base $i \in \mathcal{I}$. Assigning k ambulances to a base has associated cost γ_i^k . As reliability constraint we have:

$$\prod_{\{i \in \mathcal{I}: a_{ij}=1\}} \prod_{k=1}^{q_i} \left(1 - F_i(k)\right)^{y_i^k} \le 1 - \alpha \qquad \forall j \in \mathcal{J}$$

That is, the probability that the arriving calls of all nearby bases exceed the base capacities is less than $(1 - \alpha)$. These constraints can be linearised by taking the logarithm:

$$\sum_{i \in \mathcal{I}} a_{ij} \sum_{k=1}^{q_i} \log \left(1 - F_i(k)\right) y_i^k \le \log \left(1 - \alpha\right) \qquad \forall j \in \mathcal{J}.$$

Minimise

$$\sum_{i\in\mathcal{I}}\sum_{k=1}^{q_i}\gamma_i^ky_i^k$$

subject to

$$\begin{split} \sum_{i \in \mathcal{I}} x_i &\leq p, \\ y_i^k &\leq x_i & \forall k \in \{1, \dots, q_i\}, i \in \mathcal{I}, \\ \sum_{k=1}^{q_i} y_i^k &\leq 1 & \forall i \in \mathcal{I}, \\ \sum_{i \in \mathcal{I}} a_{ij} \sum_{k=1}^{q_i} \log (1 - F_i(k)) y_i^k &\leq \log (1 - \alpha) & \forall j \in \mathcal{J}, \\ x_i \in \mathbb{B} & \forall i \in \mathcal{I}, \\ y_i^k \in \mathbb{B} & \forall k \in \{1, \dots, q_i\}, j \in \mathcal{J}. \end{split}$$

Model 2.3.5: Reliability Perspective model (Rel-P).

When assuming that the number of calls from a demand point are uncertain with given cumulative distribution function, we can apply the model from Beraldi, Bruni, and D. Conforti (2004). The demand at a point is assumed to be independent of the other points. The model allocates ambulances at bases to each demand point in such a way that reliable service is guaranteed. Again the required number of ambulances is determined using the cumulative demand distribution and linearised using logarithms.

Other uses of reliability are the Local Reliability MEXCLP (LR-MEXCLP) by Sorensen and Church (2010) and the Hierarchical Queueing LSCM (HiQ-LSCM) by Marianov and Serra (2001). The Local Reliability MEXCLP (LR-MEXCLP) by Sorensen and Church (2010) is similar to MEXCLP, only the busy fraction coefficients are replaced by service reliability coefficients. This approach can also be viewed as a survival model.

The Hierarchical Queueing LSCM (HiQ-LSCM) by Marianov and Serra (2001) uses two hierarchical levels of bases (low and high level), where a proportion of demand handled by low level bases is referred to high level bases. The reliability constraints enforce that the probability that a call enters a queue with a length above a certain threshold is at most α . To determine these constraints, steady state probabilities of M/M/c queues are used (where $c \in \mathbb{N}$ is the number of servers).

2.3.4 Loss Models

Two recent EMS models by Restrepo (2008) try to minimise the expected number of calls for which the response time is too long and are lost. In particular, the models use a queueing system with no queues allowed. The calls occurring when all ambulances are busy are considered to be lost. For performance measures this is reasonable if short response times are required, which is the case in EMS.

The first model is the Island model, see Model 2.3.6. The model assumes that portions of demand are allocated a priori to each base, resulting in call arrival rates λ_i for base $i \in \mathcal{I}$. Furthermore, average service time for calls handled by base i is set to μ_i . The non-linear coefficient in the objective is the Erlang loss formula, which we have also encountered in reliability constraints.

An extension is the Overflow model of Restrepo (2008), where the assignment of demand is a decision variable. That is, λ_i are variables and the following constraints are added

$$\sum_{i \in \mathcal{I}} \lambda_i = \Lambda,$$
$$\lambda_i \in \mathbb{R}_{\geq 0} \qquad \forall i \in \mathcal{I},$$

where $\Lambda \in \mathbb{R}_{\geq 0}$ is the total demand.

Minimise			
	$\sum_{i \in \mathcal{I}} \lambda_i \left(\frac{\frac{1}{y_i!}}{\frac{1}{k!} \sum_{i=1}^{k}} \right)$	$\left(rac{(\lambda_i\mu_i)^{y_i}}{\sum\limits_{k=0}^{y_i}(\lambda_i\mu_i)^k} ight)$	
subject to			
	$\sum_{i \in \mathcal{I}} x_i \le p,$		
	$\sum_{i \in \mathcal{I}} y_i \le q,$		
	$y_i \le q_i x_i$	$\forall i \in \mathcal{I},$	
	$x_i \in \mathbb{B}$	$\forall i \in \mathcal{I},$	
	$u_i \in \mathbb{N}$	$\forall i \in \mathcal{I}.$	

Model 2.3.6: Island model.

2.3.5 Stochastic Programming Models

When facing uncertainty, a valid strategy is to discern several scenarios for which we want to determine optimal EMS designs. For uncertain demand, the probability distribution can be approximated by a finite set of scenarios S. The number of scenarios should not be extremely large, as computational tractability issues arise. Use of decomposition methods is usually beneficial.

The Two-Stage Stochastic LSCM model by Beraldi and Bruni (2009) applies Stochastic Programming with uncertain demand in the field of EMS. The model discerns two stages: in the first stage the opened bases and the number of stationed ambulances need to be determined, in the second stage demand points are assigned to bases. See Model 2.3.7 for the mathematical formulation.

As mentioned, we have a set of scenarios S with associated probability $p_s \in [0, 1]$ for $s \in S$. The decision variable $u_{ij}^s \in \mathbb{B}$ indicates whether demand point $j \in \mathcal{J}$ is assigned to (opened) base $i \in \mathcal{I}$ under scenario $s \in S$. Furthermore, each vehicle can handle $\tau \in \mathbb{R}_{>0}$ calls.

The model has two important constraints: all demand must be assigned to bases and all demand must be serviced. That is, the assigned demand to a base $i \in \mathcal{I}$ cannot exceed its service capacity (determined by τy_i). These constraints are modelled as probabilistic constraints, i.e., they must hold with probability $\alpha \in [0, 1]$. Therefore, the variable $w_s \in \mathbb{B}$ is introduced to keep track of whether the constraints hold (0) or not (1) for scenario $s \in S$. This is done using 'big-M' parameter $M \in \mathbb{R}_{\geq 0}$. The following constraint enforces the required reliability:

$$\sum_{s \in \mathcal{S}} p_s w_s \le (1 - \alpha).$$

The objective is to minimise expected costs, where each decision has a certain cost $\gamma_i^1(x_i)$, $\gamma_i^2(y_i)$, or $\gamma_{ij}^3(u_{ij}^s)$ for $i \in \mathcal{I}$ and $j \in \mathcal{J}$. A similar approach is shown in Noyan (2010).

Minimise

$$\sum_{i \in \mathcal{I}} (\gamma_i^1 x_i + \gamma_i^2 y_i) + \sum_{s \in \mathcal{S}} p_s \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \gamma_{ij}^3 u_{ij}^s$$

subject to

$$y_{i} \leq q_{i}x_{i} \qquad \forall i \in \mathcal{I},$$

$$\sum_{j \in \mathcal{J}} \lambda_{i}^{s} a_{ij} u_{ij}^{s} \leq \tau y_{i} + M w_{s} \qquad \forall i \in \mathcal{I}, s \in \mathcal{S},$$

$$\sum_{i \in \mathcal{I}} a_{ij} u_{ij}^{s} + M w^{s} \geq 1 \qquad \forall j \in \mathcal{J}, s \in \mathcal{S},$$

$$u_{ij}^{s} \leq x_{i} \qquad \forall i \in \mathcal{I}, s \in \mathcal{S},$$

$$\sum_{s \in \mathcal{S}} p_{s} w_{s} \leq (1 - \alpha),$$

$$u_{ij}^{s} \in \mathbb{B} \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, s \in \mathcal{S},$$

$$w_{s} \in \mathbb{B} \qquad \forall s \in \mathcal{S},$$

$$x_{i} \in \mathbb{B} \qquad \forall i \in \mathcal{I},$$

$$y_{i} \in \mathbb{N} \qquad \forall i \in \mathcal{I}.$$

Model 2.3.7: Two-Stage Stochastic LSCM model.

A different approach is the Maximal Expected Coverage Relocation problem (MECRP) discussed by Gendreau et al. (2006), see Model 2.3.8. In this model the allowed number of opened bases is uncertain. It evaluates (p+1) scenarios, $S = \{0, \ldots, p\}$, where each scenario $s \in S$ corresponds to the case that s bases are opened. The probability of the scenarios are based on the availability of $p \in \mathbb{N}$ opened bases, where each base has a given busy fraction $\rho \in [0, 1]$. That is, the probability of scenario $s \in S$ is given by the probability that (p - s) bases are busy:

$$p_s = \binom{p}{s} (1-\rho)^s \rho^{p-s}.$$

The parameter $\beta^s \in \{0, \ldots, s\}$ restricts the number of opened bases during scenario s that are closed during scenario s + 1. That is, changing which base should be opened is restricted by Equations (2.3.5), (2.3.6), (2.3.7), and (2.3.8).

Maximise

$$\sum_{s=0}^{p} {p \choose s} (1-\rho)^s \rho^{p-s} \sum_{j \in \mathcal{J}} d_j z_j^s$$

subject to

$\sum x_i^s = s$	$\forall s \in \{0, \dots, p\},$	(2.3.5)
$i \subset \mathcal{T}$		

$\sum_{i \in \mathcal{I}} a_{ij} x_i^s \ge z_j^s$	$\forall j \in \mathcal{J}, s \in \{0, \dots, p\},\$	
$x_i^s - x_i^{s+1} \le u_i^s$	$\forall i \in \mathcal{I}, s \in \{1, \dots, p-1\},\$	(2.3.6)
$\sum u_i^s \leq \beta^s$	$\forall s \in \{1, \dots, p-1\},\$	(2.3.7)

$$\forall s \in \{1, \dots, p-1\},\tag{2.3.7}$$

$$u_i^s \in \mathbb{B} \qquad \forall i \in \mathcal{I}, s \in \{1, \dots, p-1\},$$

$$x_i^s \in \mathbb{B} \qquad \forall i \in \mathcal{I}, s \in \{0, \dots, p\},$$

$$z_j^s \in \mathbb{B} \qquad \forall j \in \mathcal{J}, s \in \{0, \dots, p\}.$$

$$(2.3.8)$$

Model 2.3.8: Maximal Expected Coverage Relocation problem (MECRP).

2.4 Multiple Vehicle Types

Many of the shown EMS models in Sections 2.2 and 2.3 can be extended to incorporate multiple vehicle types. Ambulances can be differentiated according to possible medical support: Basic Life Support (BLS) and Advanced Life Support (ALS). The kind of vehicle can also be distinguished, such as motorcycles and cars.

In the mathematical formulations it is often sufficient to copy the constraints and variables for each vehicle type. For example, see the Facility Location and Equipment Emplacement Technique model (FLEET) by Schilling et al. (1979) in Model 2.4.1. FLEET is a multi-vehicle formulation for the MCLP. The set of vehicles is denoted by \mathcal{K} and the decision variable $x_i^k \in \mathbb{B}$ denotes whether a base for vehicles of type $k \in \mathcal{K}$ is opened at base $i \in \mathcal{I}$. At most $p^k \in \mathbb{N}$ of such bases can be opened. Furthermore, each type k vehicle base has its own coverage characteristics $a_{ij}^k \in \mathbb{B}$. A demand point is considered to be covered if it is covered by at least one vehicle of each type.

Other modifications are the Tandem Equipment Allocation model (TEAM) and Multi-Objective Tandem Equipment Allocation model (MOTEAM) by Schilling et al. (1979). These models add the constraint that there is a certain order in which vehicle bases can be opened:

$$x_i^k \le x_i^{k+1} \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}.$$

Looije (2013) considers the MCLP with two vehicle types. A demand point $j \in \mathcal{J}$ is covered when either an ambulance is within 15 minutes $(w_j^{c(k_1,1)} = 1)$ or a rapid responder is nearby $(w_j^{c(k_2)} = 1)$ followed up by an ambulance further away $(w_j^{c(k_1,2)} = 1)$. This is captured by the constraints:

$$w_j^{c(k_1,1)} + \frac{1}{2} \left(w_j^{c(k_1,2)} + w_j^{c(k_2)} \right) \ge z_j \qquad \forall j \in \mathcal{J}.$$

Similar coverage constraints are considered in Mandell (1998).

Maximise

$$\sum_{j \in \mathcal{J}} d_j z_j$$

subject to

$$\sum_{i \in \mathcal{I}} x_i^k \le p^k \qquad \forall k \in \mathcal{K},$$
$$\sum_{i \in \mathcal{I}} a_{ij}^k x_i^k \ge z_j \qquad \forall j \in \mathcal{J}, k \in \mathcal{K},$$
$$x_i \in \mathbb{B} \qquad \forall i \in \mathcal{I},$$
$$z_j \in \mathbb{B} \qquad \forall j \in \mathcal{J}.$$

Model 2.4.1: Facility Location and Equipment Emplacement Technique (FLEET).

2.5 Multiple Time Periods

a

Another general extension for EMS models is the inclusion of multiple time periods, which is similar to Stochastic Programming approaches but has a natural (chronological) order in the scenarios. Suppose we have a set \mathcal{T} of successive time periods. For each time period we have a classic EMS model, but the allocation of ambulances cannot differ too much from the previous time period. That is, the number of ambulance relocations is limited or any relocation is penalised. The opened bases are usually fixed over all periods, however a similar approach as with ambulances is possible.

Examples of these multiple time periods models are the Time Dependent Travel Times MEXCLP (TIMEXCLP) by Repede and Bernardo (1994) and the Dynamic Available Coverage Location model (DACL) by Rajagopalan et al. (2008). The TIMEXCLP is based on the MEXCLP and the DACL model on the EMALP. The Multi-hour Service System Design problem by Amiri (2001) considers allocation of demand points to bases per time period and penalises queueing delay. The Dynamic Double Standard model (DDSM) by Gendreau et al. (2001) is based on the DSM and considers decisions one time period ahead. It adds the following constraints to the DSM:

$$\begin{split} \sum_{k=1}^{i} v_i^k &= y_i & \forall i \in \mathcal{I}, \\ \sum_{i \in \mathcal{I}} v_i^k &= 1 & \forall k \in \{1, \dots, q\}, \\ v_i^k &\in \mathbb{B} & \forall i \in \mathcal{I}, k \in \{1, \dots, q\}. \end{split}$$

The variable $v_i^k \in \mathbb{B}$ indicates if ambulance $k \in \{1, \ldots, q\}$ is moved to base $i \in \mathcal{I}$ in the next time period. The costs for this relocation depend on the situation of the current time period.

Suppose that the time-dependent parameters have a common cycle time of $T \in \mathbb{N}$. Let the set of time periods be given by $\mathcal{T} = \{1, \ldots, T\}$, after which the cycle repeats. The Time-dependent MEXCLP with Start-up and Relocation Costs by Van den Berg and Aardal (2013) considers a cyclic time period MEXCLP extension. See Model 2.5.1 for the formulation. Almost all parameters and variables of the MEXCLP are replaced by their time-dependent version. New are the flow constraints:

$$y_i^t + \sum_{i' \in \mathcal{I}} \left(v_{i'i}^t - v_{ii'}^t \right) = y_i^{t+1} \qquad \forall i \in \mathcal{I}, t \in \mathcal{T} \setminus \{T\}$$
$$y_i^T + \sum_{i' \in \mathcal{I}} \left(v_{i'i}^T - v_{ii'}^T \right) = y_i^1 \qquad \forall i \in \mathcal{I}.$$

The variable $v_{ii'}^t \in \mathbb{N}$ is equal to the number of ambulances repositioned from base *i* to base *i'* between time periods *t* and *t* + 1. Note that this does not allow the total number of ambulances to vary per period (which is why we fix the number of ambulances to *q*).

To incorporate a varying number of ambulances, an artificial base should be added to the flow constraints. However, for the shown objective it can also be achieved by aggregating the variables $v_{ii'}^t$ into ingoing $(v_{i+}^t \in \mathbb{N})$ and outgoing $(v_{i-}^t \in \mathbb{N})$ relocations:

$$\begin{aligned} y_i^t + v_{i+}^t - v_{i-}^t &= y_i^{t+1} & \forall i \in \mathcal{I}, t \in \mathcal{T} \setminus \{T\} \\ y_i^T + v_{i+}^T - v_{i-}^T &= y_i^1 & \forall i \in \mathcal{I}. \end{aligned}$$

Maximise

$$\sum_{t \in \mathcal{T}} \left(\sum_{j \in \mathcal{J}} d_j^t \sum_{k=1}^q (1 - \rho^t) (\rho^t)^{k-1} z_j^{k,t} - \gamma_1 \sum_{i \in \mathcal{I}} \sum_{i' \in \mathcal{I}} v_{ii'}^t \right) - \gamma_2 \sum_{i \in \mathcal{I}} x_i$$

subject to

$$\begin{split} \sum_{i\in\mathcal{I}} x_i &\leq p, \\ \sum_{i\in\mathcal{I}} y_i^t &= q \qquad \forall t\in\mathcal{T}, \\ y_i^t &\leq q_i^t x_i \qquad \forall i\in\mathcal{I}, t\in\mathcal{T}, \\ \sum_{i\in\mathcal{I}} a_{ij}^t y_i^t &\geq \sum_{k=1}^q z_j^{k,t} \qquad \forall j\in\mathcal{J}, t\in\mathcal{T}, \\ y_i^t &+ \sum_{i'\in\mathcal{I}} \left(v_{i'i}^t - v_{ii'}^t\right) &= y_i^{t+1} \qquad \forall i\in\mathcal{I}, t\in\mathcal{T}\setminus\{T\}, \\ y_i^T &+ \sum_{i'\in\mathcal{I}} \left(v_{i'i}^T - v_{ii'}^T\right) &= y_i^1 \qquad \forall i\in\mathcal{I}, \\ v_{ii'}^t &\in \mathbb{N} \qquad \forall i, i'\in\mathcal{I}, t\in\mathcal{T}, \\ x_i\in\mathbb{B} \qquad \forall i\in\mathcal{I}, \\ y_i^t &\in \mathbb{N} \qquad \forall i\in\mathcal{I}, t\in\mathcal{T}, \\ z_i^{k,t}\in\mathbb{B} \qquad \forall j\in\mathcal{J}, k\in\{1,\ldots,q\}, t\in\mathcal{T}. \end{split}$$

Model 2.5.1: Time-dependent MEXCLP with Start-up and Relocation Costs.

The Time-dependent MEXCLP with Start-up and Relocation Costs is based on the Multi-period DSM (mDSM) by Schmid and Doerner (2010). They modify the DSM to include time periods and add explicit assignment of demand to bases. Let $u_{ij}^t \in \mathbb{N}$ be the demand from point $j \in \mathcal{J}$ allocated to base $i \in \mathcal{I}$ at time period $t \in \mathcal{T}$. The parameter $\lambda_j^t \in \mathbb{N}$ is the arrival rate of calls from point $j \in \mathcal{J}$ at time $t \in \mathcal{T}$. Furthermore, each vehicle can handle at most $\tau \in \mathbb{N}$ calls per time period. The added assignment constraints with vehicle capacity are:

$$\begin{split} \sum_{i \in \mathcal{I}} a_{ij}^t u_{ij}^t &= \lambda_j^t & \forall j \in \mathcal{J}, t \in \mathcal{T}, \\ \sum_{j \in \mathcal{J}} a_{ij}^t u_{ij}^t &\leq \tau y_i^t & \forall i \in \mathcal{I}, t \in \mathcal{T}, \\ u_{ij}^t \in \mathbb{N} & \forall i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T}. \end{split}$$

The discussed models have natural variants where certain parameters and constraints are changed to span either a single time period or multiple periods.

2.6 Conclusion

Selecting or developing the 'right' emergency medical service (EMS) model is not a trivial task. More complex models are not always superior to simpler variants, even though they are (usually) more realistic. In general, the added realism requires more parameters to be estimated, which also makes a parameter sensitivity analysis more difficult. Furthermore, solving complex models requires new solution methods or significantly increases the required computational time.

The strength of simple EMS models is that they are comprehensible and manageable. However, simple models can lack key features required to be useful for any decision making process. The selection of the appropriate model should therefore balance the model complexity and the desired decision making goals. Preferably, multiple EMS models should be used to see if any significant discrepancies occur. The analysis can be supported by simulations.

We have discussed a selection of EMS models ranging from simplistic linear models to more complex non-linear models. For example, one of the basic models is the Maximal Covering Location problem (MCLP). The MCLP assumes that coverage satisfies an 'all-or-nothing' relation, it disregards uncovered demand points, and does not consider the unavailability of ambulances.

Many EMS models extend the MCLP by adding features to address these assumptions and limitations. The 'deterministic' additions include regional coverage constraints, backup coverage, and a more general survival objective function. The 'probabilistic' extensions include busy fractions of ambulances, stochastic response times, reliability of service, and Stochastic Programming. Furthermore, most models can be modified to incorporate multiple vehicle types and multiple time periods.

The discussed models are not a complete overview of all EMS models, as we have focussed on coverage models. For example, other types of EMS models are based on the *p*-median or *p*-centre model. Moreover, we have primarily considered medical facility location literature. The (general or non-medical) Facility Location problem is applied to other research areas as well. Model extensions or solution methods from these other research areas can be useful for EMS models.

In the following chapters we focus on the Maximal Covering Location problem as EMS model. Due to its simplicity we can perform an extensive sensitivity analysis and analyse the performance of solution methods, both numerically and theoretically.

Chapter 3

Sensitivity Analysis for the MCLP

Contents

3.1	Intro	oduction	50
3.2	Rob	ust Optimisation	51
	3.2.1	Uncertainty Structure	51
	3.2.2	Robust Counterpart	52
	3.2.3	Other Uncertainty Structures	56
	3.2.4	Robust MCLP	56
	3.2.5	Other Robust Counterparts	61
3.3	Post	-Optimisation Neighbourhood Analysis	63
	3.3.1	Intractable Neighbourhoods	64
3.4	Nun	nerical Results	66
	3.4.1	Realistic Instances	66
	3.4.2	Number of Bases	69
	3.4.3	Adjacency Parameters	74
	3.4.4	Post-Optimisation Analysis	82
	3.4.5	Uncertain Demand	90
3.5	Two	Dutch Regional Ambulance Services in Detail	94
	3.5.1	Gooi- en Vechtstreek	94
	3.5.2	Limburg-Noord	100
3.6	Con	clusion	106

3.1 Introduction

When choosing which emergency medical service (EMS) model to use as a decision support tool, the features, underlying assumptions, and limitations of the model are not the only aspects to consider. Its complexity (both mathematical and computational) and comprehensibility are important as well. One aspect that is often disregarded is the robustness of the model with respect to data uncertainty: do small perturbations in the data impact the quality of solutions?

We will consider the sensitivity of the Maximal Covering Location problem (MCLP) to data uncertainty. Recall that for the MCLP at most $p \in \mathbb{N}$ bases can be opened (with no costs associated) and demand $d_j \in \mathbb{R}_{\geq 0}$ is non-negative for all demand points $j \in \mathcal{J}$. For easy reference, we restate the mathematical formulation below. We will analyse the sensitivity with respect to the demand d_j , the allowed number of bases p, and the adjacency parameters a_{ij} .



In Section 3.2 we consider demand that is unknown, but assumed to lie between a lower and an upper bound, where the total demand is fixed and known. We apply a worst-case optimisation technique called Robust Optimisation to construct a reformulated MCLP that is robust to the predefined demand uncertainty. That is, any solution of the reformulated model is feasible for any realisation of the demand and the optimal solution has the best worst-case coverage.

Robust Optimisation is a general technique, which also allows for other uncertainty parameters and structures. Therefore, we will derive a robust formulation for a general constraint, for which we show additional properties of the optimal solution. In Section 3.2.5 we give examples of other robust formulations using the general results.

In Section 3.4 the sensitivity to the allowed number of opened bases p is analysed by iterating over multiple values. A similar approach is used for the coverage parameter a_{ij} . Often these parameters follow from a response time threshold (e.g., within 15 minutes). By iteratively decreasing the threshold we can quantify the effect of uncertainty in the coverage parameter.

In addition to the three described types of data sensitivity, we will consider the uniqueness of solutions. Section 3.3 describes a post-optimisation neighbourhood analysis method to give an indication of the uniqueness of a solution and whether valid alternative solutions exist. All discussed methods are numerically evaluated in Sections 3.4 and 3.5.

3.2 Robust Optimisation

Robust Optimisation is a general approach to construct solutions of optimisation problems that remain feasible under parameter uncertainty. If the objective coefficients contain uncertainty, robust solutions are often suboptimal for all possible realisations separately, but optimal when considering all scenarios simultaneously. For an overview of different Robust Optimisation techniques and recent advances, see Ben-Tal, El Ghaoui, et al. (2009), Bertsimas et al. (2011), and Gabrel et al. (2013). A related technique is Recoverable Robustness by Liebchen et al. (2009) that integrates robust planning and recovery after realisation of the scenarios into one framework.

For applications of Robust Optimisation to the Facility Location problem we refer to Baron et al. (2011), Gülpmar et al. (2013), and Snyder (2006). Remarkably, Robust Optimisation (with the exception of Stochastic Programming) is seldom mentioned in the context of EMS facility location. See for example Zhang and Jiang (2013). Related is Ben-Tal, Chung, et al. (2011) where EMS logistics planning in supply chains is considered.

From the available Robust Optimisation techniques we will consider the Robust Counterpart approach of Ben-Tal, El Ghaoui, et al. (2009). This approach constructs robust solutions that are feasible for any realisation of the uncertain parameters (with specified bounds on the possible parameter values). These solutions are obtained by modifying the original optimisation problem into its so-called Robust Counterpart. Hence, it can be seen as a worst-case robust solution. Since the survival of patients is at stake, this strict approach is suitable.

As mentioned above, Robust Optimisation will be applied to uncertain demand. We assume that this uncertainty is uncorrelated to a certain degree: demand at each point has a known lower and upper bound, and the total demand is fixed and known. More details are given in Section 3.2.1. This uncertainty structure can also be applied to other parameters. Therefore, we first derive a general Robust Counterpart in Section 3.2.2, followed by the Robust Counterpart of the MCLP in Section 3.2.4.

3.2.1 Uncertainty Structure

Suppose the value of a certain parameter δ_j for each point $j \in \mathcal{J}$ is not known exactly, but can deviate from the estimated value $\hat{\delta}_j$. Furthermore, assume that the possible deviation is symmetric around the estimated value:

$$\delta_j \in \left[\hat{\delta}_j - \eta_j, \hat{\delta}_j + \eta_j\right],$$

where $\hat{\delta}_j \in \mathbb{R}$ and $\eta_j \in \mathbb{R}_{\geq 0}$. No particular probability distribution for δ_j is considered, but the robust approach can be seen as using a uniform probability distribution. The sum of the parameters δ_j is assumed to be known with great accuracy and can be considered as a known constant Δ :

$$\Delta = \sum_{j \in \mathcal{J}} \delta_j \in \left[\sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right), \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j + \eta_j \right) \right].$$

With these assumptions, we can structure the uncertainty into useful formulations. Let us describe the parameter uncertainty using perturbation variables $\zeta \in \mathbb{R}^{|\mathcal{J}|}$. That is,

$$\delta_j = \hat{\delta}_j + \eta_j \zeta_j \qquad \qquad \forall j \in \mathcal{J},$$

where we impose constraints on the perturbation. Clearly, $-1 \leq \zeta_j \leq 1$ must hold, but the total perturbation is also bounded. Let $\Xi = \Delta - \sum_{j \in \mathcal{J}} \hat{\delta}_j$. We require that

$$\sum_{j\in\mathcal{J}}\hat{\delta}_j + \sum_{j\in\mathcal{J}}\eta_j\zeta_j = \sum_{j\in\mathcal{J}}\left(\hat{\delta}_j + \eta_j\zeta_j\right) = \sum_{j\in\mathcal{J}}\delta_j = \Delta,$$

or equivalently,

$$\sum_{j \in \mathcal{J}} \eta_j \zeta_j = \Delta - \sum_{j \in \mathcal{J}} \hat{\delta}_j = \Xi.$$

Define the perturbation set $\mathcal{Z} \subseteq \mathbb{R}^{|\mathcal{J}|}$ as follows:

$$\mathcal{Z} = \left\{ \zeta \in \mathbb{R}^{|\mathcal{J}|} : -1 \le \zeta \le 1, \sum_{j \in \mathcal{J}} \eta_j \zeta_j = \Xi \right\}.$$
(3.2.1)

The parameter uncertainty can now be described using the uncertainty set \mathcal{D} , given by

$$\mathcal{D} = \left\{ \delta \in \mathbb{R}^{|\mathcal{J}|} : \delta_j = \hat{\delta}_j + \eta_j \zeta_j, j \in \mathcal{J}, \zeta \in \mathcal{Z} \right\}.$$
(3.2.2)

This uncertainty set is a natural structure when almost no information is available on the uncertainty. The values $\hat{\delta}_j$ are the estimated values that would be used in standard deterministic optimisation. The parameters η_j can be defined such that the possible deviation matches with, for instance, a 5% deviation from the estimate. Given this uncertainty set, we can construct robust versions of common location problems and constraints. Note that we do not assume that $\delta_j \geq 0$ for all $j \in \mathcal{J}$, allowing a wider range of applications of this uncertainty structure.

3.2.2 Robust Counterpart

Consider a general worst-case robust constraint of the following form:

$$\sum_{j \in \mathcal{J}} \delta_j \phi_j \ge \theta \qquad \qquad \forall \, \delta \in \mathcal{D}, \tag{3.2.3}$$

where $\theta \in \mathbb{R}$ and $\phi_j \in \mathbb{R}$ $(j \in \mathcal{J})$ are given input. The uncertainty set \mathcal{D} is given by (3.2.2) with underlying perturbation set given by (3.2.1). We can rewrite Equation (3.2.3) by using the definition of δ_j and collecting all uncertain values:

$$\sum_{j \in \mathcal{J}} \delta_j \phi_j \ge \theta \qquad \qquad \forall \delta \in \mathcal{D}$$

$$\iff \qquad \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j + \eta_j \zeta_j \right) \phi_j \ge \theta \qquad \qquad \forall \zeta \in \mathcal{Z}$$

$$\iff \sum_{j \in \mathcal{J}} \eta_j \phi_j \zeta_j \ge \theta - \sum_{j \in \mathcal{J}} \hat{\delta}_j \phi_j \qquad \forall \zeta \in \mathcal{Z}$$

$$\iff \qquad \qquad \min_{\zeta \in \mathcal{Z}} \sum_{j \in \mathcal{J}} \eta_j \phi_j \zeta_j \ge \theta - \sum_{j \in \mathcal{J}} \hat{\delta}_j \phi_j$$

Hence, we must consider the optimisation problem $\theta_p^* = \min\{\sum_{j \in \mathcal{J}} \eta_j \phi_j \zeta_j : \zeta \in \mathcal{Z}\}$ to find the worst-case realisation of the parameters (and the worst-case constraint). The optimisation problem and its dual are

MinimiseMaximise
$$\sum_{j \in \mathcal{J}} \eta_j \phi_j \zeta_j$$
$$\Xi u - \sum_{j \in \mathcal{J}} \left(v_j^+ + v_j^- \right)$$
subject to
$$\Xi u - \sum_{j \in \mathcal{J}} \left(v_j^+ + v_j^- \right)$$
subject tosubject to
$$\sum_{j \in \mathcal{J}} \eta_j \zeta_j = \Xi,$$
$$\eta_j u - v_j^+ + v_j^- = \eta_j \phi_j \quad \forall j \in \mathcal{J},$$
$$\zeta_j \leq 1 \quad \forall j \in \mathcal{J},$$
$$u \in \mathbb{R},$$
$$-\zeta_j \leq 1 \quad \forall j \in \mathcal{J},$$
$$v_j^+, v_j^- \in \mathbb{R}_{\geq 0} \quad \forall j \in \mathcal{J}.$$

Let θ_d^* be the optimal objective value of the dual problem. Notice that both problems are feasible. If dual feasible variables (u, v^+, v^-) satisfy $\Xi u - \sum_{j \in \mathcal{J}} (v_j^+ + v_j^-) \ge \theta - \sum_{j \in \mathcal{J}} \hat{\delta}_j \phi_j$, then by Strong Duality

 $\forall j \in \mathcal{J}.$

 $\zeta_i \in \mathbb{R}$

$$\theta_p^* = \theta_d^* \ge \Xi u - \sum_{j \in \mathcal{J}} \left(v_j^+ + v_j^- \right) \ge \theta - \sum_{j \in \mathcal{J}} \hat{\delta}_j \phi_j,$$

and Equation (3.2.3) holds. Vice versa, if Equation (3.2.3) holds, then again by Strong Duality there are dual feasible variables (u, v^+, v^-) such that $\Xi u - \sum_{j \in \mathcal{J}} (v_j^+ + v_j^-) \ge \theta - \sum_{j \in \mathcal{J}} \hat{\delta}_j \phi_j$ (e.g., the optimal dual solution). Therefore, the Robust Counterpart of (3.2.3) is given by

$$\Xi u - \sum_{j \in \mathcal{J}} \left(v_j^+ + v_j^- \right) \ge \theta - \sum_{j \in \mathcal{J}} \hat{\delta}_j \phi_j,$$

$$\eta_j u - v_j^+ + v_j^- = \eta_j \phi_j \qquad \forall j \in \mathcal{J},$$

$$u \in \mathbb{R},$$

$$v_j^+, v_j^- \in \mathbb{R}_{\ge 0} \qquad \forall j \in \mathcal{J}.$$

However, an equivalent formulation can be constructed that allows a clearer interpretation. Note that if $\eta_j = 0$ for some $j \in \mathcal{J}$ (no uncertainty), it is optimal to set $v_j^+ = v_j^- = 0$. Thus, we can ignore the Robust Counterpart constraints for j and treat it separately from the others. We can now safely rescale v_j^+ by replacing it by $\eta_j v_j^+$ (and similarly for v_j^-). Using the definition of Ξ and the Robust Counterpart constraints, we get the following equivalent formulation:

$$\Delta u - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j + \eta_j \right) v_j^+ + \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right) v_j^- \ge \theta,$$
$$u - v_j^+ + v_j^- = \phi_j \qquad \forall j \in \mathcal{J},$$
$$u \in \mathbb{R},$$
$$v_j^+, v_j^- \in \mathbb{R}_{\ge 0} \qquad \forall j \in \mathcal{J}.$$

The variables v_j^+ and v_j^- can be interpreted as, respectively, the overshoot and undershoot of u with respect to ϕ_j . Recall that $\Delta \geq \sum_{j \in \mathcal{J}} (\hat{\delta}_j - \eta_j)$ and $\Delta \leq \sum_{j \in \mathcal{J}} (\hat{\delta}_j + \eta_j)$. It is therefore never optimal to either overshoot for all points or undershoot for all points with variable u. Thus, it is never optimal to set $u < \min\{\phi_j : j \in \mathcal{J}\}$ or $u > \max\{\phi_j : j \in \mathcal{J}\}$. The constraints

$$\min\{\phi_j : j \in \mathcal{J}\} \le u \le \max\{\phi_j : j \in \mathcal{J}\}\$$

can be added to the model. Furthermore, it is never optimal to artificially increase both variables v_j^+ and v_j^- , since it trivially holds that $(\hat{\delta}_j + \eta_j) \ge (\hat{\delta}_j - \eta_j)$. Therefore, we can also add the constraints

$$v_j^+ \le \max\{\phi_{j'} : j' \in \mathcal{J}\} - \phi_j \qquad \forall j \in \mathcal{J}, \\ v_j^- \le \phi_j - \min\{\phi_{j'} : j' \in \mathcal{J}\} \qquad \forall j \in \mathcal{J}, \\ \forall j \in \mathcal{J}, \end{cases}$$

and (a linear formulation of)

$$v_j^+ v_j^- = 0 \qquad \qquad \forall j \in \mathcal{J}.$$

We will prove that even stronger constraints hold: u, v_j^+ and v_j^- can be restricted to finite discrete sets without loss of optimality, see Proposition 3.2.1.

Proposition 3.2.1. Consider the constraint

$$\sum_{j \in \mathcal{J}} \delta_j \phi_j \ge \theta \qquad \qquad \forall \, \delta \in \mathcal{D},$$

with uncertainty structure \mathcal{D} defined by (3.2.1) and (3.2.2), $\theta \in \mathbb{R}$ and $\phi_j \in \mathbb{R}$ for $j \in \mathcal{J}$. Its Robust Counterpart is given by

$$\Delta u - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j + \eta_j \right) v_j^+ + \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right) v_j^- \ge \theta,$$
$$u - v_j^+ + v_j^- = \phi_j \qquad \forall j \in \mathcal{J},$$
$$u \in \mathbb{R},$$
$$v_j^+, v_j^- \in \mathbb{R}_{\ge 0} \qquad \forall j \in \mathcal{J}.$$

Furthermore, the following properties hold for the optimal solution:

$u \in \{\phi_j : j \in \mathcal{J}\},\$	
$v_j^+ v_j^- = 0$	$\forall j \in \mathcal{J},$
$v_j^+ \in \{\phi_{j'} - \phi_j : \phi_{j'} \ge \phi_j, j' \in \mathcal{J}\}$	$\forall j \in \mathcal{J},$
$v_j^- \in \{\phi_j - \phi_{j'} : \phi_{j'} \le \phi_j, j' \in \mathcal{J}\}$	$\forall j \in \mathcal{J}.$

Proof. The derivation of the Robust Counterpart has been given above. Therefore, we consider the properties of the optimal solution, i.e., the domains of u, v_j^+ and v_j^- can be restricted to certain finite discrete sets. Let $\phi_{(1)} \leq \phi_{(2)} \leq \ldots \leq \phi_{(|\mathcal{J}|)}$ be the non-decreasing order of elements ϕ_j . Without loss of generality, we can assume that this order is strictly-increasing (only consider the unique values of ϕ_j , $j \in \mathcal{J}$). We will use that $v_j^+v_j^- = 0$ for all $j \in \mathcal{J}$.

Consider u when we restrict it to the left-open interval $(\phi_{(k)}, \phi_{(k+1)}]$ for some $k \in \{1, \ldots, |\mathcal{J}|-1\}$. For these values, u overshoots $\phi_{(1)}, \ldots, \phi_{(k)}$ and undershoots $\phi_{(k+1)}, \ldots, \phi_{(|\mathcal{J}|)}$. Note that there is no undershoot for $\phi_{(k+1)}$ if $u = \phi_{(k+1)}$. Therefore, we can state without loss of optimality that

$$\begin{cases} v_{(j)}^{+} = u - \phi_{(j)} \\ v_{(j)}^{-} = 0 \\ v_{(j)}^{+} = 0 \\ v_{(j)}^{-} = \phi_{(j)} - u \end{cases} \quad \forall j \in \{k + 1, \dots, |\mathcal{J}|\}.$$

Now consider the following constraint of the Robust Counterpart:

$$\Delta u - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j + \eta_j \right) v_j^+ + \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right) v_j^- \ge \theta.$$

It is optimal to choose u in such a way that the left-hand side is maximised. We can simplify the left-hand side to

$$\Phi(k,u) = \Delta u - \sum_{j=1}^{k} \left(\hat{\delta}_{(j)} + \eta_{(j)} \right) \left(u - \phi_{(j)} \right) + \sum_{j=k+1}^{|\mathcal{J}|} \left(\hat{\delta}_{(j)} - \eta_{(j)} \right) \left(\phi_{(j)} - u \right).$$

Since $\Phi(k, u)$ is a linear function in u, we only have to consider the subdomain boundaries for the extrema. Therefore, note that for $\epsilon \in (0, \phi_{(k+1)} - \phi_{(k)})$ small enough

$$\Phi(k,\phi_{(k+1)}) - \Phi(k,\phi_{(k)} + \epsilon) = (\phi_{(k+1)} - (\phi_{(k)} + \epsilon)) \left(\Delta - \sum_{j=1}^{k} \left(\hat{\delta}_{(j)} + \eta_{(j)} \right) - \sum_{j=k+1}^{|\mathcal{J}|} \left(\hat{\delta}_{(j)} - \eta_{(j)} \right) \right).$$
(3.2.4)

Furthermore, between two adjacent subdomains we have:

$$\Phi(k,\phi_{(k+1)}) - \Phi(k-1,\phi_{(k)}) = (\phi_{(k+1)} - \phi_{(k)}) \left(\Delta - \sum_{j=1}^{k} \left(\hat{\delta}_{(j)} + \eta_{(j)} \right) - \sum_{j=k+1}^{|\mathcal{J}|} \left(\hat{\delta}_{(j)} - \eta_{(j)} \right) \right).$$
(3.2.5)

For clarity, let $\Gamma(k) = \Delta - \sum_{j=1}^{k} (\hat{\delta}_{(j)} + \eta_{(j)}) - \sum_{j=k+1}^{|\mathcal{J}|} (\hat{\delta}_{(j)} - \eta_{(j)})$. Note that Γ is non-increasing in k: $\Gamma(k) \ge \Gamma(k+1)$. We have to distinguish the three cases of the sign of $\Gamma(k)$.

• $\Gamma(k) > 0$:

In this case is it optimal to set $u = \phi_{(k+1)}$ on the domain $(\phi_{(k)}, \phi_{(k+1)}]$, since it holds that $\Phi(k, \phi_{(k+1)}) - \Phi(k, \phi_{(k)} + \epsilon) > 0$.

• $\Gamma(k) = 0$:

Since $\Phi(k, \phi_{(k+1)}) - \Phi(k, \phi_{(k)} + \epsilon) = 0$, we have that $\Phi(k, u)$ is constant for u in the entire interval $(\phi_{(k)}, \phi_{(k+1)}]$ (non-unique maxima). We can simply choose $u = \phi_{(k+1)}$ to be optimal on $(\phi_{(k)}, \phi_{(k+1)}]$.

• $\Gamma(k) < 0$:

In this case is it optimal to set $u = \phi_{(k)} + \epsilon$, as $\Phi(k, \phi_{(k+1)}) - \Phi(k, \phi_{(k)} + \epsilon) < 0$. However, by combining (3.2.4) and (3.2.5) notice that $\Phi(k, \phi_{(k)} + \epsilon) - \Phi(k - 1, \phi_{(k)}) < 0$ as well. Therefore, limiting u to $(\phi_{(k)}, \phi_{(k+1)}]$ is not optimal overall and we can ignore the k-th subdomain. With these observations, we can conclude that the optimal u on the entire domain $[\phi_{(1)}, \phi_{(|\mathcal{J}|)}]$ is an element of the finite discrete set $\{\phi_j : j \in \mathcal{J}\}$. We have also shown how to find the optimal element by considering the sign of $\Gamma(k), k \in \{1, \ldots, |\mathcal{J}| - 1\}$. Consequently, the finite discrete sets for v_i^+ and v_j^- are trivial.

3.2.3 Other Uncertainty Structures

Robust Counterparts can be obtained for more complex uncertainty structures, such as convex cones, by using the general result stated in Ben-Tal, El Ghaoui, et al. (2009). The underlying method for the stated general Robust Counterpart is the same as shown in Section 3.2.2, but Conic Programming duality is used instead of LP duality. For instance, using ellipsoids as uncertainty structure results in a Quadratic Programming formulation.

We will mention a variation of the uncertainty set given in Section 3.2.1. We have a set of $|\mathcal{K}|$ subsets of $\mathcal{J}, \mathcal{J}_k \subseteq \mathcal{J}$ for $k \in \mathcal{K}$, not necessarily disjoint. For each subset \mathcal{J}_k the sum of the parameter δ_j with $j \in \mathcal{J}_k$ is set to Δ^k :

$$\Delta^{k} = \sum_{j \in \mathcal{J}_{k}} \delta_{j} \in \left[\sum_{j \in \mathcal{J}_{k}} \left(\hat{\delta}_{j} - \eta_{j} \right), \sum_{j \in \mathcal{J}_{k}} \left(\hat{\delta}_{j} + \eta_{j} \right) \right].$$

Care must be taken to ensure feasibility (the values of Δ^k cannot contradict each other). Define the sets $\mathcal{K}_j = \{k \in \mathcal{K} : j \in \mathcal{J}_k\}$ for each $j \in \mathcal{J}$. For this uncertainty set, the Robust Counterpart becomes

$$\sum_{k \in \mathcal{K}} \Delta^{k} u^{k} - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_{j} + \eta_{j} \right) v_{j}^{+} + \sum_{j \in \mathcal{J}} \left(\hat{\delta}_{j} - \eta_{j} \right) v_{j}^{-} \ge \theta,$$

$$\sum_{k \in \mathcal{K}_{j}} u^{k} - v_{j}^{+} + v_{j}^{-} = \phi_{j} \qquad \forall j \in \mathcal{J}, \qquad (3.2.6)$$

$$u^{k} \in \mathbb{R} \qquad \forall k \in \mathcal{K},$$

$$v_{j}^{+}, v_{j}^{-} \in \mathbb{R}_{\ge 0} \qquad \forall j \in \mathcal{J}.$$

In particular, if the subsets \mathcal{J}_k are disjoint, Equation (3.2.6) is disjoint for each $j \in \mathcal{J}$. This allows the use of decompositions methods such as the Dantzig-Wolfe decomposition (see Dantzig and Wolfe (1960)). Disjoint subsets of demand points, each with a fixed total demand, are useful when there are regions with independent demand.

3.2.4 Robust MCLP

Recall that the goal of the Maximal Covering Location problem is to open up to $p \in \mathbb{N}$ sites such that the covered demand is maximised. The Robust MCLP is similar to the MCLP, but it maximises the worst-case covered demand. Demand is assumed to be uncertain but bounded by a known lower and upper bound. We denote the uncertain demand by δ_j for point $j \in \mathcal{J}$. The uncertainty is uncorrelated to a certain degree as the total demand is assumed to be fixed to Δ . This leads to the uncertainty structure \mathcal{D} defined by (3.2.1) and (3.2.2). See Model 3.2.1 for the Robust MCLP.

Maximise		
heta		
subject to		
$ heta \leq \sum_{j \in \mathcal{J}} \delta_j z_j$	$\forall \delta \in \mathcal{D},$	(3.2.7)
$\sum_{i \in \mathcal{I}} x_i \le p,$		
$\sum_{i \in \mathcal{I}} a_{ij} x_i \ge z_j$	$\forall j \in \mathcal{J},$	
$x_i \in \mathbb{B}$	$\forall i \in \mathcal{I},$	
$z_j \in \mathbb{B}$	$\forall j \in \mathcal{J}.$	

Model 3.2.1: Robust Maximal Covering Location problem.

In order to construct the Robust Counterpart of the Robust MCLP, we only have to derive the Robust Counterpart of Equation (3.2.7). We can apply Proposition 3.2.1 with $\phi_j = z_j$ for $j \in \mathcal{J}$. The resulting model is the RC-MCLP, see Model 3.2.2.

Maximise

$$\Delta u - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j + \eta_j \right) v_j^+ + \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right) v_j^-$$
subject to

$$u - v_j^+ + v_j^- = z_j \qquad \forall j \in \mathcal{J},$$

$$\sum_{i \in \mathcal{I}} x_i \le p,$$

$$\sum_{i \in \mathcal{I}} a_{ij} x_i \ge z_j \qquad \forall j \in \mathcal{J},$$

$$u \in \mathbb{R},$$

$$v_j^+, v_j^- \in \mathbb{R}_{\ge 0} \qquad \forall j \in \mathcal{J},$$

$$x_i \in \mathbb{B} \qquad \forall i \in \mathcal{I},$$

$$z_j \in \mathbb{B} \qquad \forall j \in \mathcal{J}.$$

Model 3.2.2: Robust Counterpart of the Maximal Covering Location problem (RC-MCLP).

The variables v_j^+ and v_j^- can be interpreted as respectively the overshoot and undershoot of u with respect to z_j . We interpret the terms in the objective for different cases:

• If u = 0, it is optimal to set $v_j^+ = 0$ and $v_j^- = z_j$ for all $j \in \mathcal{J}$. The resulting objective value is $\sum_{j \in \mathcal{J}} (\hat{\delta}_j - \eta_j) z_j$, equal to the minimum possible (worst-case) demand covered by this choice of z_j .

- Similarly, if $0 \le u \le z_j$ for all $j \in \mathcal{J}$, $v_j^+ = 0$ and $v_j^- = z_j u$ are optimal. In this case, the term Δu can be seen as the fraction of total demand that is covered with certainty. The term $\sum_{j \in \mathcal{J}} (\hat{\delta}_j \eta_j) v_j^-$ is the worst-case demand covered with this choice of z_j .
- If $u > z_j$ for some j, variable u overshoots the actual coverage for point j and it is optimal to set $v_j^+ = u z_j$ and $v_j^- = 0$ for that point j. The term $\sum_{j \in \mathcal{J}} (\hat{\delta}_j + \eta_j) v_j^+$ corrects the covered demand by subtracting the best-case surplus demand covered by the overshoot. Hence, the resulting objective value is still a worst-case value.

The optional constraints on the domain of u, v_j^+ and v_j^- of Proposition 3.2.1 simplify to binary constraints. In fact, the binary restriction of u can be derived in a more direct way than shown in the proof of Proposition 3.2.1. Let $\mathcal{J}_1 = \{j \in \mathcal{J} : z_j = 1\}$ contain the covered points and $\mathcal{J}_0 = \mathcal{J} \setminus \mathcal{J}_1$ the uncovered points. We can rewrite the objective to

$$\begin{aligned} \Delta u &- \sum_{j \in \mathcal{J}_0} \left(\hat{\delta}_j + \eta_j \right) u + \sum_{j \in \mathcal{J}_1} \left(\hat{\delta}_j - \eta_j \right) (1 - u) \\ &= \left(\Delta - \sum_{j \in \mathcal{J}_0} \left(\hat{\delta}_j + \eta_j \right) - \sum_{j \in \mathcal{J}_1} \left(\hat{\delta}_j - \eta_j \right) \right) u + \sum_{j \in \mathcal{J}_1} \left(\hat{\delta}_j - \eta_j \right) \\ &= \left(\Delta - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j + \eta_j \right) (1 - z_j) - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right) z_j \right) u + \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right) z_j. \end{aligned}$$

Thus, the optimal value of u is:

$$u = \begin{cases} 1 & \text{if } \Delta - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j + \eta_j \right) (1 - z_j) - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right) z_j \ge 0\\ 0 & \text{if } \Delta - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j + \eta_j \right) (1 - z_j) - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right) z_j < 0 \end{cases}$$

This corresponds to the cases of $\Gamma(k)$ in the proof of Proposition 3.2.1.

We show that these results can also be obtained using intuitive reasoning. For given z_j $(j \in \mathcal{J})$, the worst-case realisation of the demand is the realisation where the covered points have the least possible demand. That is, covered points have minimum demand $\sum_{j \in \mathcal{J}_1} (\hat{\delta}_j - \eta_j)$ and uncovered points maximum demand $\sum_{j \in \mathcal{J}_0} (\hat{\delta}_j + \eta_j)$. However, the total demand is fixed to Δ , so a correction is needed.

The worst-case realisation of covered demand is:

$$\sum_{j \in \mathcal{J}_{1}} \left(\hat{\delta}_{j} - \eta_{j} \right) + \max \left\{ 0, \left(\Delta - \sum_{j \in \mathcal{J}_{1}} \left(\hat{\delta}_{j} - \eta_{j} \right) \right) - \sum_{j \in \mathcal{J}_{0}} \left(\hat{\delta}_{j} + \eta_{j} \right) \right\}$$
$$= \sum_{j \in \mathcal{J}} \left(\hat{\delta}_{j} - \eta_{j} \right) z_{j} + \max \left\{ 0, \Delta - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_{j} + \eta_{j} \right) (1 - z_{j}) - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_{j} - \eta_{j} \right) z_{j} \right\}, \quad (3.2.8)$$

corresponding exactly to our derived results for the RC-MCLP model. Therefore, an alternative model is:

Maximise

$$\sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right) z_j + u$$

subject to

 $\sum x_i < p_i$

$$u + \Delta v \le \Delta, \tag{3.2.9}$$
$$u - Mv \le \Delta - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j + \eta_j \right) (1 - z_j) - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right) z_j, \tag{3.2.10}$$

$$\sum_{i \in \mathcal{I}}^{i \in \mathcal{I}} a_{ij} x_i \ge z_j \qquad \qquad \forall j \in \mathcal{J},$$

$$egin{aligned} & u \in \mathbb{R}_{\geq 0}, \ & v \in \mathbb{B}, \ & x_i \in \mathbb{B} & & orall i \in \mathcal{I}, \ & z_j \in \mathbb{B} & & orall j \in \mathcal{J}. \end{aligned}$$

The variable u corresponds to the worst-case coverage in addition to the minimum coverage (the maximum in Equation (3.2.8)). The variable v is required to capture the case that the maximum in Equation (3.2.8) is zero, where we have used a 'big-M' formulation. A valid choice for M would be

$$M = 2\sum_{j\in\mathcal{J}}\eta_j,$$

since by definition of Δ :

$$-2\sum_{j\in\mathcal{J}}\eta_j = \sum_{j\in\mathcal{J}} \left(\hat{\delta}_j - \eta_j\right) - \sum_{j\in\mathcal{J}} \left(\hat{\delta}_j + \eta_j\right) \le \Delta - \sum_{j\in\mathcal{J}} \left(\hat{\delta}_j + \eta_j\right) (1 - z_j) - \sum_{j\in\mathcal{J}} \left(\hat{\delta}_j - \eta_j\right) z_j.$$

Equation (3.2.9) ensures us that uv = 0 (where $u \leq \Delta$ is a trivial non-tight upper bound). Consequently, the maximum in Equation (3.2.8) can be captured in constraint (3.2.10) and the objective function.

Both models are equivalent, as can be seen when conditioning on the optimal value of u (for given z_j) in either models. Substitution of these optimal values in the objective function results in the same model. The major difference between the two robust models is the number of additional constraints and variables. The Robust Counterpart model has at least $|\mathcal{J}|$ extra constraints and $2|\mathcal{J}| + 1$ additional variables, where the extra variables are either all continuous or all binary. The alternative model has 2 extra constraints for the Robust Counterpart model are independent of the demand parameters. This can be beneficial for solution methods.

3.2.4.1 A Common Uncertainty Structure

It can be difficult to obtain (empirical) data to determine the appropriate uncertainty structure. In that case, a common approach is to consider robust solutions for a constant relative 'measure error' $\epsilon \in [0, 1]$ (say 5%). That is, we set $\eta_j = \epsilon \hat{\delta}_j$ for $j \in \mathcal{J}$:

$$\delta_j \in \left[(1-\epsilon)\hat{\delta}_j, (1+\epsilon)\hat{\delta}_j \right]$$

We will show that the MCLP and its Robust Counterpart are equivalent for these uncertainty parameters. The objective of the RC-MCLP (recall Model 3.2.2) can be rewritten to:

$$\Delta u - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j + \eta_j \right) v_j^+ + \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right) v_j^- = \Delta u - (1+\epsilon) \sum_{j \in \mathcal{J}} \hat{\delta}_j v_j^+ + (1-\epsilon) \sum_{j \in \mathcal{J}} \hat{\delta}_j v_j^-.$$

As the optimal u is binary, we consider two cases:

• Suppose u = 0, then $v_j^+ = 0$ and $v_j^- = z_j$ for all $j \in \mathcal{J}$ (as derived before). The resulting objective is

$$(1-\epsilon)\sum_{j\in\mathcal{J}}\hat{\delta}_j z_j,$$

which has the same optimal solution as the MCLP (with $d_j = \hat{\delta}_j$).

• Suppose u = 1, then $v_j^+ = (1 - z_j)$ and $v_j^- = 0$ for all $j \in \mathcal{J}$. The objective simplifies to:

$$\Delta - (1+\epsilon) \sum_{j \in \mathcal{J}} \hat{\delta}_j (1-z_j) = \Delta - (1+\epsilon) \sum_{j \in \mathcal{J}} \hat{\delta}_j + (1+\epsilon) \sum_{j \in \mathcal{J}} \hat{\delta}_j z_j$$

Again, the optimal solution is the same as that of the MCLP (with $d_j = \hat{\delta}_j$).

Hence, the MCLP is robust for this common uncertainty structure and the Robust Counterpart has no added value. There are uncertainty structures where the Robust Counterpart leads to different solutions, as shown by the next example.

3.2.4.2 An Example Where Robustness Matters

We give an example of an MCLP instance where the robust solution is different than the nonrobust solution. This instance consists of three ambulance bases $\mathcal{I} = \{1, 2, 3\}$ of which only one can be opened. Base coverage is disjoint and each base covers exactly one demand point. That is, we have $\mathcal{J} = \{1, 2, 3\}$, where each point corresponds to a base. The estimated demand of the first two points are equal, $\hat{\delta}_1 = \hat{\delta}_2$, and so are the uncertainty parameters $(0 < \eta_1 = \eta_2 \leq \hat{\delta}_1)$. The third base has the largest estimated demand $\hat{\delta}_3 > \hat{\delta}_1$, but twice the uncertainty of the other bases, $\eta_3 = 2\eta_1$. Furthermore, we require that $0 \leq \hat{\delta}_3 - \eta_3 = \hat{\delta}_3 - 2\eta_1 < \hat{\delta}_1 - \eta_1$. The total demand is fixed to the total estimated demand: $\Delta = \hat{\delta}_1 + \hat{\delta}_2 + \hat{\delta}_3$.

Opening base 1 results in a non-robust coverage of $\hat{\delta}_1$ and a robust coverage of $\hat{\delta}_1 - \eta_1$. The same holds for opening base 2. For base 3 the non-robust coverage is $\hat{\delta}_3$ and the robust coverage is $\hat{\delta}_3 - \eta_3$. By construction it is optimal for the MCLP to open base 3, but for the RC-MCLP opening base 1 (or 2) is optimal. Therefore, the MCLP is not robust for all uncertainty structures. For explicit example, set $\hat{\delta}_1 = \hat{\delta}_2 = 3$, $\hat{\delta}_3 = 4$, and $\eta_1 = \eta_2 = \frac{1}{2}\eta_3 = 2$.

Note that such an instance with two bases is not possible: as $\delta_1 + \delta_2 = \Delta = \hat{\delta}_1 + \hat{\delta}_2$ we can assume without loss of generality that $\eta_1 = \eta_2$. Hence, if we assume that $\hat{\delta}_2 > \hat{\delta}_1$ it automatically implies that $\hat{\delta}_2 - \eta_2 > \hat{\delta}_1 - \eta_1$. Opening base 2 would be optimal. Consequently, the above described instance with three bases is the smallest possible instance where robustness matters.

3.2.5 Other Robust Counterparts

The derived Robust Counterpart in Proposition 3.2.1 can also be applied to other constraints. For instance, a common constraint on the minimum coverage is to require that the fraction of covered demand is at least $\alpha \in [0, 1]$:

$$\sum_{j \in \mathcal{J}} d_j z_j \ge \alpha \sum_{j \in \mathcal{J}} d_j.$$

The robust version of this constraint is

$$\sum_{j \in \mathcal{J}} \delta_j z_j \ge \alpha \Delta \qquad \qquad \forall \, \delta \in \mathcal{D}$$

This is equivalent to Equation (3.2.7) with $\theta = \alpha \Delta$ or Proposition 3.2.1 with $\phi_j = z_j$ and $\theta = \alpha \Delta$. Thus, the Robust Counterpart of the constraint is

$$\Delta u - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j + \eta_j \right) v_j^+ + \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right) v_j^- \ge \alpha \Delta,$$
$$u - v_j^+ + v_j^- = z_j \qquad \forall j \in \mathcal{J},$$
$$u \in \mathbb{R}_{\ge 0},$$
$$v_j^+, v_j^- \in \mathbb{R}_{> 0} \qquad \forall j \in \mathcal{J}.$$

As mentioned at the derivation of the Robust Counterpart MCLP, the optimal variables u, v_j^+ and v_j^- are binary. Alternatively, we can use the robust model derived through intuitive reasoning:

$$\sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right) z_j + u \ge \alpha \Delta,$$

$$u + \Delta v \le \Delta,$$

$$u - Mv \le \Delta - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j + \eta_j \right) (1 - z_j) - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right) z_j,$$

$$u \in \mathbb{R}_{\ge 0},$$

$$v \in \mathbb{B}.$$

Another example is the Robust Counterpart for the Maximum Expected Covering Location problem (see Model 2.3.1). We can apply Proposition 3.2.1 by taking $\phi_j = \sum_{k=1}^{q} (1-\rho)\rho^{k-1}z_j^k$ for $j \in \mathcal{J}$. Naturally, the Robust Counterpart is very similar to the RC-MCLP model. The new objective is to maximise

$$\Delta u - \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j + \eta_j \right) v_j^+ + \sum_{j \in \mathcal{J}} \left(\hat{\delta}_j - \eta_j \right) v_j^-.$$

Furthermore, the following constraints are added:

$$\begin{aligned} u - v_j^+ + v_j^- &= \sum_{k=1}^q (1 - \rho) \rho^{k-1} z_j^k & \forall j \in \mathcal{J}, \\ z_j^{k+1} &\leq z_j^k & \forall j \in \mathcal{J}, k \in \{1, \dots, q-1\}, \\ u \in \mathbb{R}, & \\ v_j^+, v_j^- \in \mathbb{R}_{>0} & \forall j \in \mathcal{J}. \end{aligned}$$

Of course, the optional extra constraints stated in Proposition 3.2.1 can be included. Note that we have added the constraints $z_j^{k+1} \leq z_j^k$, because it is not obvious that this follows automatically from the formulation.

3.3 Post-Optimisation Neighbourhood Analysis

Suppose we have obtained a final solution of our optimisation problem using any method. Note that it is not required that the solution is optimal. It is useful to analyse whether this solution is unique, i.e., are there (similar) solutions of practically the same quality? For instance, would changing one base location have a large impact on the objective? Likewise, there could be completely different solutions with (almost) the same objective value. We have to analyse the sensitivity of the objective to solution changes.

If valid alternatives exist, it is useful to present these to the decision maker. EMS models cannot capture the actual decision process (the real objective and feasible region), as they always simplify reality. By presenting alternatives, the decision maker can counteract some limitations of the model. However, it is not trivial how to do this. Obviously, a full search of the feasible region is intractable for the problems considered.

Depending on the solution method, a selection of valid alternatives can be readily available. For instance, randomised algorithms most likely give a different final solution each time and are usually performed multiple times. Local Search methods can be executed with different starting points. Furthermore, the best intermediate solutions can be stored. Branch-and-Bound procedures can store a list of the best feasible solutions found among the different branches (although the pruning of branches would need some modifications). The decision maker should decide which approach is appropriate. In particular, how many alternatives should be given and should these be similar or dissimilar to the final solution?

We will focus on a method that does not depend on the used solution method and provides alternatives similar to the final solution. Given a final solution, the method evaluates the neighbourhood of this solution. Which neighbourhood is used must be provided. Regarding the MCLP, a natural neighbourhood is the set of solutions that differ at $\rho \in \mathbb{N}$ or less places. That is, each solution in the neighbourhood can be obtained from the final solution by closing up to ρ bases and opening the same number of different bases. Opening less bases is not considered, as this does not improve coverage. Such a neighbourhood is called the ρ -Swap neighbourhood (see also the Swap Local Search method in Section 4.4).

The evaluation of the neighbourhood is to simply determine the objective values of the corresponding solutions (and possibly store a list of the best solutions). These objective values are then normalised by the objective value of the final solution and sorted in non-increasing order. These ordered values can be visualised, which we will call the ordered performance curve (OPC). Figure 3.3.1 shows five types of OPCs, where we have rescaled the performance such that the worst solution in the neighbourhood has performance zero. For clarity, the figure assumes that the final solution is the best solution in the neighbourhood.

The five types of OPCs are:

- Uniform performance (Figure 3.3.1a). The neighbourhood has an equal distribution of high, average and low performance. This implies that there are a few valid alternatives in the neighbourhood.
- High performance (Figure 3.3.1b). The neighbourhood has many solutions with high performance (many valid alternatives).

- Low performance (Figure 3.3.1c). The neighbourhood has many solutions with low performance (very few valid alternatives).
- Extreme performance (Figure 3.3.1d). The neighbourhood has many solutions with high or low performance, but few with average performance (a decent number of valid alternatives).
- Average performance (Figure 3.3.1e). The neighbourhood has many solutions with average performance, but few with high or low performance (very few valid alternatives).

Note that the above interpretation of the performance regarding valid alternatives is relative. A neighbouring solution with low performance can still be a valid alternative, as the performance is an ordered measure (its objective value can still be acceptable). Nevertheless, the concept and the different OPC types remain applicable.

Constructing the neighbourhood OPC for a given final solution serves two purposes: valid alternatives are found and a general indication about the uniqueness of the solution is obtained. Of course, this indication depends on the choice of the neighbourhood and should be interpreted in this context.

3.3.1 Intractable Neighbourhoods

For $\rho \in \mathbb{N}$ small enough a complete evaluation of the ρ -Swap neighbourhood is tractable for the MCLP. Otherwise, additional restrictions on the swaps have to be imposed. One approach is to partition the bases into subsets and to define the neighbourhood as the combination of the smaller Swap neighbourhood of each subset. This is particularly useful if these neighbourhood subsets follow directly from the real world, e.g., cities or provinces.

Another approach is to take (uniformly) random samples from the Swap neighbourhood and approximate the OPC. This approach is similar to Ordinal Optimisation techniques, see for instance L. Lee et al. (1999) and Shen et al. (2010). Ordinal Optimisation focusses on qualitative measures instead of quantitative objectives: the order of solutions plays a central part (i.e., one solution is better than the other). By uniformly sampling from the entire feasible region, probability statements on the order of a heuristic solution can be made. For instance, a sampling approach can be constructed to substantiate statements as: 'the given heuristic solution belongs to the best 5% of all solutions with probability 0.99'.

However, it is often the case with NP-hard problems¹ that there is an abundance of bad solutions and sensible heuristics construct reasonably good solutions. This leads to insignificant statements on the order of the heuristic solution (i.e., it will belong the best 1% solutions, but still relatively far away from the optimal solution). The described post-optimisation neighbourhood analysis tries to avoid these phenomena by only considering the neighbourhood of the final solution. This eliminates a large proportion of the bad solutions from the sampling, resulting in a more meaningful OPC. A disadvantage of this approach is that we cannot derive statements on the overall order of the final solution, as only the neighbourhood is considered. We will apply this method in Section 3.4.

¹See also Appendix A for the definition of NP-hard problems.



Figure 3.3.1: Types of ordered performance curves (OPCs).

3.4 Numerical Results

We perform the described sensitivity analysis to a set of MCLP instances, which are based on realistic data of the 24 Regional Ambulance Services of The Netherlands. For the corresponding 24 MCLP instances the set of possible base locations is equal to the set of demand points ($\mathcal{I} = \mathcal{J}$). The number of demand points ranges from 40 to 474, but most instances have around 100 to 200 points. All demand points have strictly positive demand weights. On average 9 bases are allowed to be opened, where the number of bases to open ranges from 3 to 19. A detailed description is given below in Section 3.4.1.

Sections 3.4.2, 3.4.3, and 3.4.5 discuss the sensitivity analysis approaches, starting with the sensitivity to the number of available bases. In Section 3.4.4 we apply the post-optimisation neighbourhood analysis. Common to all analyses are the properties of the MCLP, in particular the 'all-or-nothing' coverage. Since the objective does not distinguish between covered points, the way a point is covered has no part in the optimisation. For example, covering a demand point in 5 minutes or 10 minutes is regarded as the same by the MCLP. Furthermore, the availability of ambulances is not included. Therefore, the MCLP provides optimistic results. The results should be seen in the context of the limitations of the model.

In Section 3.5 we go into the details of two RAV regions, namely RAV14 (Gooi- en Vechtstreek) and RAV23 (Limburg-Noord). For example, we will compare which bases are opened. Here, we restrict the analysis to attainable coverage and do not discuss the actual solutions.

3.4.1 Realistic Instances

The Netherlands is divided into 24 EMS regions, which are called 'Regionale Ambulancevoorzieningen' (RAV). These are shown in Figure 3.4.1. We note that there used to be 25 regions: Amsterdam-Amstelland (Region 11) and Zaanstreek-Waterland (Region 13) have been merged. Each RAV has a single coordination centre (call centre and dispatcher) and one or multiple ambulance services. In practice, the RAVs collaborate and sometimes provide service to adjacent regions. However, for our test purposes, we assume that all regions operate independently. For an analysis where The Netherlands is regarded as one EMS region, see Looije (2013).

We discretise and aggregate The Netherlands into postal code areas, using only four digits (instead of six characters). For the Maximal Covering Location problem we assume that an ambulance base can be positioned at each of these postal code areas. Based on Kommer and Zwakhals (2011), we have the following data available:

- the number of inhabitants of each postal code area,
- the number of available ambulance bases for each RAV,
- the postal code areas assigned to each RAV,
- average travel times of ambulances with sirens between postal code areas.

Kommer and Zwakhals (2011) have built a driving time model that incorporates aspects such as the time of day and road type. Parameters are estimated from historical data. It provides average travel times of ambulances with sirens between the centres of 4-digit postal code areas. We will use the estimated travel times for rush hours, as these provide the most conservative performance measures.



Figure 3.4.1: The 24 Regional Ambulance Services of The Netherlands. (Modified from source: www.zorgatlas.nl.)

In The Netherlands, an ambulance is considered to reach an emergency call in time if the response time is within 15 minutes. The response time includes the time to handle the call, to dispatch the ambulance, and the travel time. The time up to and including the dispatching requires about 3 minutes. Therefore, we use a threshold of 12 minutes for the travel times to determine whether demand points are covered. As a result, we obtain our adjacency parameters for the EMS models.

The list of provided available ambulance bases contains different types of bases. For instance, some are not manned 24 hours a day, and others serve only as starting points for shifts. Unfortunately, no common definition is used between the RAVs. Therefore, we do not distinguish the bases and include all to determine the number of bases to open. For actual decision support, we recommend to first consult the RAV under consideration to adequately set the parameters.

Historical data to estimate the arrival rates of emergency calls is unavailable. Instead, we will use the number of inhabitants as an indication of the arrival rates. In fact, we assume that the arrival rate of calls is proportional to the number of inhabitants in that area, using the same proportion for all postal code areas. As a result, we can directly use the number of inhabitants as demand weights, implicitly rescaling the weights by a constant factor.

One could argue that historical data should be preferred. For instance, industrial areas have few inhabitants, but severe accidents are more likely to occur than in a residential area. Similar arguments hold for motorways. However, the usefulness of historical data all depends on its reliability. If the frequency of calls is low or if the time span of the historical data is short, care has to be taken to prevent extreme biases: too much focus can be placed on what has happened in the past (by chance) and other scenarios will be neglected. Estimating arrival distributions can be troublesome if the frequency is low. Ideally, a combination of historical data, the number of inhabitants, and expert judgement is used to estimate the weights for the demand points.

In the provided data the postal code areas without any inhabitants have been filtered out (e.g., parks, small forests, new residential areas under construction). As all of our models use weighted constraints or objectives, this should not impact the coverage of a set of bases. However, it does limit the possible base locations. Therefore, we have to assume that postal code areas without any inhabitants are unsuitable for an ambulance base. Further research can determine whether this assumption affects the coverage for the RAVs. The resulting number of 4-digit postal code areas per RAV are shown in Table 3.4.1, as are the number of available bases and the total number of inhabitants.

RAV Name	RAV Region	Number of Opened Bases	Number of Postal Codes	Number of Inhabitants
Groningen	1	13	250	576615
Friesland	2	19	474	646045
Drenthe	3	13	255	489910
IJsselland	4	10	170	506845
Twente	5	9	120	623050
Noord- en Oost-Gelderland	6	13	201	809865
Gelderland-Midden	7	7	134	655725
Gelderland-Zuid	8	11	158	526835
Utrecht	9	11	217	1220125
Noord-Holland-Noord	10	9	167	641805
Amsterdam/Waterland	11	9	161	1261997
Kennemerland	12	7	98	519757
Gooi- en Vechtstreek	14	3	40	243540
Haaglanden	15	8	141	1016400
Hollands-Midden	16	10	124	760930
Rotterdam-Rijnmond	17	10	185	1247858
Zuid-Holland-Zuid	18	6	98	479435
Zeeland	19	11	153	381395
Midden-en West-Brabant	20	13	217	1070885
Brabant-Noord	21	7	146	636870
Brabant-Zuidoost	22	7	137	734841
Limburg-Noord	23	7	137	513855
Limburg-Zuid	24	4	141	607540
Flevoland	25	6	91	386184
Total		223	4015	16558307

Table 3.4.1: Specifications of all Regional Ambulance Services (RAV) regions. Note that the postal codes are aggregated into 4-digit areas.

3.4.2 Number of Bases

The RAV data prescribes the number of available bases $p \in \mathbb{N}$ for each RAV region. We analyse the sensitivity to the number of available bases by iterating over all possible values (i.e., from 1 to $|\mathcal{I}|$). Of course, if full coverage is attainable for some number of bases, this is also the case when allowing more bases. Therefore, the iterative procedure can be terminated prematurely.

A graphical representation of the results is shown in Figures 3.4.2 and 3.4.3. For each RAV region the maximum attainable coverage is shown for a range of number of bases to open. The coverage is normalised by the total demand of the region (so full coverage corresponds to 1). When full coverage is attained the results for additional bases are not shown, as these also have full coverage. See Table 3.4.2 for the minimum number of bases required for complete coverage. Note that the RAVs are numbered 1 to 25, where 13 is skipped (as mentioned in Section 3.4.1).

As expected, all RAVs show a decrease in the gain of adding an additional base (which is called submodularity, as we will show in Chapter 4). Smaller regions (RAV12, RAV14, RAV15, and RAV24) need only a few bases to attain full coverage, whereas large regions (RAV2, RAV3, and RAV6) need many. In particular, RAV2 (Friesland) needs 22 bases for complete coverage, although a coverage of 99% is possible with only 16 bases. One could argue whether such an investment in additional bases is worth it. Note that for almost all RAV regions it is expensive to improve a coverage of 99%.

In Table 3.4.3 the detailed results are given, rounded to four decimals. Therefore, a coverage of 1.0000 implies almost complete coverage (e.g., see RAV12 for 3 bases). Full coverage is not rounded. Once complete coverage is attained no further results are given, similar to Figures 3.4.2 and 3.4.3. Hence, an empty entry implies that complete coverage is possible.

As we are given the actual number of available bases for each RAV, we have highlighted the corresponding entry in green. For 12 RAV regions there is a surplus of bases according to the MCLP, i.e., full coverage can be maintained with fewer bases. The required number of bases of 9 RAVs matches exactly the available number of bases. Finally, 3 RAV regions have incomplete coverage: RAV2 (Friesland), RAV7 (Gelderland-Midden), and RAV23 (Limburg-Noord). However, these regions still have a coverage of more than 99%. The surplus or shortage of bases for full coverage is summarised in Table 3.4.2.

These positive results are not surprising: historical data shows that the real coverage is about 92% (see Boers et al. (2010)) and the MCLP is optimistic in the resulting coverage. In practice, we cannot conclude that most RAVs require fewer bases, because of the model limitations of the MCLP. However, we can conclude that most likely additional bases are required for RAV2, RAV7, and RAV23, if complete coverage is the objective.

RAV Region	Number of Opened Bases	Number of Bases for Full Coverage	Surplus of Bases
1	13	12	+1
2	19	22	-3
3	13	12	+1
4	10	10	0
5	9	6	+3
6	13	13	0
7	7	8	-1
8	11	8	+3
9	11	10	+1
10	9	8	+1
11	9	6	+3
12	7	4	+3
14	3	2	+1
15	8	4	+4
16	10	8	+2
17	10	10	0
18	6	6	0
19	11	11	0
20	13	11	+2
21	7	7	0
22	7	7	0
23	7	8	-1
24	4	4	0
25	6	6	0
Total	223	203	+20

Table 3.4.2: Minimum number of bases to attain full coverage. The surplus of bases is highlighted in green if it is positive and in red if negative.



Figure 3.4.2: Maximum coverage attainable with the shown number of bases (1 of 2).



Figure 3.4.3: Maximum coverage attainable with the shown number of bases (2 of 2).
RAV	Maximum Coverage for Number of Opened Bases												
Region	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.4462	0.6077	0.7077	0.7962	0.8566	0.9100	0.9586	0.9872	0.9949	0.9974	0.9995	1	
2	0.2433	0.4006	0.5506	0.6627	0.7614	0.8119	0.8519	0.8811	0.9076	0.9302	0.9528	0.9654	0.9747
3	0.2418	0.4761	0.6454	0.7473	0.8211	0.8891	0.9352	0.9623	0.9852	0.9960	0.9995	1	
4	0.3526	0.6302	0.7685	0.8521	0.9288	0.9718	0.9891	0.9959	0.9994	1			
5	0.4655	0.7803	0.9106	0.9686	0.9998	1							
6	0.2521	0.4317	0.5622	0.6907	0.8064	0.8928	0.9314	0.9605	0.9833	0.9956	0.9988	0.9997	1
7	0.4662	0.7307	0.8548	0.9307	0.9702	0.9909	0.9998	1					
8	0.4969	0.7320	0.8691	0.9487	0.9863	0.9981	0.9997	1					
9	0.4455	0.7027	0.7931	0.8611	0.9208	0.9600	0.9919	0.9958	0.9987	1			
10	0.3969	0.6763	0.8315	0.9238	0.9528	0.9774	0.9989	1					
11	0.6762	0.8893	0.9564	0.9981	0.9999	1							
12	0.6319	0.9544	1.0000	1									
14	0.9640	1											
15	0.8269	0.9424	0.9925	1									
16	0.4424	0.7068	0.8515	0.9345	0.9813	0.9931	0.9999	1					
17	0.6180	0.7930	0.8808	0.9209	0.9492	0.9770	0.9892	0.9969	1.0000	1			
18	0.6075	0.7993	0.9655	0.9846	0.9958	1							
19	0.3045	0.5210	0.6968	0.7757	0.8423	0.9033	0.9445	0.9687	0.9844	0.9979	1		
20	0.3170	0.5691	0.7443	0.8516	0.8979	0.9336	0.9609	0.9806	0.9938	0.9994	1		
21	0.4594	0.7138	0.8280	0.9414	0.9815	0.9996	1						
22	0.5535	0.7878	0.8772	0.9492	0.9753	0.9884	1						
23	0.3292	0.5856	0.7255	0.8391	0.9076	0.9635	0.9958	1					
24	0.5140	0.7952	0.9792	1									
25	0.4854	0.7067	0.8697	0.9439	0.9962	1							
RAV		Max	imum C	overage	for Nun	nber of	Opened 1	Bases		-			
Region	14	15	16	17	18	19	20	21	22	-			
2	0.9821	0.9875	0.9916	0.9952	0.9970	0.9985	0.9991	0.9999	1				

Table 3.4.3: Maximum coverage attainable with the shown number of bases. The entries highlighted in green correspond to the actual number of available bases of the RAV. Duplicate entries of full coverage are not shown, i.e., an empty entry implies full coverage.

73

3.4.3 Adjacency Parameters

As described in Section 3.4.1, the adjacency parameters $a_{ij} \in \mathbb{B}$ for the RAV instances are determined by the travel time between bases and demand points. A point can be covered by a base if the corresponding travel time is less than or equal to a certain time threshold (12 minutes in practice). We will analyse the effect of the travel time threshold on the resulting maximum coverage as follows. We iteratively increase the threshold by 10 seconds, determine the adjacency parameters, and solve the corresponding MCLP. We obtain the maximum coverage for the travel time thresholds and the minimum threshold to attain full coverage (accurate up to 10 seconds).

We also consider the robustness of a fixed solution with respect to the travel time threshold. First, we solve the MCLP for the realistic threshold of 12 minutes. The resulting set of opened bases is the fixed MCLP solution. Next, we iterate over the travel time thresholds as before, but keep the solution fixed to the original. That is, given the fixed MCLP solution based on a threshold of 720 seconds, what is the coverage if we modify the travel time threshold?

The results are shown in Figures 3.4.4 and 3.4.5, where the maximum coverage is given in red and the fixed MCLP solution in blue. For now, please ignore the green and magenta lines. The coverage is normalised by the total demand in the RAV region. The thresholds ranges up to and including the time for which the fixed solution attains complete coverage for the first time.

For all RAVs the maximum coverage hardly increases for thresholds ranging from 0 to 100 seconds, with on average a coverage of 20%. Note that the maximum coverage is never equal to zero, as the travel time from a base to its own postal code area is zero. From a threshold of around 100 seconds the maximum coverage increases almost linearly to approximately 80%. From thereon, the gain in coverage slowly decreases as the threshold increases, until full coverage is attained. The fixed MCLP solution has a lower starting coverage (around 0.5%) and for most RAVs increases at the same rate as the maximum coverage.

For four thresholds the maximum coverage and the coverage attained by the fixed solution are shown in Table 3.4.5. The travel time thresholds are 6, 8, 10, and 12 minutes. For a threshold of 12 minutes both approaches are the same and thus no distinction is made. Please ignore the 2-Stage columns for now.

It is clear that there is a significant difference between the maximum coverage and the coverage obtained by the fixed solution for the various thresholds. As the MCLP uses 'all-or-nothing' coverage, we should not expect it to be robust to changes in the travel time threshold. Indeed, the results show that the solutions are very sensitive to the threshold (in particular for RAV4, RAV12, and RAV15).

Table 3.4.4 gives the minimum travel time thresholds to attain complete coverage (again, please ignore the 2-Stage column). Note that RAV2, RAV7, and RAV23 require a threshold of more than 720 seconds as these do not have full coverage when using the actual travel time thresholds. Several RAV regions can be completely covered with relatively low thresholds, namely RAV5, RAV8, RAV11, RAV12, RAV14, and RAV15. These regions appear to have insensitive solutions, but are not found (or returned) by the solver². In practice, the insensitive solutions would be preferred.

 $^{^{2}}$ We use Gurobi as general Mixed Integer Programming solver, see Appendix B for more details.

We conclude that primarily regions with a surplus of bases (overcapacity) have solutions that are relatively insensitive to modifications to the travel time threshold. This does not hold for RAV regions with a shortage of bases. Remarkably, RAV25 has no surplus or shortage of available bases, but is also relatively insensitive to the threshold. Full coverage can be attained with a threshold of 10 minutes. The results do lead to concerns: the MCLP cannot distinguish which optimal solutions are insensitive. To counteract this limitation, we present a two-stage optimisation approach.

3.4.3.1 Two-Stage Optimisation

Multi-stage optimisation is to sequentially solve multiple optimisation problems, where each optimisation problem depends on the previous ones. In our case, we will use two optimisation stages: the maximum coverage is determined first with the MCLP, followed by an optimisation model that minimises the required travel time threshold to attain a certain coverage. The required coverage is set to the maximum coverage, determined in the first stage. Therefore, the final solution of the two-stage optimisation maximises coverage first and then minimises the required travel time threshold. These solutions are exactly the optimal solutions that are relatively insensitive to modifications to the threshold.

Let $\tau_{ij} \in \mathbb{R}_{\geq 0}$ be the travel time from base $i \in \mathcal{I}$ to point $j \in \mathcal{J}$. Model 3.4.1 shows the optimisation model of the second stage. In the model, demand points are assigned to a base, where it is optimal to assign it to the nearest base. The decision variables u_{ij} captures the assignment: u_{ij} is equal to one if point $j \in \mathcal{J}$ is assigned to base $i \in \mathcal{I}$ and zero otherwise. Each point is assigned to exactly one base. Consequently, the expression

$$\sum_{i\in\mathcal{I}}a_{ij}u_{ij}$$

indicates whether a point is covered and is equal to z_j of the MCLP model. Therefore, a minimum fraction $\alpha \in [0, 1]$ of covered demand can be achieved by requiring that

$$\sum_{j \in \mathcal{J}} d_j \sum_{i \in \mathcal{I}} a_{ij} u_{ij} \ge \alpha \sum_{j \in \mathcal{J}} d_j.$$

Finally, the objective is to minimise the travel time threshold, captured by the decision variable $v \in \mathbb{R}_{\geq 0}$. This is equivalent to minimising the maximum travel time between bases and their assigned points. Here, we can choose to include or exclude uncovered points. As the MCLP disregards uncovered points, we have chosen to exclude uncovered points. The corresponding constraint is:

$$\sum_{i \in \mathcal{I}} a_{ij} \tau_{ij} u_{ij} \le v \qquad \forall j \in \mathcal{J}.$$

To include all points we should remove the adjacency parameters a_{ij} from the expression. Note that the shown constraint includes demand points with zero weight $(d_j = 0)$. By multiplying both sides of the inequality with d_j , these points can be ignored. As all points of the RAV regions have strictly positive demand, this issue does not apply in our case.

Notice that the multi-stage optimisation approach can be extended by or modified to other secondary criteria, always selecting a solution from the set of optimal MCLP solutions. For example, the average weighted travel time can be minimised.

Model 3.4.1: Minimum Travel Time Threshold model.

There are two main arguments for using the described two-stage approach. When there is a surplus of available bases, the MCLP generally has solutions that are insensitive to changes to the travel time threshold, but it cannot make the distinction. In practice, these insensitive solutions are preferred. Furthermore, in The Netherlands 12 of the 24 RAV regions have a surplus of bases according to the MCLP. Therefore, this issue also arises in practice. The two-stage approach allows us to select an insensitive solution from the set of optimal MCLP solutions.

We have applied the two-stage approach to the RAV regions. As mentioned before, the required covered fraction of demand α is equal to the coverage of the fixed MCLP solution. While the travel time threshold changes, we keep the two-stage solution fixed. The results are given in the 2-Stage columns of Tables 3.4.4 and 3.4.5, and shown in green in Figures 3.4.4 and 3.4.5.

For most RAV regions the two-stage solution improves the coverage with respect to the fixed MCLP solution, but there are exceptions (see RAV21). We are only guaranteed a non-strict improvement when the original coverage (fraction α) is attained. Table 3.4.5 gives the coverage of the two-stage solution for the four selected thresholds. By construction, for a threshold of 12 minutes the coverage is the same as the fixed MCLP coverage. Therefore, no distinction is shown. If the two-stage coverage improves the coverage of the fixed MCLP solution, the corresponding entry is highlighted in green.

In Table 3.4.4 the 2-Stage entries are highlighted in green if the minimum threshold for full coverage is strictly lower than that of the fixed MCLP solution. Recall that we have chosen to exclude uncovered points. This choice only affects RAV regions where the fixed solution has incomplete coverage (RAV2, RAV7, and RAV23). It turns out that including all points improves the minimum threshold of the two-stage solution for RAV2 to 970 seconds. The thresholds for RAV7 and RAV23 are unchanged.

The regions RAV5, RAV8, RAV11, RAV12, RAV14, and RAV15 motivated the use of a two-stage optimisation. We see that indeed the two-stage solution leads to great improvement for these RAVs. Similar improvement is also achieved for other regions, see RAV9, RAV16, RAV20, and RAV25. Note that for RAV14 the two-stage solution achieves a coverage exceptionally close to the maximum coverage for thresholds greater than 6 minutes.

We conclude that the two-stage approach is a valid strategy for determining solutions that are less sensitive to changes in the travel time threshold. The greatest improvement is achieved if full coverage can be attained with lower thresholds than would be used normally. However, notice that the fixed two-stage solution of RAV4 is superior to the fixed MCLP solution, even though the minimum travel time threshold of 12 minutes cannot be significantly improved. This implies that the two-stage approach should always be considered if full coverage can be attained. For RAV regions with partial (incomplete) coverage the method shows no added benefit.

3.4.3.2 Current Locations

The two-stage optimisation approach is a method to model multi-criteria objective functions. In reality, the locations of ambulance bases are determined using multiple criteria. In particular, bases are usually placed closer to large cities, as these are prioritised over rural areas. We can evaluate if the current base locations are less sensitive to changes in the travel time threshold compared to the fixed MCLP solution.

Before we discuss the results, we have to note that the current base locations are translated to 4-digit postal code areas. Consequently, the actual travel times can differ from those used by the model. This holds in particular for large postal code areas. Therefore, we do not state the attained coverage in tables, as these can easily be misinterpreted.

Figures 3.4.4 and 3.4.5 show the (MCLP) coverage attained by the current locations in magenta. In general, for thresholds around 720 seconds the current coverage is less than the maximum attainable coverage. However, for threshold below 600 seconds the current coverage outperforms that of the fixed MCLP solution. Moreover, for lower thresholds it outperforms the coverage of the fixed two-stage solution, see for instance RAV5, RAV7, RAV8, RAV11, and RAV18. This is not always the case, as can be seen for RAV25.

Notice that the current coverage for thresholds below 100 seconds is generally higher than the coverage of the other fixed solutions. This suggests that current bases are indeed located closer to postal code areas with many inhabitants, resulting in less sensitive coverage.

RAV	Travel Time Threshold for Full Coverage									
Region	Max	Fixed	2-Stage							
1	670	720	670							
2	770	1030	1030							
3	680	720	680							
4	720	720	720							
5	570	710	570							
6	690	720	690							
7	730	990	950							
8	540	720	540							
9	650	720	650							
10	630	720	630							
11	580	720	580							
12	520	690	520							
14	510	710	510							
15	460	720	460							
16	600	710	600							
17	690	720	690							
18	720	720	720							
19	710	710	710							
20	690	720	690							
21	690	720	690							
22	710	720	710							
23	750	820	820							
24	700	720	700							
25	600	720	600							

Table 3.4.4: Minimum travel time threshold (in seconds) to attain full coverage. Shown are the maximum attainable coverage solution, the fixed MCLP solution based on a threshold of 720 seconds, and the fixed two-stage solution based on the fixed MCLP solution. If the fixed two-stage solution improves the fixed MCLP solution, the corresponding entry is highlighted in green. The results are determined using increments of 10 seconds.



Figure 3.4.4: Maximum coverage attainable with the shown travel time threshold (1 of 2). The results are determined using increments of 10 seconds. In red the maximum attainable coverage is shown, in blue the coverage of the fixed MCLP solution based on a threshold of 720 seconds, in green the coverage of the fixed two-stage solution based on the fixed MCLP solution, and in magenta the approximated coverage of the current base locations.



Figure 3.4.5: Maximum coverage attainable with the shown travel time threshold (2 of 2). The results are determined using increments of 10 seconds. In red the maximum attainable coverage is shown, in blue the coverage of the fixed MCLP solution based on a threshold of 720 seconds, in green the coverage of the fixed two-stage solution based on the fixed MCLP solution, and in magenta the approximated coverage of the current base locations.

	Coverage for Travel Time Threshold											
RAV	360 (s)				480 (s))		720 (s)				
Region	Max	Fixed	2-Stage	Max	Fixed	2-Stage	Max	Fixed	2-Stage			
1	0.7636	0.4874	0.5060	0.9218	0.7451	0.7934	0.9946	0.9208	0.9465	1		
2	0.7250	0.2991	0.3103	0.8642	0.5266	0.5357	0.9686	0.8010	0.8346	0.9985		
3	0.6886	0.4218	0.3417	0.8805	0.6832	0.6329	0.9883	0.8724	0.8602	1		
4	0.8047	0.3861	0.3962	0.9130	0.6057	0.7117	0.9924	0.7701	0.9004	1		
5	0.8075	0.3642	0.4944	0.9630	0.7026	0.8421	1	0.9032	1	1		
6	0.6509	0.4038	0.3933	0.8418	0.6414	0.5874	0.9706	0.8839	0.9072	1		
7	0.7320	0.3279	0.2650	0.8746	0.5507	0.4871	0.9849	0.8119	0.8189	0.9998		
8	0.8279	0.2838	0.5233	0.9752	0.5090	0.9216	1	0.8369	1	1		
9	0.7201	0.2628	0.4002	0.9110	0.5299	0.7965	0.9981	0.8121	0.9735	1		
10	0.7665	0.2930	0.2811	0.9027	0.5177	0.5738	0.9950	0.7902	0.9920	1		
11	0.8468	0.3029	0.4895	0.9932	0.5024	0.8344	1	0.7612	1	1		
12	0.8951	0.2274	0.6037	0.9990	0.4959	0.8433	1	0.7248	1	1		
14	0.8249	0.3183	0.7802	0.9991	0.4785	0.9991	1	0.7193	1	1		
15	0.9339	0.2680	0.7674	1	0.4740	1	1	0.7175	1	1		
16	0.7879	0.2852	0.4805	0.9411	0.5983	0.8651	1	0.8452	1	1		
17	0.8172	0.3743	0.4140	0.9525	0.5763	0.7660	0.9882	0.9041	0.9142	1		
18	0.7177	0.3488	0.2977	0.8904	0.5631	0.6088	0.9876	0.8450	0.8584	1		
19	0.6978	0.4044	0.4114	0.9102	0.6544	0.6537	0.9767	0.8670	0.8251	1		
20	0.7398	0.3378	0.4620	0.9105	0.5597	0.6728	0.9881	0.7993	0.9103	1		
21	0.6260	0.3424	0.2518	0.8813	0.5520	0.4741	0.9856	0.8430	0.8310	1		
22	0.6462	0.2034	0.3483	0.8378	0.4956	0.6630	0.9628	0.8532	0.8941	1		
23	0.5801	0.3642	0.3570	0.7821	0.5585	0.5698	0.9319	0.8251	0.8251	0.9958		
24	0.5797	0.2539	0.2944	0.8018	0.5370	0.5738	0.9797	0.8749	0.8711	1		
25	0.8869	0.4408	0.7391	0.9551	0.7648	0.9052	1	0.9284	1	1		

Table 3.4.5: Maximum coverage attainable with the shown travel time threshold. Shown are the maximum attainable coverage solutions, the fixed MCLP solution based on a threshold of 720 seconds, and the fixed two-stage solution based on the fixed MCLP solution. If the fixed two-stage solution improves the fixed MCLP solution, the corresponding entry is highlighted in green.

3.4.4 Post-Optimisation Analysis

The results of the sensitivity analysis of the travel time threshold show that the RAV regions have multiple optimal solutions and secondary objectives can be used to select solutions with preferred properties. These observations further justify the use of post-optimisation analyses, e.g., the analysis described in Section 3.3. We will construct the ordered performance curve (OPC) of the neighbourhood of the optimal solution. Since the optimal solutions returned by the solver are somewhat arbitrary, we have chosen to use the two-stage MCLP solution (see Section 3.4.3.1). By using two-stage optimisation, we reduce the arbitrariness of the resulting optimal solution.

We also have to specify which neighbourhood is under consideration. As mentioned in Section 3.3, the Swap neighbourhood is a natural choice for the MCLP. We use two neighbourhoods: the set of all solutions that differ exactly one and exactly two bases with the optimal solution, respectively. Note that these neighbourhoods are disjoint and depend on the chosen optimal solution. In particular, we are interested in the number of (alternative) optimal solutions in the neighbourhood.

Alternative optimal solutions can lie outside these two neighbourhoods. To give an indication how many optimal MCLP solutions a region has, we iteratively determine a new optimal solution by cutting off the previous ones. Suppose we have an optimal set of opened bases $\mathcal{I}^* \subseteq \mathcal{I}$. This solution can be cut off by adding the following constraint to the MCLP:

$$\sum_{i\in\mathcal{I}^*} x_i \le |\mathcal{I}^*| - 1 = p - 1$$

When the original coverage cannot be attained by the new solution, we know that we have determined all optimal solutions. However, the number of alternative optimal solutions can be extremely large. Therefore, we limit the search to 100 alternative solutions. Note that this search can be seen as an unrestricted neighbourhood search where any number of bases can be replaced.

Table 3.4.6 shows the number of (alternative) optimal solutions in the discussed neighbourhoods. As we have mentioned, the search of the complete solution space (shown in the last column) has been limited to find up to 100 alternative optimal solutions. For most RAVs there are between 20 and 40 optimal alternatives where one base has been replaced. RAV3, RAV5, RAV8, RAV11, and RAV12 have even more: between 250 and 300. Increasing the neighbourhood to two base replacements generally leads to a significant number of additional optimal solutions found (easily up to 10 times as many alternatives).

There are exceptions, such as RAV2: there are at least 100 alternatives in the complete solution space, but in the two restricted neighbourhoods only 14 and 76 optimal solutions can be found. A similar statement holds for RAV7, but for RAV7 we have determined that there are 71 alternative optimal solutions in total. RAV23 is exceptional: only one alternative exists and lies in the 1-Swap neighbourhood. These three regions have in common that the realistic number of bases is not sufficient for complete coverage, i.e., there is no surplus in capacity. Completely opposite is RAV15, which has twice as many bases than needed for full coverage. This region has a huge number of alternatives: 720 and more than 100 000 in the two restricted neighbourhoods. We can conclude that excess capacity leads to more optimal solutions, which is expected.

The ordered performance curves of the two restricted neighbourhoods are shown in Figures 3.4.6 to 3.4.9. Also recall the five types of typical OPCs shown in Figure 3.3.1. The neighbourhood OPCs are not smooth and do not always have a clear match with one of the five types. However, we can conclude that there is no OPC of the low performance type. In Figures 3.4.6 and 3.4.7 (one base replaced) we see mostly high performance curves. RAV6, RAV10, and RAV23 have approximately a uniform (linear) performance. Furthermore, RAV12 and RAV18 match reasonably with the extreme performance curve.

For the neighbourhood with two bases replaced (Figures 3.4.8 and 3.4.9) mostly the high and average performance types match with the OPCs. For instance, RAV1, RAV14, and RAV15 have high performance, and RAV2, RAV7, and RAV23 average performance. In general, we see high performance OPCs for regions with surplus capacity and average performance for regions with no surplus or a small shortage. The observation that surplus capacity leads to many alternative optimal solutions and a high performance OPC is not surprising. However, it is reassuring that when there is no surplus, this does not lead to a low performance OPC.

To confirm that surplus capacity is the main contributor to having multiple optimal solutions, we perform a search of the complete solution space for a range of allowed number of bases to open. As before, we limit the search to finding at most 100 alternative optimal solutions. The results are shown in Table 3.4.7. There are almost no alternatives when only a few bases are available (incomplete coverage). When a coverage of 98% or more can be attained, there are generally more alternatives. In particular, opening exactly the minimum number of bases for full coverage (see also Table 3.4.2) results in at least 100 alternatives, with the exception of RAV14. To conclude, the existence of multiple optimal solutions occurs primarily when there is at least 98% coverage.

RAV	Number of Optimal Solutions in Neighbourhood								
Region	1 Base	2 Bases	Any						
1	73	2208	≥ 100						
2	14	76	≥ 100						
3	293	12149	≥ 100						
4	30	329	≥ 100						
5	277	13534	≥ 100						
6	12	80	≥ 100						
7	8	24	71						
8	259	21845	≥ 100						
9	49	1051	≥ 100						
10	40	669	≥ 100						
11	270	20014	≥ 100						
12	259	15918	≥ 100						
14	36	439	≥ 100						
15	720	104481	≥ 100						
16	61	1910	≥ 100						
17	38	623	≥ 100						
18	23	151	≥ 100						
19	32	425	≥ 100						
20	42	867	≥ 100						
21	20	165	≥ 100						
22	9	29	≥ 100						
23	1	0	1						
24	14	44	≥ 100						
25	31	359	≥ 100						

Table 3.4.6: Number of optimal solutions in the neighbourhood of the optimal MCLP solution of the two-stage minimum travel time threshold model. The shown neighbourhoods indicate how many open bases are replaced in the alternative optimal solution.



Figure 3.4.6: Ordered performance curves of the neighbourhood of the optimum of the two-stage minimum travel time threshold model (1 of 2). The neighbourhood consists of all solutions for which one open base has been replaced.



Figure 3.4.7: Ordered performance curves of the neighbourhood of the optimum of the two-stage minimum travel time threshold model (2 of 2). The neighbourhood consists of all solutions for which one open base has been replaced.



Figure 3.4.8: Ordered performance curves of the neighbourhood of the optimum of the two-stage minimum travel time threshold model (1 of 2). The neighbourhood consists of all solutions for which two open bases have been replaced.



Figure 3.4.9: Ordered performance curves of the neighbourhood of the optimum of the two-stage minimum travel time threshold model (2 of 2). The neighbourhood consists of all solutions for which two open bases have been replaced.

RAV			Alter	native (Optima	l Solut	ions for	· Numb	er of C	pened	Bases		
Region	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	0	0	0	0	0	1	3	1	49	≥ 100	
2	0	0	0	0	0	0	0	0	0	0	0	7	0
3	0	0	0	0	0	0	0	0	6	5	≥ 100	≥ 100	
4	0	0	0	1	1	1	1	62	59	≥ 100			
5	0	0	0	0	1	≥ 100							
6	0	0	0	1	1	1	1	0	1	1	7	≥ 100	≥ 100
7	0	0	0	0	3	19	71	≥ 100					
8	0	0	0	2	41	≥ 100	≥ 100	≥ 100					
9	0	0	0	0	0	0	0	≥ 100	≥ 100	≥ 100			
10	0	0	0	0	0	0	2	≥ 100					
11	0	0	0	0	33	≥ 100							
12	0	0	1	≥ 100									
14	0	15	≥ 100										
15	0	0	8	≥ 100									
16	0	0	0	1	1	1	85	≥ 100					
17	0	0	0	6	20	20	≥ 100	≥ 100	≥ 100	≥ 100			
18	0	0	0	17	17	≥ 100			> 100	> 400	> 100		
19	0	0	0	1	5	5	11	≥ 100	≥100	≥ 100	≥ 100		
20	0	0	0	0	0	0	1	0	1	11	≥ 100		
21	0	0	0	0	12	40	≥ 100						
22	1	1	0	0	2	33 1	≥100	> 100					
23	1	1	0	0	1	1	L	≥100					
24	0	0	0	≥ 100 ≥ 100	>100	>100							
20	0	0	0	≥100	≥100	≥ 100				_			
RAV	Altern	ative	Optima	l Solut	ions for	r Numł	oer of C	Opened	Bases	_			
Region	14	15	16	17	18	19	20	21	22	=			
2	3	7	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100	≥ 100				

Table 3.4.7: Number of alternative optimal solutions with the shown number of bases. The entries highlighted in green correspond to the actual number of available bases of the RAV. Duplicate entries of at least 100 optimal solutions with full coverage are not shown.

68

3.4.5 Uncertain Demand

Although the number of inhabitants in a postal code area is known to great certainty in The Netherlands, we can analyse the sensitivity to uncertainty in the demand. Assume that the weights of the demand points (the number of inhabitants) have the uncertainty structure as discussed in Section 3.2.1. We apply the Robust-Counterpart MCLP (Model 3.2.2) to construct a robust solution. This requires us to specify an estimated value $\hat{\delta}_j$ for the demand weight and its possible deviation η_j for $j \in \mathcal{J}$. Recall that the demand weight δ_j lies in the interval $[\hat{\delta}_j - \eta_j, \hat{\delta}_j + \eta_j]$. We have shown in Section 3.2.4.1 that a common relative deviation (e.g., 5%) would lead to the same solutions as the non-robust MCLP.

We can view the arrival of calls as a Poisson process, where the arrival rate is equal to the number of inhabitants in the postal code area. The rate of a Poisson process can be more accurately estimated for higher arrival rates. Therefore, we assume that areas with many inhabitants have a smaller relative deviation compared to areas with few inhabitants. A natural choice is to use the expected number of calls as estimated demand weight and a multiple of the standard deviation as possible perturbation. As the standard deviation of a Poisson distributed random variable is the square root of its expected value, areas with many inhabitants would have a smaller relative deviation (exactly as desired).

Thus, the estimated demand $\hat{\delta}_j$ is set to the number of inhabitants of that postal code area and the total demand Δ to the sum of all inhabitants in the RAV, $\Delta = \sum_{j \in \mathcal{J}} \hat{\delta}_j$. Finally, the perturbation η_j is set to a multiple $c_j \in \mathbb{N}$ of the square root of the estimated demand:

$$\delta_j \in \left[\hat{\delta}_j - c_j \sqrt{\hat{\delta}_j}, \hat{\delta}_j + c_j \sqrt{\hat{\delta}_j}\right]$$

As we do not allow negative weights, we require that $c_j \leq \sqrt{\hat{\delta}_j}$. In fact, we set

$$c_j = \min\left\{c, \sqrt{\hat{\delta}_j}\right\},\,$$

where $c \in \mathbb{N}$ is a fixed parameter for all points. For example, c = 0 corresponds to the nonrobust MCLP, as there is no perturbation. We have chosen three scenarios: $c \in \{1, 5, 10\}$, i.e., perturbations up to ten times the standard deviation. The reason is as follows. The average number of inhabitants in all postal code areas is approximately 4500. The three scenarios would lead to a deviation of 67 (1.5%), 335 (7.5%), and 670 (15%), respectively. Postal code areas with fewer inhabitants than average will have a higher relative deviation, areas with more inhabitants a lower relative deviation. Figure 3.4.10 shows the relative perturbation for the three scenarios of parameter c. Note that the relative perturbation is defined as $\eta_j/\hat{\delta}_j$.

In Section 3.4.2 we have seen that for the realistic number of bases most RAV regions have full coverage, in which case the MCLP and Robust Counterpart MCLP (RC-MCLP) are equivalent. Therefore, we iterate over the number of available bases, similar to the analysis of the sensitivity to the number of bases. Each time we compare the Robust Counterpart MCLP solution with the MCLP solution. In particular, we determine whether the MCLP solution has maximum robust coverage and whether the RC-MCLP has maximum non-robust coverage. In the first case, the MCLP solution is also robust. In the second case, the RC-MCLP solution is also optimal for the MCLP. If neither case applies, then there is a significant difference between the two solutions.



Figure 3.4.10: Relative perturbation in the number of inhabitants for parameter c. Dark blue corresponds with c = 1, light blue with c = 5, and cyan with c = 10.

Since there usually exist multiple optimal solutions, the considered solutions should maximise both the robust as well as the non-robust coverage. That is, the MCLP solution should have the best robust coverage of all optimal MCLP solutions, and similarly for the RC-MCLP. This is achieved using a two-stage approach (see also Section 3.4.3.1).

In almost all cases the solutions of both models have the same robust coverage, i.e., the MCLP is robust with respect to the considered uncertainty scenario. When there was a difference in robust coverage, also the non-robust coverage differed. The results are shown in Tables 3.4.8 and 3.4.9. Only the cases where the MCLP solution is not robust are given, the other entries are omitted. In particular, completely omitted are RAV4, RAV5, RAV8, RAV10, RAV11, RAV14 to RAV18, RAV20, RAV24, and RAV25. In all other cases the sets of covered points are equal between the two models. Note that the results are rounded to four decimals, so an absolute gap of 0.0000 implies there is a very small gap (see RAV1 with 10 bases for scenario c = 10).

We could not discern a pattern or rule for which number of available bases the MCLP is not robust: it ranges from one base to almost the required number for full coverage. Furthermore, RAV2 shows that whether the MCLP is robust can alternate with respect to the number of bases to open. However, the number of cases (where the MCLP is not robust) appears to be non-decreasing in the uncertainty parameter c.

For each uncertainty scenario we have evaluated 203 cases divided among the 24 RAV regions. Only in 32 of the 609 cases (5%) do the two models differ. If the MCLP solution is not robust, the absolute gap in coverage is very small: for robust and non-robust coverage all gaps are below 0.75% and most below 0.05%. Note that this also implies that the Robust Counterpart MCLP provides very good solutions when considering non-robust coverage. Most likely, these results heavily depend on the used uncertainty structure. However, we have considered two natural types of uncertainty structures: a fixed relative perturbation (theoretically analysed), and a structure based on the standard deviation of Poisson distributed random variables (empirically analysed). From these results, we have obtained no indications that either model is superior when considering robust and non-robust coverage.

		Robust Coverage									
		Scenario $c = 1$				Scenario $c =$	5	Scenario $c = 10$			
RAV Region	Opened Bases	MCLP	RC-MCLP	Absolute Gap	MCLP	RC-MCLP	Absolute Gap	MCLP	RC-MCLP	Absolute Gap	
1	6				0.8980	0.8981	0.0001	0.8861	0.8878	0.0017	
	10							0.9959	0.9959	0.0000	
2	5	0.7543	0.7543	0.0000	0.7258	0.7261	0.0003	0.6912	0.6921	0.0009	
	7				0.8284	0.8287	0.0003	0.8056	0.8064	0.0008	
	9				0.8899	0.8902	0.0003	0.8730	0.8738	0.0008	
	16				0.9894	0.9894	0.0001	0.9874	0.9875	0.0001	
3	5	0.8163	0.8170	0.0006	0.7974	0.8009	0.0035	0.7742	0.7811	0.0069	
	8							0.9508	0.9509	0.0001	
6	9				0.9806	0.9807	0.0001	0.9779	0.9784	0.0006	
7	3				0.8416	0.8417	0.0000	0.8285	0.8287	0.0002	
9	3							0.7639	0.7639	0.0000	
12	1	0.6270	0.6276	0.0007	0.6072	0.6109	0.0036	0.5827	0.5900	0.0073	
19	8							0.9608	0.9608	0.0000	
	9							0.9793	0.9793	0.0000	
21	3				0.8128	0.8149	0.0021	0.7976	0.8026	0.0050	
22	6				0.9872	0.9873	0.0001	0.9859	0.9863	0.0003	
23	3				0.7017	0.7023	0.0006	0.6779	0.6793	0.0013	

Table 3.4.8: Robust coverage of the MCLP and Robust Counterpart MCLP (RC-MCLP) for the three uncertainty scenarios and all possible number of bases to open. RAV regions and corresponding entries are only shown if the robust and non-robust coverage differ between the two models.

		Scenario $c = 1$				Scenario $c =$	5	Scenario $c = 10$			
RAV Region	Opened Bases	MCLP	RC-MCLP	Absolute Gap	MCLP	RC-MCLP	Absolute Gap	MCLP	RC-MCLP	Absolute Gap	
1	6				0.9100	0.9085	0.0014	0.9100	0.9085	0.0014	
	10							0.9974	0.9974	0.0001	
2	5	0.7614	0.7613	0.0001	0.7614	0.7613	0.0001	0.7614	0.7609	0.0005	
	7				0.8519	0.8514	0.0004	0.8519	0.8514	0.0004	
	9				0.9076	0.9072	0.0004	0.9076	0.9072	0.0004	
	16				0.9916	0.9916	0.0000	0.9916	0.9916	0.0000	
3	5	0.8211	0.8210	0.0001	0.8211	0.8210	0.0001	0.8211	0.8210	0.0001	
	8							0.9623	0.9620	0.0003	
6	9				0.9833	0.9829	0.0004	0.9833	0.9829	0.0004	
7	3				0.8548	0.8547	0.0001	0.8548	0.8547	0.0001	
9	3							0.7931	0.7929	0.0001	
12	1	0.6319	0.6318	0.0001	0.6319	0.6318	0.0001	0.6319	0.6318	0.0001	
19	8							0.9687	0.9680	0.0007	
	9							0.9844	0.9838	0.0007	
21	3				0.8280	0.8272	0.0008	0.8280	0.8272	0.0008	
22	6				0.9884	0.9883	0.0001	0.9884	0.9883	0.0001	
23	3				0.7255	0.7253	0.0002	0.7255	0.7253	0.0002	

Table 3.4.9: Non-robust coverage of the MCLP and Robust Counterpart MCLP (RC-MCLP) for the three uncertainty scenarios and all possible number of bases to open. RAV regions and corresponding entries are only shown if the robust and non-robust coverage differ between the two models.

3.5 Two Dutch Regional Ambulance Services in Detail

In the previous discussion of the numerical results we have only considered the attained coverage and not the actual positions of the ambulance bases. For the RAVs of The Netherlands we know the current positions of the bases. Therefore, we can compare the current base locations with those suggested by the EMS models. We will only consider two of the 24 RAV regions: RAV14 (Gooi- en Vechtstreek) and RAV23 (Limburg-Noord).

RAV14 is the smallest RAV region (3 bases and 40 demand points), making detailed visualisations possible. It has surplus capacity: only two bases are required for full coverage according to the MCLP. Furthermore, in Section 3.4.3 we have seen that the two-stage MCLP solution is extremely robust to changes in the travel time threshold, more than any other region.

RAV23 is larger than RAV14: it has 7 bases, 137 demand points, and cannot achieve full coverage (a shortage of one additional base). The fact that RAV23 has only two optimal MCLP solutions sets it apart from the other regions. Furthermore, when we restrict RAV23 to having three bases, there is a difference between the MCLP and the Robust Counterpart MCLP. That is, a different selection of bases is more robust to changes in demand.

In the following sections we discuss both RAVs, giving detailed information on the regions, their demand distributions, and the various EMS model solutions. We start with RAV14 in Section 3.5.1, followed by RAV23 in Section 3.5.2.

3.5.1 Gooi- en Vechtstreek

The Regional Ambulance Services 'Gooi- en Vechtstreek' (RAV14) is part of the province Noord-Holland, see the highlighted RAV region in Figure 3.5.1. The region consists of nine municipalities: Blaricum, Bussum, Hilversum, Huizen, Laren, Muiden, Naarden, Weesp, and Wijdemeren. It has 40 4-digit and approximately 7250 6-digit postal code areas. In Figure 3.5.2 the centres of the 4- and 6-digit areas are depicted and the corresponding city names are shown. Nederhorst den Berg, Ankeveen, Kortenhoef, and Breukeleveen are part of the municipality Wijdemeren, which has several large lakes (hence the name).



Figure 3.5.1: Gooi- en Vechtstreek (RAV14) of The Netherlands.



Figure 3.5.2: The centres of the 4- and 6-digit postal code areas of Gooi- en Vechtstreek (RAV14). The centres of the 6-digit areas are shown in green (unlabelled) and the centres of the 4-digit areas in magenta (labelled).



Figure 3.5.3: The number of inhabitants (in black) of each 4-digit postal code area (in magenta) of Gooi- en Vechtstreek (RAV14). The area of the green discs is proportional to the number of inhabitants.

Usually the number of 6-digit postal codes is proportional to the number of inhabitants in the area. This is indeed the case for RAV14: Figure 3.5.3 gives the number of inhabitants (in black) for each 4-digit postal code area (in magenta). Furthermore, the area of the green discs is proportional to the number of inhabitants, so densely populated areas are clearly visible. Most inhabitants in RAV14 live in the eastern part of the region. In the west the city Weesp has the most inhabitants.

From the sensitivity analysis we know that RAV14 can attain full coverage with only two of the three available bases. Figure 3.5.4 shows the optimal base locations when there are one and two bases available. In fact, the optimum shown for two bases (Figure 3.5.4b) is only one of the 16 optimal solutions (recall Table 3.4.7). All optimal base location pairs for RAV14 with two bases are given in Figure 3.5.5. Each optimal pair consists of one 'blue' base and one 'magenta' base. We conclude that with two bases available a centrally located base is required to be able to cover both the eastern and western demand points, and a southern base is needed to cover Breukeleveen (3625) and Nieuw Loosdrecht (1231).



(a) Optimum with 1 base.

(b) Optimum with 2 bases.

Figure 3.5.4: Optimal MCLP base locations (in blue) and covered (green) or uncovered (red) demand points for RAV14.



(a) First set of alternative optimal locations.

(b) Second set of alternative optimal locations.

Figure 3.5.5: Optimal MCLP base location pairs (the base in blue with one base in magenta) for RAV14 with 2 bases.

Adding a third base allows for many more optimal configurations: any optimal solution with two bases can be arbitrarily expanded. Furthermore, different optimal solutions are possible. For example, the current base locations of RAV14 lead to full coverage. These are located in the 4-digit postal code areas 1213 (Hilversum), 1261 (Blaricum), and 1381 (Weesp), see Figure 3.5.6a. Do note that the shown bases use the 4-digit (aggregated) postal code areas. For large postal code areas, such as 1261 (Blaricum), the actual base location can differ.

As discussed in Section 3.4.3, a two-stage optimisation approach can be used to select a preferred optimal MCLP solution. The two-stage MCLP solution obtained with the Minimum Travel Time Threshold model (Model 3.4.1) is shown in Figure 3.5.6b. The current base locations are similar to the two-stage MCLP solution. The two-stage MCLP solution cannot disregard Breukeleveen (3625), even though it has relatively few inhabitants. This explains why the base in Hilversum is located in the south-western part of the city.

Figure 3.5.7 depicts the optimal MCLP base locations for three travel time thresholds, namely for 6, 8, and 8.5 minutes. These correspond to the entries in Table 3.4.5. As full coverage can be attained with a travel time threshold of 506 seconds (almost 8.5 minutes), the solution in Figure 3.5.7c is also optimal for all larger thresholds. Finally, notice that the two-stage MCLP base locations closely correspond to the solutions in Figure 3.5.7, which explains its robustness with respect to the travel time threshold.



Figure 3.5.6: Base locations (blue) and covered (green) or uncovered (red) demand points for RAV14. The current locations and the two-stage MCLP solution are shown.



(a) Optimum for a travel time threshold of 6 minutes.



(b) Optimum for a travel time threshold of 8 minutes.



(c) Optimum for a travel time threshold of 8.5 minutes.

Figure 3.5.7: Optimal MCLP base locations (in blue) and covered (green) or uncovered (red) demand points for RAV14 with different travel time thresholds.

3.5.2 Limburg-Noord

The northern part of the province Limburg is the Regional Ambulance Services 'Limburg-Noord' (RAV23), depicted in Figure 3.5.8. Among its 15 municipalities are Roermond, Venlo, Venray, and Weert. The river Maas runs through the region, passing the western side of Roermond and Venlo, all the way north to pass Molenhoek. The region consists of 137 4-digit and approximately 14 000 6-digit postal code areas, see Figure 3.5.9. Unfortunately, we cannot show all cities of RAV23 due to the size of the region. Similar to RAV14, we have visualised the number of inhabitants in each 4-digit postal code area in Figure 3.5.10. We do not give the exact number of inhabitants, as the visualisation of many postal code areas overlap (smaller areas are displayed on top).

RAV23 can almost be covered completely with the seven available bases: the maximum MCLP coverage is 99.58%. Full coverage is possible with one additional base. In Figure 3.5.11 we give the optimal base locations for the shown number of available bases. Do note that with eight bases there are more than 100 optimal solutions (recall Table 3.4.7). Most base locations are relatively stable. For example, having a base in Venlo is optimal in all cases. When there are at least three bases, two should be assigned to Weert and Roermond. Finally, with six or more bases the northern demand points should be covered by a base in Ottersum.

Consider the configuration in Figure 3.5.11g, where seven bases are opened. The two uncovered demand points are Wellerlooi in the north and Buggenum in the south. Both bases lie on the other side of the river Maas with respect to their nearest base. This is most likely the reason why they cannot be reached in time.

Although the RAV region has many optimal solutions with eight opened bases, there are only two for the current number of bases (i.e., seven bases). These two optimal solutions are almost the same, see Figures 3.5.11g and 3.5.12b. Only the most southern base differs, but the two possible locations are adjacent and overlap partially (see also Figure 3.5.9). The alternative optimal solution corresponds to the two-stage MCLP solution using the Minimum Travel Time Threshold model. As discussed in Section 3.4.3.1, the two-stage MCLP solution does not lead to any improvement for RAV23. Given the similarity between the optimal solutions, this is not surprising.



Figure 3.5.8: Limburg-Noord (RAV23) of The Netherlands.



Figure 3.5.9: The centres of the 4- and 6-digit postal code areas of Limburg-Noord (RAV23). The centres of the 6-digit areas are shown in green and the centres of the 4-digit areas in magenta.



Figure 3.5.10: The number of inhabitants of each 4-digit postal code area of Limburg-Noord (RAV23). The area of the green discs is proportional to the number of inhabitants.



Figure 3.5.11: Optimal MCLP base locations (blue) and covered (green) or uncovered (red) demand points for RAV23.

In Figure 3.5.12a the current base locations are depicted. Again, we remark that these are aggregated into the 4-digit postal code areas. Nevertheless, it is clear that the northern demand points are uncovered. In practice, these are most likely partially served by other RAVs (Brabant-Noord and Gelderland-Zuid). Most current bases have a similar location as the optimal MCLP solution. For bases in or near the large cities (Roermond, Venlo, Venray, and Weert) this is expected. The most southern and the central current bases are also near the optimal MCLP locations.

Finally, we also consider the Robust Counterpart MCLP solutions (recall Table 3.4.8). The MCLP differs from the Robust Counterpart MCLP only when three bases can be opened, and only for the two scenarios with the largest uncertainty (c = 5 and c = 10). It turns out that the Robust Counterpart solution is the same for these two scenarios. The base locations are shown in Figure 3.5.13. Compared to the MCLP solution, the base north of Venlo is positioned in between Venlo and Venray. The two southern bases in Roermond and Weert are the same. The reason to reposition the northern base seems to be to cover both Venlo and Venray. Both cities have postal code areas with many inhabitants, which have a relatively lower uncertainty due to the chosen uncertainty structure.



(a) Current base locations.

(b) Minimum Travel Time Threshold two-stage MCLP solution.

Figure 3.5.12: Base locations (blue) and covered (green) or uncovered (red) demand points for RAV23. The current locations and the two-stage MCLP solution are shown.



Figure 3.5.13: Base locations (blue) and covered (green) or uncovered (red) demand points for RAV23. The MCLP and Robust Counterpart MCLP solutions are shown.

3.6 Conclusion

The simplicity of the Maximal Covering Location problem (MCLP) allowed us to perform an extensive sensitivity analysis of the model. For complex models the computational time required to determine the optimal solution can be too long for a complete analysis. If so, partial analyses can be performed or one can evaluate the sensitivity using smaller (possibly artificial) instances. In all cases, it is very useful to know the effect or impact a parameter has on the model.

We have analysed the sensitivity of the MCLP for instances based on the 24 Regional Ambulance Services (RAVs) of The Netherlands. If near-complete coverage is attainable with a decent number of bases, changing the number of bases to open has little impact on the resulting coverage. For the RAV 'Limburg-Noord' the base locations are relatively stable when the number of available bases changes. However, the base locations can also be very different, as seen for the RAV 'Gooi- en Vechtstreek'.

Alternative optimal solutions are rarely discussed in the emergency medical service literature. For the RAV regions in The Netherlands alternative optimal MCLP solutions have to be considered, as there are many due to surplus capacity. In general, alternative optimal solutions seem to exist if there is near-complete coverage (98% or more).

In particular, the alternative optimal solutions are of importance for the sensitivity to changes in the travel time threshold. The travel time threshold determines the coverage of each base. Most regions have relatively insensitive (i.e., robust) optimal MCLP solutions, but the MCLP cannot make the distinction. This can be counteracted by applying a two-stage optimisation approach, where the preferred optimal MCLP solution is selected from all optimal MCLP solutions.

The robustness of the MCLP with respect to uncertainty in demand can be determined by constructing the Robust Counterpart MCLP. The Robust Counterpart constructs a solution that has maximum worst-case coverage. We have considered a certain polytopal perturbation structure for the demand. Using the Robust Counterpart MCLP we can conclude that the MCLP is robust with respect to these fluctuations in demand, as both models generally result in the same base locations. The Robust Counterpart can be generalised to be applicable for other optimisation models as well.

The analysis can be extended in the following ways. First, other uncertainty structures can be considered for the uncertain demand. Second, the effect of aggregating the 6-digit postal codes into 4-digit areas is relevant. Finally, we have assumed that all considered postal code areas are suitable for an ambulance base. For 4-digit postal code areas this assumption is realistic, but not for 6-digit postal code areas. By restricting the set of possible base locations one would expect that the solutions are more robust to parameter changes due to lack of choice.

Chapter 4

Approximations for the MCLP

Contents

4.1 Int	roduction
4.2 Sub	modular Functions 110
4.3 Gre	edy Search for the MCLP 114
4.3.1	Greedy Guarantees for Submodular Functions
4.3.2	Greedy Approximation for the MCLP 119
4.3.3	Reversing the Greedy Search 123
4.4 Swa	ap Local Search for the MCLP 125
4.4.1	Swap Guarantees for Submodular Functions
4.4.2	Swap Locality Gap for the MCLP 128
4.5 Nu	merical Results
4.5.1	Greedy Search Performance for Realistic Instances
4.5.2	Swap Local Search Performance for Realistic Instances
4.5.3	Generated Instances
4.5.4	Performance for Generated Instances
4.6 Tw	o Dutch Regional Ambulance Services in Detail 158
4.6.1	Gooi- en Vechtstreek
4.6.2	Limburg-Noord
4.7 Cor	nclusion

4.1 Introduction

One of the main uses of emergency medical service (EMS) models is to relatively quickly analyse multiple 'what-if' scenarios to assess the consequences of EMS design decisions. The computational complexity of the EMS model and solution methods are vital for such research: if it takes an unacceptable amount of time to find an optimal solution for one scenario, a complete scenario analysis would be infeasible. We will analyse the computational complexity and solution methods for the Maximal Covering Location problem (MCLP). Recall that the MCLP is given by the following mathematical model:

Maximise	
$\sum d_j z_j$	
$j\in\mathcal{J}$	
subject to	
$\sum_{i \in \mathcal{I}} x_i \le p,$	
$\sum_{i\in\mathcal{I}}a_{ij}x_i\geq z_j$	$\forall j \in \mathcal{J},$
$x_i \in \mathbb{B}$	$\forall i \in \mathcal{I},$
$z_j \in \mathbb{B}$	$\forall j \in \mathcal{J}.$

The MCLP is NP-hard, i.e., it cannot be solved to optimality efficiently (unless P = NP). The NP-hardness can be derived by a straightforward polynomial time reduction from the Set Covering problem to the MCLP with uniform demand ($d_j = 1$ for all $j \in \mathcal{J}$). The Set Covering problem is equivalent to the Location Set Covering model (Model 2.2.1) and Karp (1972) has shown that it is NP-hard. For more details on the proof and NP-hardness, we refer to Appendix A.

The MCLP can be solved (reasonably) well with Branch-and-Bound procedures using Linear Programming (LP) relaxations, as the LP relaxation usually results in few fractional values (see Church and ReVelle (1974)). We will consider two heuristics to construct or improve solutions for the MCLP: the Greedy Search and the Swap Local Search. The Greedy Search constructs a feasible solution by iteratively opening an additional subset of bases. Which bases are opened is determined in a greedy way. The Swap Local Search improves an initial feasible solution by closing a subset of opened bases and opening a different subset of bases.

When applying heuristics, it is useful to determine whether guarantees can be given on the quality of the resulting solution. Consider a heuristic H. Let Ω be the set of all MCLP instances. For each MCLP instance $\omega \in \Omega$ we have the objective value $\theta^H(\omega)$ of the heuristic solution and the global maximum $\theta^*(\omega)$. In general, the heuristic objective value is smaller than the global maximum. Hence, there is a gap between the two. For each MCLP instance we can determine this gap a posteriori (provided that we know the optimum). However, can we also guarantee a bound on the gap a priori? That is, what is the worst possible gap for the heuristic when considering all MCLP instances?
We define the relative optimality gap for instance $\omega \in \Omega$ as

$$\frac{\theta^*(\omega) - \theta^H(\omega)}{\theta^*(\omega)}.$$

For $\alpha \in [0, 1]$, we say that heuristic H has an α -guarantee (a bound) on the relative optimality gap if

$$\frac{\theta^*(\omega) - \theta^H(\omega)}{\theta^*(\omega)} \le (1 - \alpha) \qquad \qquad \forall \, \omega \in \Omega.$$

An equivalent definition is given by

$$\theta^H(\omega) \ge \alpha \theta^*(\omega) \qquad \qquad \forall \, \omega \in \Omega.$$

We call such a bound tight if

$$\sup_{\omega \in \Omega} \left\{ \frac{\theta^*(\omega) - \theta^H(\omega)}{\theta^*(\omega)} \right\} = (1 - \alpha).$$

For instance, an α -guarantee of $\alpha = 3/4$ implies that the difference between the heuristic objective value and the global maximum is at most 1/4 of the global maximum. A higher value of $\alpha \in [0, 1]$ implies a better guaranteed performance for the heuristic. Also note that $\alpha = 0$ is the trivial lower bound for MCLP (as the objective is always non-negative).

Some NP-hard problems can be solved approximately to any degree of accuracy by so-called approximation schemes. However, Feige (1998) proved that the best possible polynomial time guarantee for the MCLP is $1 - e^{-1}$ (unless P = NP), see Theorem 4.1.1.

Theorem 4.1.1 (Feige (1998)). For any $\epsilon > 0$ the MCLP cannot be approximated in polynomial time with a guarantee of $1 - e^{-1} + \epsilon$, unless P = NP.

We will discuss several heuristics for the MCLP and provide guarantees on the quality of the heuristic solutions. It is useful to introduce submodular functions, as the MCLP corresponds to maximising a submodular function. Submodularity is discussed in Section 4.2. Using the general framework of submodular functions we can derive performance guarantees for the Greedy Search and Swap Local Search for the MCLP (Sections 4.3 and 4.4). The performance of the heuristics is numerically evaluated in Sections 4.5 and 4.6.

4.2 Submodular Functions

Submodular functions encompass a variety of discrete functions, similar to concave and convex functions in continuous optimisation. The key property of a submodular function is the decrease in marginal gain in function value as the size of the argument increases. To be precise, see Definition 4.2.1.

Definition 4.2.1. A function $\phi : 2^{\mathcal{N}} \to \mathbb{R}$ for some finite discrete set \mathcal{N} is called submodular if

$$\phi(\mathcal{U}) + \phi(\mathcal{V}) - \phi(\mathcal{U} \cap \mathcal{V}) \ge \phi(\mathcal{U} \cup \mathcal{V}) \qquad \qquad \forall \mathcal{U}, \mathcal{V} \subseteq \mathcal{N}.$$

There are several equivalent definitions for submodular functions, see also Nemhauser, Wolsey, and Fisher (1978) for an overview. We will state two of these additional definitions in Proposition 4.2.2, albeit slightly modified.

Proposition 4.2.2. The following properties are equivalent and define a submodular function $\phi: 2^{\mathcal{N}} \to \mathbb{R}$:

(i)
$$\phi(\mathcal{U}) + \phi(\mathcal{V}) - \phi(\mathcal{U} \cap \mathcal{V}) \ge \phi(\mathcal{U} \cup \mathcal{V}) \text{ for all } \mathcal{U}, \mathcal{V} \subseteq \mathcal{N}.$$

- (ii) $\phi(\mathcal{U} \cup \mathcal{W}) \phi(\mathcal{U}) \ge \phi(\mathcal{V} \cup \mathcal{W}) \phi(\mathcal{V})$ for all $\mathcal{U} \subseteq \mathcal{V} \subseteq \mathcal{N}$ and $\mathcal{W} \subseteq \mathcal{N} \setminus \mathcal{V}$.
- (iii) $\phi(\mathcal{V}) \leq \phi(\mathcal{U}) + \sum_{n=1}^{a} \left(\phi(\mathcal{U} \cup \mathcal{X}_n) \phi(\mathcal{U}) \right) \sum_{n=1}^{b} \left(\phi(\mathcal{U} \cup \mathcal{V}) \phi((\mathcal{U} \setminus \mathcal{Y}_n) \cup \mathcal{V}) \right)$ for all $\mathcal{U}, \mathcal{V} \subseteq \mathcal{N}$ and partitions $\mathcal{V} \setminus \mathcal{U} = \bigcup_{n=1}^{a} \mathcal{X}_n$ and $\mathcal{U} \setminus \mathcal{V} = \bigcup_{n=1}^{b} \mathcal{Y}_n$ with $a, b \in \mathbb{N}$.

Proof. We will show the equivalence of (i) and (ii), followed by that of (ii) and (iii).

• (i)
$$\implies$$
 (ii)

Let $\mathcal{U} \subseteq \mathcal{V} \subseteq \mathcal{N}$ and $\mathcal{W} \subseteq \mathcal{N} \setminus \mathcal{V}$. It holds that

$$\phi(\mathcal{U} \cup \mathcal{W}) + \phi(\mathcal{V}) - \phi(\mathcal{U}) = \phi(\mathcal{U} \cup \mathcal{W}) + \phi(\mathcal{V}) - \phi((\mathcal{U} \cup \mathcal{W}) \cap \mathcal{V})$$

$$\stackrel{(i)}{\geq} \phi(\mathcal{U} \cup \mathcal{W} \cup \mathcal{V}) = \phi(\mathcal{V} \cup \mathcal{W}),$$

which is equivalent to (ii).

• (ii)
$$\Longrightarrow$$
 (i)

Let $\mathcal{U}, \mathcal{V} \subseteq \mathcal{N}$, we have:

$$\phi(\mathcal{U}) - \phi(\mathcal{U} \cap \mathcal{V}) = \phi((\mathcal{U} \cap \mathcal{V}) \cup (\mathcal{U} \setminus \mathcal{V})) - \phi(\mathcal{U} \cap \mathcal{V})$$

$$\stackrel{(\mathrm{ii})}{\geq} \phi(\mathcal{V} \cup (\mathcal{U} \setminus \mathcal{V})) - \phi(\mathcal{V}) = \phi(\mathcal{U} \cup \mathcal{V}) - \phi(\mathcal{V})$$

Reordering the terms gives (i).

• (ii) \implies (iii)

Let $\mathcal{U}, \mathcal{V} \subseteq \mathcal{N}$ with any partition $\mathcal{V} \setminus \mathcal{U} = \bigcup_{n=1}^{a} \mathcal{X}_n$ and $\mathcal{U} \setminus \mathcal{V} = \bigcup_{n=1}^{b} \mathcal{Y}_n$ with $a, b \in \mathbb{N}$. Notice that

$$\phi(\mathcal{U}\cup\mathcal{V}) - \phi(\mathcal{U}) = \sum_{n=1}^{a} \left(\phi\left(\mathcal{U}\cup\bigcup_{n'=1}^{n}\mathcal{X}_{n'}\right) - \phi\left(\mathcal{U}\cup\bigcup_{n'=1}^{n-1}\mathcal{X}_{n'}\right) \right)$$

$$\stackrel{\text{(ii)}}{\leq} \sum_{n=1}^{a} \left(\phi(\mathcal{U}\cup\mathcal{X}_{n}) - \phi(\mathcal{U}) \right).$$
(4.2.1)

Similarly, we have

$$\phi(\mathcal{U}\cup\mathcal{V}) - \phi(\mathcal{V}) = \sum_{n=1}^{b} \left(\phi\left(\mathcal{V}\cup\bigcup_{n'=1}^{n}\mathcal{Y}_{n'}\right) - \phi\left(\mathcal{V}\cup\bigcup_{n'=1}^{n-1}\mathcal{Y}_{n'}\right) \right)$$
$$\stackrel{(\mathrm{ii})}{\geq} \sum_{n=1}^{b} \left(\phi(\mathcal{U}\cup\mathcal{V}) - \phi((\mathcal{U}\setminus\mathcal{Y}_{n})\cup\mathcal{V}) \right).$$
(4.2.2)

Subtracting Equation (4.2.2) from (4.2.1) results in (iii):

$$\phi(\mathcal{V}) - \phi(\mathcal{U}) \leq \sum_{n=1}^{a} \left(\phi(\mathcal{U} \cup \mathcal{X}_n) - \phi(\mathcal{U}) \right) - \sum_{n=1}^{b} \left(\phi(\mathcal{U} \cup \mathcal{V}) - \phi((\mathcal{U} \setminus \mathcal{Y}_n) \cup \mathcal{V}) \right).$$

• (iii) \Longrightarrow (ii)

Let $\mathcal{U} \subseteq \mathcal{V} \subseteq \mathcal{N}$ and $\mathcal{W} \subseteq \mathcal{N} \setminus \mathcal{V}$. Note that $\mathcal{U} \setminus \mathcal{V} = \emptyset$, so we can set b = 0 (no subsets \mathcal{Y}_n). Consider the partition $\mathcal{X}_1 = \mathcal{V} \setminus \mathcal{U}$ and $\mathcal{X}_2 = \mathcal{W}$ of $(\mathcal{V} \cup \mathcal{W}) \setminus \mathcal{U}$. We use (iii) for the following:

$$\phi(\mathcal{V} \cup \mathcal{W}) \stackrel{(\mathrm{III})}{\leq} \phi(\mathcal{U}) + \phi(\mathcal{U} \cup (\mathcal{V} \setminus \mathcal{U})) - \phi(\mathcal{U}) + \phi(\mathcal{U} \cup \mathcal{W}) - \phi(\mathcal{U})$$
$$= \phi(\mathcal{V}) + \phi(\mathcal{U} \cup \mathcal{W}) - \phi(\mathcal{U}).$$

Again, reordering the terms results in (ii).

As mentioned before, for submodular functions the marginal function value $\phi(\mathcal{U} \cup \mathcal{W}) - \phi(\mathcal{U})$ decreases when elements are added to the argument set \mathcal{U} . This interpretation follows clearly from property (ii). From property (iii) we can derive further properties for non-decreasing submodular functions, see Proposition 4.2.3.

Proposition 4.2.3. Consider a non-decreasing submodular function $\phi : 2^{\mathcal{N}} \to \mathbb{R}$. For all sets $\mathcal{U}, \mathcal{V} \subseteq \mathcal{N}$ and partitions $\mathcal{V} = \bigcup_{n=1}^{a} \mathcal{V}_n$ with $a \in \mathbb{N}_{\geq 1}$ it holds that:

$$\phi(\mathcal{V}) \le \phi(\mathcal{U}) + \sum_{n=1}^{a} \left(\phi(\mathcal{U} \cup \mathcal{V}_n) - \phi(\mathcal{U})\right).$$

Proof. Since ϕ is non-decreasing, we can bound the right-hand side of (iii) as follows. Partition $\mathcal{V} \setminus \mathcal{U}$ into $a \in \mathbb{N}_{\geq 1}$ disjoint subsets $\{\mathcal{X}_n\}_{n=1}^a$ in such a way that $\mathcal{X}_n \subseteq \mathcal{V}_n$ for all $n \in \{1, \ldots, a\}$ (e.g., $\mathcal{X}_n = \mathcal{V}_n \setminus \mathcal{U}$). Note that some of the subsets \mathcal{X}_n can be empty. Partition $\mathcal{U} \setminus \mathcal{V} = \bigcup_{n=1}^b \mathcal{Y}_n$ arbitrarily. It holds that:

$$\begin{split} \phi(\mathcal{V}) &\stackrel{\text{(iii)}}{\leq} \phi(\mathcal{U}) + \sum_{n=1}^{a} \left(\phi(\mathcal{U} \cup \mathcal{X}_{n}) - \phi(\mathcal{U}) \right) - \sum_{n=1}^{b} \left(\phi(\mathcal{U} \cup \mathcal{V}) - \phi((\mathcal{U} \setminus \mathcal{Y}_{n}) \cup \mathcal{V}) \right) \\ &\leq \phi(\mathcal{U}) + \sum_{n=1}^{a} \left(\phi(\mathcal{U} \cup \mathcal{X}_{n}) - \phi(\mathcal{U}) \right) \\ &\leq \phi(\mathcal{U}) + \sum_{n=1}^{a} \left(\phi(\mathcal{U} \cup \mathcal{V}_{n}) - \phi(\mathcal{U}) \right), \end{split}$$

where we have used twice that ϕ is non-decreasing.

A broad class of optimisation problems can be captured by minimising or maximising a certain submodular function. In general, minimising a submodular function can be done in strongly polynomial time, see Grötschel et al. (1981) and Schrijver (2000). However, maximisation is in general difficult (NP-hard). Therefore, the design of approximations is frequently investigated in the literature, see for example Feige et al. (2007), Vondrák (2007), and Vondrák (2009). Approximating submodular functions using polynomial function value queries is discussed in Goemans et al. (2009). Fast heuristics with performance guarantees for the maximisation of submodular functions are given by Ashwinkumar and Vondrák (2014).

In particular, the Maximum *p*-Coverage problem is analysed in the literature. Let $\phi : 2^{\mathcal{N}} \to \mathbb{R}$ be a non-decreasing submodular function and $p \in \mathbb{N}$. The Maximum *p*-Coverage problem for $p \in \mathbb{N}$ is given by the optimisation problem

$$\max\left\{\phi(\mathcal{U}): |\mathcal{U}| \le p, \mathcal{U} \subseteq \mathcal{N}\right\}.$$

This optimisation problem is NP-hard and Feige (1998) and Nemhauser and Wolsey (1978) proved that the best possible performance guarantee in polynomial time is $1-e^{-1}$ unless P = NP, see Theorem 4.2.4. Related results are treated in Vondrák (2009).

Theorem 4.2.4 (Feige (1998) and Nemhauser and Wolsey (1978)). Let $\phi : 2^{\mathcal{N}} \to \mathbb{R}$ be a non-decreasing submodular function and $p \in \mathbb{N}$. Consider the optimisation problem

$$\max \left\{ \phi(\mathcal{U}) : |\mathcal{U}| \le p, \mathcal{U} \subseteq \mathcal{N} \right\}$$

For any $\epsilon > 0$ this optimisation problem cannot be approximated in polynomial time with a guarantee of $1 - e^{-1} + \epsilon$, unless P = NP. Furthermore, no algorithm requiring polynomially many value queries of ϕ can have a better guarantee.

For approximations and bounds, see Hochbaum and Pathria (1998), Nemhauser and Wolsey (1978), and Nemhauser, Wolsey, and Fisher (1978). Results for generalisations of the Maximum p-Coverage problem are discussed in for example M. Conforti and Cornuéjols (1984) and Goundan and Schulz (2007). The Maximum p-Coverage problem can also be seen as maximising a submodular function under a p-uniform matroid¹ constraint.

The MCLP is a special case of the Maximum *p*-Coverage problem, where $\phi(U)$ is defined to be the demand of covered points by bases in $\mathcal{U} \subseteq I$. To be precise, notice that the MCLP is equivalent to:

$$\max\left\{\phi(\mathcal{U}): |\mathcal{U}| \le p, \mathcal{U} \subseteq \mathcal{I}\right\} = \max\left\{\sum_{j \in \mathcal{J}} d_j \max\left\{a_{ij}: i \in \mathcal{U}\right\}: |\mathcal{U}| \le p, \mathcal{U} \subseteq \mathcal{I}\right\}.$$
 (4.2.3)

This equivalence does not hold for negative weights d_j , unless the MCLP is reformulated. The standard formulation allows z_j to be set to zero (uncovered point) for points $j \in \mathcal{J}$ with negative demand, which is not possible in Equation (4.2.3).

1. if $\mathcal{U} \subseteq \mathcal{V}$ and $\mathcal{V} \in \mathcal{M}$ then $\mathcal{U} \in \mathcal{M}$,

2. if $\mathcal{U}, \mathcal{V} \in \mathcal{M}$ and $|\mathcal{U}| < |\mathcal{V}|$ then there exists a $\nu \in \mathcal{V} \setminus \mathcal{U}$ such that $\mathcal{U} \cup \{\nu\} \in \mathcal{M}$.

For the *p*-uniform matroid the independent sets are defined as $\mathcal{M} = \{\mathcal{U} \subseteq \mathcal{N} : |\mathcal{U}| \le p\}.$

¹A matroid is a combinatorial structure with a finite ground set \mathcal{N} and a family of so-called independent sets $\mathcal{M} \subseteq \mathcal{N}$ satisfying:

We show that ϕ defined as in Equation (4.2.3) is indeed a submodular function. Let $\mathcal{U} \subseteq \mathcal{V} \subseteq \mathcal{I}$ and $\mathcal{W} \subseteq \mathcal{I} \setminus \mathcal{V}$:

$$\begin{split} \phi(\mathcal{V} \cup \mathcal{W}) - \phi(\mathcal{U} \cup \mathcal{W}) &= \sum_{j \in \mathcal{J}} d_j \max\{a_{ij} : i \in \mathcal{V} \setminus \mathcal{W}\} - \sum_{j \in \mathcal{J}} d_j \max\{a_{ij} : i \in \mathcal{U} \setminus \mathcal{W}\} \\ &= \sum_{j \in \mathcal{J}} d_j \max\{a_{ij} : i \in \mathcal{V}\} - \sum_{j \in \mathcal{J}} d_j \max\{a_{ij} : i \in \mathcal{U}\} \\ &= \phi(\mathcal{V}) - \phi(\mathcal{U}). \end{split}$$

Therefore, submodular property (ii) holds and ϕ is a submodular function. It is trivial that ϕ is non-decreasing and non-negative for the MCLP. Also note that $\phi(\emptyset) = 0$.

Due to the inherent decrease in marginal gain in EMS models, submodular functions can be used as a general framework. However, it is important to verify submodularity for each EMS model and analyse whether the constraints fall into known structures (such as matroids).

4.3 Greedy Search for the MCLP

A straightforward way to construct reasonably good feasible solution for the Maximal Covering Location problem is to use Greedy Search. The Greedy Search method starts with all bases closed and iteratively adds a subset of bases to be opened. The method stops when p bases are opened or when opening any additional base does not improve coverage. At each iteration at most $\rho \in \mathbb{N}_{\geq 1}$ additional bases can be opened and the selection of the additional bases is done in a greedy way: the chosen subset of bases leads to the greatest improvement in objective value with respect to the previous iteration.

For the MCLP and non-decreasing submodular functions there is an optimal solution that opens exactly p bases. We can therefore focus on the Greedy Search that opens exactly ρ bases in each iteration (except possibly the last iteration), see also Algorithm 4.3.1. In Appendix B we give a detailed Greedy Search algorithm for the MCLP.

Algorithm 4.3.1 Greedy Search for Non-Decreasing Submodular Functions

Input: non-decreasing submodular function $\phi : 2^{\mathcal{N}} \to \mathbb{R}$ and parameters $p, \rho \in \mathbb{N}$ with $p \leq |\mathcal{N}|$ **Output:** ρ -Greedy maximum solution $\mathcal{N}^G \subseteq \mathcal{N}$

1: procedure ρ -GREEDY SEARCH(input) set: $\mathcal{N}^G = \emptyset$ 2: set: $\tau = \min\{\rho, p - |\mathcal{N}^G|\}$ 3: while $\tau > 0$ do 4: set: $\mathcal{N}^G = \mathcal{N}^G \cup \operatorname{argmax} \{ \phi(\mathcal{N}^G \cup \mathcal{N}^+) : \mathcal{N}^+ \subseteq \mathcal{N} \setminus \mathcal{N}^G, |\mathcal{N}^+| = \tau \}$ 5: set: $\tau = \min\{\rho, p - |\mathcal{N}^G|\}$ 6: end while 7: return \mathcal{N}^G 8: 9: end procedure

For some submodular functions the Greedy Search has a polynomial computational time (i.e., the heuristic is mathematically efficient). If we can derive an α -guarantee for the Greedy Search and the search has polynomial computational time, then it is called an α -approximation algorithm, see Definition 4.3.1.

Definition 4.3.1. Consider a maximisation problem with a heuristic solution procedure H and let Ω be the set of all instances. Each instance $\omega \in \Omega$ has a heuristic solution with objective value $\theta^{H}(\omega)$ and global maximum $\theta^{*}(\omega)$. The heuristic H is called an α -approximation algorithm with $\alpha \in [0, 1]$ if the computational time of H for every instance in Ω is polynomial and if

$$\frac{\theta^*(\omega) - \theta^H(\omega)}{\theta^*(\omega)} \le (1 - \alpha) \qquad \qquad \forall \, \omega \in \Omega.$$

The Greedy Search method provides reasonably good solutions for non-decreasing submodular functions: bounds on the optimality gap are known, see for instance Nemhauser, Wolsey, and Fisher (1978). We will provide these performance guarantees in Section 4.3.1. Using these performance guarantees, we show in Section 4.3.2 that the 1-Greedy Search for the MCLP is a $(1 - e^{-1})$ -approximation algorithm and this guarantee is tight.

4.3.1 Greedy Guarantees for Submodular Functions

Before we discuss performance guarantees for the Greedy Search, we derive a useful property of the Greedy Search for submodular functions, see Proposition 4.3.2. It shows that the average marginal contribution of an element to the function value is the greatest when selecting elements in a greedy way. This holds in particular when comparing sets of different sizes.

Proposition 4.3.2. Let $\phi : 2^{\mathcal{N}} \to \mathbb{R}$ be a non-decreasing submodular function and $\mathcal{U}^G \subseteq \mathcal{N}$ the subset selected by applying one iteration of the ρ -Greedy Search for some $\rho \in \mathbb{N}_{\geq 1}$. That is,

$$\mathcal{U}^G = \operatorname{argmax} \left\{ \phi(\mathcal{U}) : \mathcal{U} \subseteq \mathcal{N}, |\mathcal{U}| = \rho \right\}.$$

Consider an arbitrary subset $\mathcal{V} \subseteq \mathcal{N}$. If $|\mathcal{V}| < |\mathcal{U}^G|$ we have $\phi(\mathcal{U}^G) \ge \phi(\mathcal{V})$. If $|\mathcal{V}| \ge |\mathcal{U}^G|$ the following holds:

$$\phi(\mathcal{U}^G) \ge \frac{|\mathcal{U}^G|}{|\mathcal{V}|} \phi(\mathcal{V}).$$

Proof. The case $|\mathcal{V}| < |\mathcal{U}^G|$ is trivial by greediness, since ϕ is non-decreasing. Therefore, suppose $|\mathcal{V}| \ge |\mathcal{U}^G|$. We are going to apply Proposition 4.2.3 to \mathcal{V} with a certain partition of \mathcal{V} . Let $\mathcal{M}_{\max} \subseteq \mathcal{V}$ with $|\mathcal{M}| = |\mathcal{U}^G|$ be the subset of \mathcal{V} with cardinality $|\mathcal{U}^G|$ that would lead to the greatest increase in function value. That is,

$$\mathcal{M}_{\max} = \operatorname{argmax} \left\{ \phi(\mathcal{M}) - \phi(\emptyset) : \mathcal{M} \subseteq \mathcal{V}, |\mathcal{M}| = |\mathcal{U}^G| \right\}.$$

We have by greediness of the search that $\phi(\mathcal{U}^G) - \phi(\emptyset) \ge \phi(\mathcal{M}_{\max}) - \phi(\emptyset)$. Order and label the elements of \mathcal{M}_{\max} such that $\mathcal{M}_{\max} = \{\nu_1, \ldots, \nu_{|\mathcal{U}^G|}\}$ and

$$\zeta_1 \geq \ldots \geq \zeta_{|\mathcal{U}^G|} \geq 0,$$

where the marginal increase in function value ζ_n is defined by:

$$\zeta_n = \phi\left(\bigcup_{n'=1}^n \{\nu_{n'}\}\right) - \phi\left(\bigcup_{n'=1}^{n-1} \{\nu_{n'}\}\right) \qquad \forall n \in \{1, \dots, |\mathcal{U}^G|\}.$$

Note that the 1-Greedy Search selection order of elements of \mathcal{M}_{max} is such an ordering.

Partition and label the elements in \mathcal{V} into $|\mathcal{V}| - |\mathcal{U}^G| + 1$ subsets: $\mathcal{V} = \mathcal{M}_{\max} \cup \bigcup_{k=|\mathcal{U}^G|+1}^{|\mathcal{V}|} \{\nu_k\}$. By definition of \mathcal{M}_{\max} and submodularity, we have for any $k \in \{|\mathcal{U}^G| + 1, \dots, |\mathcal{V}|\}$:

$$\phi\left(\mathcal{M}_{\max}\cup\{\nu_k\}\right) - \phi\left(\mathcal{M}_{\max}\right) \leq \phi\left(\bigcup_{n'=1}^{|\mathcal{U}^G|-1}\{\nu_{n'}\}\cup\{\nu_k\}\right) - \phi\left(\bigcup_{n'=1}^{|\mathcal{U}^G|-1}\{\nu_{n'}\}\right)$$
$$\leq \zeta_{|\mathcal{U}^G|} \leq \frac{1}{|\mathcal{U}^G|}\sum_{n=1}^{|\mathcal{U}^G|}\zeta_n = \frac{1}{|\mathcal{U}^G|}\left(\phi\left(\mathcal{M}_{\max}\right) - \phi\left(\emptyset\right)\right)$$
$$\leq \frac{1}{|\mathcal{U}^G|}(\phi(\mathcal{U}^G) - \phi(\emptyset)).$$

Finally, Proposition 4.2.3 and the above derived results show that

$$\begin{split} \phi(\mathcal{V}) - \phi(\emptyset) &\leq \phi\left(\mathcal{M}_{\max}\right) - \phi\left(\emptyset\right) + \sum_{k=|\mathcal{U}^G|+1}^{|\mathcal{V}|} \left(\phi\left(\mathcal{M}_{\max} \cup \{\nu_k\}\right) - \phi\left(\mathcal{M}_{\max}\right)\right) \\ &\leq \left(1 + \frac{|\mathcal{V}| - |\mathcal{U}^G|}{|\mathcal{U}^G|}\right) \left(\phi(\mathcal{U}^G) - \phi(\emptyset)\right) = \frac{|\mathcal{V}|}{|\mathcal{U}^G|} (\phi(\mathcal{U}^G) - \phi(\emptyset)). \end{split}$$

The desired result follows from the fact that $|\mathcal{V}| \geq |\mathcal{U}^G|$.

In Nemhauser, Wolsey, and Fisher (1978) performance guarantees are given for the Greedy Search method for non-decreasing submodular functions. See Theorem 4.3.3 for their result and proof.

Theorem 4.3.3 (Nemhauser, Wolsey, and Fisher (1978)). Let $\phi : 2^{\mathcal{N}} \to \mathbb{R}$ be a non-decreasing submodular function and $p \in \mathbb{N}$. Consider the optimisation problem

$$\max\left\{\phi(\mathcal{U}): |\mathcal{U}| \le p, \mathcal{U} \subseteq \mathcal{N}\right\}$$

The ρ -Greedy Search is applied to this problem, where $p = a\rho - b$ with $\rho, a \in \mathbb{N}_{\geq 1}$ and $b \in \{0, \ldots, \rho - 1\}$ (see Algorithm 4.3.1). The following bound holds for the ρ -Greedy maximum θ^G and the global maximum θ^* :

$$\frac{\theta^* - \theta^G}{\theta^* - \phi(\emptyset)} \le \left(\frac{a - \frac{1}{\rho}(\rho - b)}{a}\right) \left(\frac{a - 1}{a}\right)^{a - 1}$$

If b = 0, then this bound is tight.

Proof. Let Φ be the set of all non-decreasing submodular functions on \mathcal{N} . For $\phi \in \Phi$ we denote the corresponding ρ -Greedy maximum by $\theta^G(\phi)$ and the corresponding global maximum by $\theta^*(\phi)$. We want to derive an upper bound on the relative gap between $\theta^*(\phi)$ and $\theta^G(\phi)$. Note that the upper bound holds for any submodular function $\phi \in \Phi$. Therefore, we are interested in an upper bound for the optimisation problem

$$\sup\left\{\frac{\theta^*(\phi) - \theta^G(\phi)}{\theta^*(\phi) - \phi(\emptyset)} : \phi \in \Phi\right\} = 1 - \inf\left\{\frac{\theta^G(\phi) - \phi(\emptyset)}{\theta^*(\phi) - \phi(\emptyset)} : \phi \in \Phi\right\}$$
$$= 1 - \inf\left\{\theta^G(\phi) - \phi(\emptyset) : \phi \in \Phi, \theta^*(\phi) - \phi(\emptyset) = 1\right\}.$$
(4.3.1)

Suppose we have a value $\theta'(\phi)$ related to $\theta^G(\phi)$ with $\theta^G(\phi) \ge \theta'(\phi)$ for all $\phi \in \Phi$. In this case, any upper bound for

$$1 - \inf \left\{ \theta'(\phi) - \phi(\emptyset) : \phi \in \Phi, \theta^*(\phi) - \phi(\emptyset) = 1 \right\}$$

$$(4.3.2)$$

would also be valid² for the supremum in Equation (4.3.1). We will define $\theta'(\phi)$ as follows.

Fix $\phi \in \Phi$ and let $\mathcal{N}^G \subseteq \mathcal{N}$ be the ρ -Greedy solution. The Greedy solution has a natural partition into *a* disjoint sets, $\{\mathcal{U}_n^G\}_{n=1}^a$, one for each iteration. Note that the first (a-1) sets have cardinality ρ and the last set has cardinality $(\rho - b)$.

²For example, setting $\theta'(\phi) = \phi(\emptyset)$ for all $\phi \in \Phi$ leads to the trivial upper bound of 1.

By greediness of the search it holds that

$$\delta_1 \geq \ldots \geq \delta_a \geq 0,$$

where δ_n is the net increase in objective value at Greedy step $n \in \{1, \ldots, a\}$:

$$\delta_n = \phi\left(\bigcup_{n'=1}^n \mathcal{U}_{n'}^G\right) - \phi\left(\bigcup_{n'=1}^{n-1} \mathcal{U}_{n'}^G\right) \qquad \forall n \in \{1, \dots, a\}.$$

Similarly, define η_n as the maximum possible increase at each Greedy step $n \in \{1, ..., a\}$ if exactly ρ elements can be selected:

$$\eta_n = \max\left\{\phi\left(\bigcup_{n'=1}^{n-1} \mathcal{U}_{n'}^G \cup \mathcal{N}^+\right) - \phi\left(\bigcup_{n'=1}^{n-1} \mathcal{U}_{n'}^G\right) : \mathcal{N}^+ \subseteq \mathcal{N}, |\mathcal{N}^+| = \rho\right\} \quad \forall n \in \{1, \dots, a\}.$$

Notice that \mathcal{N}^+ is allowed to overlap with elements already selected by the Greedy Search. Furthermore, we have that $\delta_n = \eta_n$ for $n \in \{1, \ldots, a-1\}$, but $\eta_a \ge \delta_a$. In fact, we can use η_a to give two bounds for δ_a :

$$\left(\frac{\rho-b}{\rho}\right)\eta_a \le \delta_a \le \eta_a.$$

The first inequality follows from Proposition 4.3.2 with a $(\rho - b)$ -Greedy Search iteration applied. Hence, a lower bound for θ^G is

$$\theta^G = \phi(\emptyset) + \sum_{n=1}^a \delta_n \ge \phi(\emptyset) + \sum_{n=1}^{a-1} \eta_n + \left(\frac{\rho - b}{\rho}\right) \eta_a \equiv \theta'.$$

This definition of θ' is strongly related to the Greedy Search solution and θ^G .

As mentioned before, any upper bound for (4.3.2) is valid for the supremum in Equation (4.3.1). Therefore, we focus on the infimum

$$\inf \left\{ \theta'(\phi) - \phi(\emptyset) : \phi \in \Phi, \theta^*(\phi) - \phi(\emptyset) = 1 \right\}.$$
(4.3.3)

We will derive a bound for (4.3.3) by solving a Linear Programming model. The constraints of this model are derived as follows. Let $\{\mathcal{U}_n^*\}_{n=1}^a$ be a partition of the optimal solution \mathcal{N}^* , where $|\mathcal{U}_n^*| \leq \rho$. By definition, we have:

$$\phi\left(\bigcup_{n'=1}^{n-1}\mathcal{U}_{n'}^G\cup\mathcal{U}_k^*\right)-\phi\left(\bigcup_{n'=1}^{n-1}\mathcal{U}_{n'}^G\right)\leq\eta_n\qquad\forall k\in\{1,\ldots,a\},n\in\{1,\ldots,a\}.$$

Consequently, by Proposition 4.2.3 a valid inequality for each $n \in \{1, \ldots, a\}$ is:

$$\theta^* \leq \phi\left(\bigcup_{n'=1}^{n-1} \mathcal{U}_{n'}^G\right) + \sum_{k=1}^a \left(\phi\left(\bigcup_{n'=1}^{n-1} \mathcal{U}_{n'}^G \cup \mathcal{U}_k^*\right) - \phi\left(\bigcup_{n'=1}^{n-1} \mathcal{U}_{n'}^G\right)\right)$$
$$\leq \phi(\emptyset) + \sum_{n'=1}^{n-1} \eta_{n'} + a\eta_n.$$
(4.3.4)

Thus, the infimum in (4.3.3) can be bounded from below by the optimal objective value of the following relaxation:

Minimise

$$\sum_{n=1}^{a-1} \eta_n + \left(\frac{\rho - b}{\rho}\right) \eta_a$$
subject to

$$\sum_{n'=1}^{n-1} \eta_{n'} + a\eta_n \ge 1 \qquad \forall n \in \{1, \dots, a\}.$$

The objective is equal to $\theta' - \phi(\emptyset)$. The constraints are derived from Equation (4.3.4) and the fact we have used the normalisation $\theta^*(\phi) - \phi(\emptyset) = 1$. As stated in Nemhauser, Wolsey, and Fisher (1978), the following solution is primal-dual feasible:

$$\eta_n = \frac{1}{a} \left(\frac{a-1}{a}\right)^{n-1} \qquad \forall n \in \{1, \dots, a\},$$
$$u_n = \frac{1}{a} \left(\frac{a-\frac{1}{\rho}(\rho-b)}{a-1}\right) \left(\frac{a-1}{a}\right)^{a-n} \qquad \forall n \in \{1, \dots, a-1\},$$
$$u_a = \frac{1}{a} \left(\frac{\rho-b}{\rho}\right),$$

where the dual is given by:

 \mathbf{S}

Maximise
$$\sum_{n=1}^{a} u_n$$
subject to

$$au_n + \sum_{n'=n+1}^{a} u_{n'} = 1 \qquad \forall n \in \{1, \dots, a-1\},$$
$$au_a = \frac{1}{\rho}(\rho - b),$$
$$u_n \ge 0 \qquad \forall n \in \{1, \dots, a\}.$$

The resulting primal and dual objective values are equal to

$$1 - \left(\frac{a - \frac{1}{\rho}(\rho - b)}{a}\right) \left(\frac{a - 1}{a}\right)^{a - 1}$$

Hence, by Strong Duality we have solved the relaxation and found the bound

$$\frac{\theta^* - \theta^G}{\theta^* - \phi(\emptyset)} \le \left(\frac{a - \frac{1}{\rho}(\rho - b)}{a}\right) \left(\frac{a - 1}{a}\right)^{a - 1}.$$

Nemhauser, Wolsey, and Fisher (1978) provide examples for the Uncapacitated Location problem³ to show that the bound is tight for b = 0.

Note that if we apply 1-Greedy Search ($\rho = 1$), Theorem 4.3.3 states that

$$\frac{\theta^*-\theta^G}{\theta^*-\phi(\emptyset)} \leq \left(\frac{p-1}{p}\right)^p = \left(1-\frac{1}{p}\right)^p \leq \frac{1}{e}.$$

Thus, the 1-Greedy Search attains the performance bound stated in Theorem 4.2.4, implying that the 1-Greedy Search is the best possible heuristic in this sense. Of course, other heuristics can still outperform the Greedy Search on average, as the guarantee holds for the worst-case instances.

4.3.2 Greedy Approximation for the MCLP

The family of Uncapacitated Location problems given in Nemhauser, Wolsey, and Fisher (1978) show that the bound in Theorem 4.3.3 is tight for $p = a\rho$ (i.e., b = 0). This family of instances can be modified into a similar family of MCLP instances. As a result, the tightness of the bound for this particular case also holds for the MCLP, see Corollary 4.3.4.

Corollary 4.3.4. Consider an arbitrary MCLP instance $\omega \in \Omega$ and suppose that the ρ -Greedy Search is applied to this problem, where $p = a\rho$ with $\rho, a \in \mathbb{N}_{\geq 1}$ (see Algorithm 4.3.1). The following bound holds and is tight for the ρ -Greedy maximum $\theta^G(\omega)$ and the global maximum $\theta^*(\omega)$:

$$\frac{\theta^*(\omega) - \theta^G(\omega)}{\theta^*(\omega)} \le \left(\frac{a-1}{a}\right)^a.$$

Consequently, the 1-Greedy Search is a $(1-e^{-1})$ -approximation for the MCLP and the guarantee is tight. For fixed $\rho \in \mathbb{N}_{\geq 2}$ the ρ -Greedy Search is at best a $(1-e^{-1})$ -approximation.

Proof. We modify the Uncapacitated Location problem⁴ instances given in Nemhauser, Wolsey, and Fisher (1978) (or in Cornuéjols et al. (1977)) to MCLP instances in the following way. First, we consider worst-case instances for the 1-Greedy Search. These instances also form the basis for worst-case instances for the ρ -Greedy Search.

Let $\rho = 1$ and $p \in \mathbb{N}_{\geq 2}$ (for p = 1 the 1-Greedy Search is optimal). We construct an MCLP instance with $|\mathcal{I}| = 2p - 1$ bases and $|\mathcal{J}| = 2p(p-1)$ demand points. Label the bases as $\mathcal{I} = \{1, \ldots, 2p-1\}$ and the points as $\mathcal{J} = \{1, \ldots, 2p(p-1)\}$. We choose the MCLP parameters such that $\{1, \ldots, p\} \subset \mathcal{I}$ will be the 1-Greedy Search solution and $\{p, \ldots, 2p-1\} \subset \mathcal{I}$ the optimal solution.

³See also Section 4.3.2.

⁴See also Remark 4.3.5.

Define the adjacency parameters as follows: for $i \in \mathcal{I}$ and $j \in \mathcal{J}$ set

$$a_{ij}^{(p)} = \begin{cases} 1 & \text{if } 1 \le i \le p-1, j \text{ is odd and } 2p(i-1) + 1 \le j \le 2pi \\ 1 & \text{if } p \le i \le 2p-1 \text{ and } 2(i-p) \le ((j-1) \mod 2p) \le 2(i-p) + 1 \\ 0 & \text{otherwise} \end{cases}$$

For $\epsilon > 0$ insignificantly small, the demand of point $j \in \mathcal{J}$ is set to:

$$d_{j} = \begin{cases} (p-1)p^{p-2} \left(\frac{p-1}{p}\right)^{k-1} + \epsilon & \text{if } j \text{ is odd} \\ p^{p-1} - (p-1)p^{p-2} \left(\frac{p-1}{p}\right)^{k-1} & \text{if } j \text{ is even} \end{cases},$$

where $k = \left\lceil \frac{j}{2p} \right\rceil$. By definition, the covered demand by base $i \in \mathcal{I}$ is given by

$$\sum_{j \in \mathcal{J}} a_{ij} d_j = \begin{cases} (p-1)p^{p-1} \left(\frac{p-1}{p}\right)^{i-1} + p\epsilon & \text{if } 1 \le i \le p-1\\ (p-1)p^{p-1} + (p-1)\epsilon & \text{if } p \le i \le 2p-1 \end{cases}$$

For example, see Figure 4.3.1 for two worst-case MCLP instances for the 1-Greedy Search.





(b) Example for p = 3.

Figure 4.3.1: Worst-case examples for 1-Greedy Search. Bases are depicted as squares, demand points as circles. Points are connected to adjacent bases and their demand is shown in brackets.

As bases $\{p, \ldots, 2p-1\} \subset \mathcal{I}$ have the largest covered demand and have disjoint coverage, these bases form the optimal solution. Suppose the 1-Greedy Search opens bases $\{1, \ldots, p\} \subset \mathcal{I}$. The resulting objective values are:

$$\begin{aligned} \theta^* &= (p-1)p^p + p(p-1)\epsilon, \\ \theta^G &= \sum_{k=1}^{p-1} (p-1)p^{p-1} \left(\frac{p-1}{p}\right)^{k-1} + \sum_{k=1}^{p-1} \left(p^{p-1} - (p-1)p^{p-2} \left(\frac{p-1}{p}\right)^{k-1}\right) + p(p-1)\epsilon \\ &= (p-1)p^p \left(1 - \left(\frac{p-1}{p}\right)^p\right) + p(p-1)\epsilon. \end{aligned}$$

Therefore, the optimality gap is

$$\frac{\theta^* - \theta^G}{\theta^*} = \frac{(p-1)p^p \left(\frac{p-1}{p}\right)^p}{(p-1)p^p + p(p-1)\epsilon} \le \left(\frac{p-1}{p}\right)^p.$$

Note that for small $\epsilon > 0$ the gap can be arbitrarily close to the bound of Theorem 4.3.3. This bound is equal to e^{-1} in the limit as $p \to \infty$.

It remains to show that the Greedy Search indeed gives $\{1, \ldots, p\} \subset \mathcal{I}$ (the first p bases) as solution. From the covered demand of the bases it is clear that the first base is selected in the first iteration (at least a difference of ϵ with other bases). Suppose the first k bases are opened by the Greedy Search for some $k \in \{1, \ldots, p-2\}$. The net coverage of bases $k + 1, \ldots, p-1$ remains the same, as these bases have disjoint coverage. The net coverage of the last p bases does decrease as the Greedy Search proceeds. At iteration k + 1 the net coverage of a base $i \in \mathcal{I}$ with $i \geq p$ is:

$$(p-1)p^{p-1} - \sum_{k'=1}^{k} (p-1)p^{p-2} \left(\frac{p-1}{p}\right)^{k'-1} + (p-1-k)\epsilon$$
$$= (p-1)p^{p-1} \left(\frac{p-1}{p}\right)^{k} + (p-1-k)\epsilon,$$

which is less than the (net) coverage of base k + 1 (a difference of $(k + 1)\epsilon$). Hence, the Greedy Search selects base k + 1 at iteration k + 1. Finally, at iteration p only identical bases remain and we can assume without loss of generality that base p is chosen.

To conclude, for $\rho = 1$ we have given a family of MCLP instances that has an optimality gap arbitrarily close to the given bound in Corollary 4.3.4. For $\rho \in \mathbb{N}_{\geq 2}$ we can construct similar instances with $p = a\rho$ for some $a \in \mathbb{N}_{\geq 1}$ as follows. Consider the constructed instances for 1-Greedy with a number of opened bases. Copy the smaller 1-Greedy instance ρ times such that the resulting MCLP instance has ρ disjoint 1-Greedy regions. It is trivial to see that the ρ -Greedy Search selects the k-th base of each region at iteration k, resulting in the same relative optimality gap as before.

It is open for research whether similar worst-case MCLP instances with $p = a\rho - b$ for b > 0 $(b \in \{1, ..., \rho - 1\})$ can be constructed. However, for a given ρ -Greedy Search, these worst-case instances with $p = a\rho$ for $a \in \mathbb{N}_{\geq 1}$ show that the ρ -Greedy Search has at best a guarantee of $(1 - e^{-1})$. A straightforward implementation requires the evaluation of $\mathcal{O}(\binom{|\mathcal{I}|}{\rho}) = \mathcal{O}(\frac{1}{\rho}|\mathcal{I}|^{\rho})$ subsets of bases, each taking at most $\mathcal{O}(\rho|\mathcal{J}|)$ time. As $\mathcal{O}(\frac{p}{\rho})$ iterations are needed, the computational time of the search is $\mathcal{O}(\frac{p}{\rho}|\mathcal{I}|^{\rho}|\mathcal{J}|)$, so polynomial in the input size (since $p \leq |\mathcal{I}|$). Note that the submodularity of the objective can be used to implement a lazy evaluation (updating the value of a subset only when required), thus improving the (average) computational time in practice. \Box

Remark 4.3.5. The transformation of Uncapacitated Location problem instances to MCLP instances implicitly used in the proof of Corollary 4.3.4 can be generalised to any Uncapacitated Location problem. In the Uncapacitated Location problem $p \in \mathbb{N}$ bases are opened and each demand point is assigned to an open base. Such an assignment of point $j \in \mathcal{J}$ to base $i \in \mathcal{I}$ increases the objective value by weight $c_{ij} \in \mathbb{R}_{\geq 0}$. The objective is to maximise these gains:

$$\max\left\{\sum_{j\in\mathcal{J}}\max\left\{c_{ij}:i\in\mathcal{U}\right\}:\mathcal{U}\subseteq\mathcal{I},|\mathcal{U}|\leq p\right\}.$$

Clearly, the MCLP is a special case of the Uncapacitated Location problem: take $c_{ij} = a_{ij}d_j$ for all $i \in \mathcal{I}$ and $j \in \mathcal{J}$. To transform an Uncapacitated Location problem into an MCLP we split each demand point $j \in \mathcal{J}$ into $|\mathcal{I}|$ new points as follows. Let $j \in \mathcal{J}$ and temporarily order $i \in \mathcal{I}$ such that the weights c_{ij} are non-decreasing:

$$c_{ij} \le c_{i'j} \qquad \qquad \forall i, i' \in \mathcal{I}, i \le i'.$$

The demand of the k-th new point (obtained by splitting point j) is set to

$$d_j^k = c_{kj} - c_{(k-1)j} \ge 0,$$

where we define $c_{0j} = 0$. Each base $i \in \mathcal{I}$ covers the first new point up to (and including) the *i*-th new point, resulting in a covered demand of c_{ij} of these points. See Figure 4.3.2 for an example. Note that points with zero demand can be removed to gain efficiency. This process is repeated for all original demand points $j \in \mathcal{J}$ and an equivalent MCLP instance is obtained.



(a) Uncapacitated Location problem instance.

(b) MCLP instance.

Figure 4.3.2: Example of an Uncapacitated Location problem instance transformation to an MCLP instance. Bases are depicted as squares, demand points as circles. Points are connected to adjacent bases and their objective coefficients are shown in brackets.

4.3.3 Reversing the Greedy Search

The Greedy Search builds a feasible solution by iteratively opening bases, such that coverage is maximised at each iteration. We can reverse this construction by starting with all bases opened and iteratively closing bases, again such that coverage is maximised at each iteration. That is, we close the bases that would lead to the least decrease in coverage. We call this method the Reverse Greedy Search, see Algorithm 4.3.2.

Algorithm 4.3.2 Reverse Greedy Search for Non-Decreasing Submodular Functions

Input: non-decreasing submodular function $\phi : 2^{\mathcal{N}} \to \mathbb{R}$ and parameters $p, \rho \in \mathbb{N}$ with $p \leq |\mathcal{N}|$ **Output:** Reverse ρ -Greedy maximum solution $\mathcal{N}^{RG} \subseteq \mathcal{N}$

1: procedure REVERSE ρ -GREEDY SEARCH(input) 2: set: $\mathcal{N}^{RG} = \mathcal{N}$ 3: set: $\tau = \min\{\rho, |\mathcal{N}^{RG}| - p\}$ 4: while $\tau > 0$ do 5: set: $\mathcal{N}^{RG} = \mathcal{N}^{RG} \setminus \operatorname{argmax}\{\phi(\mathcal{N}^{RG} \setminus \mathcal{N}^+) : \mathcal{N}^+ \subseteq \mathcal{N}^{RG}, |\mathcal{N}^+| = \tau\}$ 6: set: $\tau = \min\{\rho, |\mathcal{N}^{RG}| - p\}$ 7: end while 8: return \mathcal{N}^{RG} 9: end procedure

Note that the Reverse Greedy Search requires many iterations if $|\mathcal{N}|$ is large and p is small. In this case, the Greedy Search would most likely be preferred. One could reason that the Reverse Greedy Search would be preferred over the Greedy Search if more than half of the bases are to be opened $(p > \frac{1}{2}|\mathcal{I}|)$. However, if we consider worst-case performance for general $p \in \mathbb{N}$, this is not true. The worst-case performance of the Reverse Greedy Search is the worst possible: there exists a family of MCLP instances such that the optimality gap approaches infinity. This result is formulated in Theorem 4.3.6.

Theorem 4.3.6. Let $\phi : 2^{\mathcal{N}} \to \mathbb{R}$ be a non-decreasing submodular function and $p \in \mathbb{N}_{\geq 1}$. Consider the optimisation problem

$$\max\left\{\phi(\mathcal{U}): |\mathcal{U}| \le p, \mathcal{U} \subseteq \mathcal{N}\right\}.$$

The Reverse ρ -Greedy Search with $\rho \in \mathbb{N}_{\geq 1}$ is applied to this problem, see Algorithm 4.3.2. The following (trivial) bound holds and is tight for the Reverse ρ -Greedy maximum θ^{RG} and the global maximum θ^* :

$$\frac{\theta^* - \theta^{RG}}{\theta^* - \phi(\emptyset)} \le 1.$$

Therefore, the optimality gap can be arbitrarily large.

Proof. We construct a family of MCLP instances such that the relative optimality gap approaches one. As the MCLP is a special case of the considered Maximum *p*-Coverage problem, the result holds for non-decreasing submodular functions in general.

First, consider the case that $p = \rho = 1$ and take $c \in \mathbb{N}_{\geq 2}$ arbitrarily. The set of bases is $\mathcal{I} = \{0, 1, \ldots, c\}$ and there are almost twice as many points, $\mathcal{J} = \{1, \ldots, 2c\}$. The first base, base 0, covers points $\{1, \ldots, c\} \subseteq \mathcal{J}$. Base $i \in \mathcal{I} \setminus \{0\}$ covers points $\{i, i + c\} \subseteq \mathcal{J}$. Thus, the bases $\mathcal{I} \setminus \{0\}$ cover two points each and their coverage does not overlap with each other.

For $\epsilon \in (0, 1)$ the demand of point $j \in \mathcal{J}$ is defined as

$$d_j = \begin{cases} 1 & \text{if } j \in \{1, \dots, c\} \\ \epsilon & \text{otherwise} \end{cases}.$$

See Figure 4.3.3 for an example with c = 4.

The Reverse Greedy Search will close base 0 in the first iteration, as no coverage is lost. Closing any other base would lead to a loss of ϵ . From the second iteration onwards, the instance only consists of bases and points that are disjoint and equal. Without loss of generality, we can conclude that the resulting solution is $\mathcal{I}^{RG} = \{1\}$ with a coverage of $\theta^{RG} = 1 + \epsilon$. Clearly, opening base 0 is optimal, $\mathcal{I}^* = \{0\}$ with $\theta^* = c$.

The relative optimality gap is:

$$\frac{\theta^*-\theta^{RG}}{\theta^*}=\frac{c-(1+\epsilon)}{c},$$

which approaches 1 as $c \to \infty$.

For arbitrary $\rho \in \mathbb{N}_{\geq 1}$ and $p \in \mathbb{N}_{\geq 1}$ we copy the above MCLP instance $p\rho$ times, resulting in exactly the same relative optimality gap. Therefore, we have constructed a family of worst-case instances with an arbitrarily large optimality gap.



Figure 4.3.3: Worst-case example for Reverse 1-Greedy Search with p = 1 and c = 4. Bases are depicted as squares, demand points as circles. Points are connected to adjacent bases and their demand is shown in brackets.

4.4 Swap Local Search for the MCLP

Local Search methods try to improve an initial solution by iteratively making adjustments to the considered solution. At each iteration, a specified 'neighbourhood' of the current solution is evaluated, and a neighbouring solution with an improvement in objective value is selected. The selected solution becomes the new solution in the next iteration. The Local Search method terminates if no improvement can be made. Typically, the neighbourhood consists of solutions very similar to the considered solution.

Since there are no costs associated with the bases in the MCLP, it is optimal to open exactly p bases. Consequently, the only sensible Local Search method is to swap bases (closing a base and opening a different base). This method is called Swap Local Search and the corresponding neighbourhood is the Swap neighbourhood. We assume that we have an arbitrary initial feasible solution with p opened bases, for instance obtained with the Greedy Search. The Swap Local Search method terminates if there are no improvements possible in the Swap neighbourhood, that is, if a Swap local maximum has been found. If we limit the number of allowed simultaneous swaps to $\rho \in \mathbb{N}_{>1}$, we denote these local maxima by ρ -Swap local maxima.

The Swap Local Search method for non-decreasing submodular functions is shown in Algorithm 4.4.1. Notice that we do not prescribe in what way improvements are found, i.e., how the neighbourhood is searched. The derived results are valid for any improvement method. Furthermore, the initial solution is arbitrary and all following results hold for any initial feasible solution. For a detailed example of a Swap Local Search implementation, see Appendix B.

Algorithm 4.4.1 Swap Local Search for Non-Decreasing Submodular Functions

Input: non-decreasing submodular function $\phi : 2^{\mathcal{N}} \to \mathbb{R}$, initial feasible solution $\mathcal{N}^0 \subseteq \mathcal{N}$ with $|\mathcal{N}^0| = p$ and parameter $\rho \in \mathbb{N}_{\geq 1}$

Output: ρ -Swap local maximum solution

```
1: procedure \rho-SWAP LOCAL SEARCH(input)
          set: t = 0
 2:
 3:
          loop
               try to find: \mathcal{N}^{t+1} \subseteq \mathcal{N} such that |\mathcal{N}^{t+1}| = p, |\mathcal{N}^{t+1} \setminus \mathcal{N}^t| \leq \rho and \phi(\mathcal{N}^{t+1}) > \phi(\mathcal{N}^t)
 4:
               if such \mathcal{N}^{t+1} is found (successful improvement) then
 5:
                    set: t = t + 1
 6:
 7:
               else
                    return \mathcal{N}^t
 8:
               end if
 9:
          end loop
10:
11: end procedure
```

Let Ω be the set of all MCLP instances. For each MCLP instance $\omega \in \Omega$ we have a set of ρ -Swap local maximum solutions $\mathcal{L}(\omega)$ with objective values $\{\theta^L(\omega) : L \in \mathcal{L}(\omega)\}$, and the global maximum $\theta^*(\omega)$. Similar to bounds on optimality gaps, we can analyse the worst possible gap when considering all MCLP instances and all corresponding local maxima. This (tight) bound on the gap is called the relative locality gap $\alpha \in [0, 1]$ and is defined in Definition 4.4.1.

Definition 4.4.1. Consider a maximisation problem and let Ω be the set of all instances. Each instance $\omega \in \Omega$ has a set of ρ -Swap local maxima $\{\theta^L(\omega) : L \in \mathcal{L}(\omega)\}$ and global maximum $\theta^*(\omega)$. The relative locality gap $\alpha \in [0, 1]$ is defined as:

$$\sup_{\omega\in\Omega, L\in\mathcal{L}(\omega)} \left\{ \frac{\theta^*(\omega) - \theta^L(\omega)}{\theta^*(\omega)} \right\} = (1 - \alpha).$$

Consequently, for relative locality gap $\alpha \in [0, 1]$ it holds that

$$\frac{\theta^*(\omega) - \theta^L(\omega)}{\theta^*(\omega)} \le (1 - \alpha) \qquad \forall L \in \mathcal{L}(\omega), \omega \in \Omega,$$

or equivalently

$$\theta^L(\omega) \ge \alpha \theta^*(\omega) \qquad \qquad \forall L \in \mathcal{L}(\omega), \omega \in \Omega.$$

Note that a higher value of $\alpha \in [0, 1]$ implies a better guaranteed performance for the Swap Local Search. In contrast to bounds on optimality gaps, the relative locality gap is tight by definition. Furthermore, it holds for all local maxima, that is, for all initial solutions of the Swap Local Search.

Performance guarantees (bounds on the relative locality gap) for the Swap Local Search method for submodular functions are known. We give these results in Section 4.4.1. As the MCLP is a special case of the maximisation of a submodular function, it could be that these guarantees can be improved for the MCLP. We prove in Section 4.4.2 that this is the case: we derive a tight bound for the MCLP. We note that the results in Section 4.4.2 have been obtained independently from Nemhauser, Wolsey, and Fisher (1978).

4.4.1 Swap Guarantees for Submodular Functions

In Theorem 4.4.2 we state a general bound for the relative locality gap for the Swap Local Search method for non-decreasing submodular functions. This is a result by Nemhauser, Wolsey, and Fisher (1978). The proof is based on Proposition 4.2.3, where the partition is chosen such that we can use the properties of a Swap local maximum. It will become apparent that its proof is significantly different from our proof in Section 4.4.2 (which only considers the MCLP).

Theorem 4.4.2 (Nemhauser, Wolsey, and Fisher (1978)). Let $\phi : 2^{\mathcal{N}} \to \mathbb{R}$ be a non-decreasing submodular function and $p \in \mathbb{N}$. Consider the optimisation problem

$$\max\left\{\phi(\mathcal{U}): |\mathcal{U}| \le p, \mathcal{U} \subseteq \mathcal{N}\right\}.$$

The ρ -Swap Local Search is applied to this problem with an arbitrary initial feasible solution, where $p = a\rho - b$ with $\rho, a \in \mathbb{N}_{\geq 1}$ and $b \in \{0, \ldots, \rho - 1\}$ (see Algorithm 4.4.1). The following bound holds for any ρ -Swap local maximum θ^L and the global maximum θ^* :

$$\frac{\theta^* - \theta^L}{\theta^* - \phi(\emptyset)} \le \frac{p - \rho + b}{2p - \rho + b}.$$

If b = 0, then this bound is tight.

Proof. Let $\mathcal{N}^L \subseteq \mathcal{N}$ be the solution of the ρ -Swap Local Search method. Partition⁵ \mathcal{N}^L into disjoint sets, $\{\mathcal{U}_n^L\}_{n=1}^a$, where the first set \mathcal{U}_1^L has cardinality $(\rho - b)$ and the other (a - 1) sets have cardinality ρ . The partition is done using appropriate Greedy Search iterations on \mathcal{N}^L , starting with a $(\rho - b)$ -Greedy iteration and followed by ρ -Greedy iterations. Hence,

$$\delta_2 \ge \ldots \ge \delta_a \ge 0,$$

where δ_n is the marginal increase in objective defined by:

$$\delta_n = \phi\left(\bigcup_{n'=1}^n \mathcal{U}_{n'}^L\right) - \phi\left(\bigcup_{n'=1}^{n-1} \mathcal{U}_{n'}^L\right) \qquad \forall n \in \{1, \dots, a\}$$

For the first set (n = 1) we can use Proposition 4.3.2 with the submodularity of ϕ to note that

$$\delta_1 \ge \left(\frac{\rho-b}{\rho}\right)\delta_a.$$

By definition we have $\theta^L = \phi(\emptyset) + \sum_{n=1}^a \delta_n$ and the following bound $\beta \in \mathbb{R}_{\geq 0}$:

$$\theta^{L} - \phi(\emptyset) = \sum_{n=1}^{a} \delta_{n} \ge \left(\frac{\rho - b}{\rho} + (a - 1)\right) \delta_{a} = \left(\frac{p}{\rho}\right) \delta_{a} \equiv \beta \delta_{a}.$$
(4.4.1)

Let $\mathcal{N}^* \subseteq \mathcal{N}$ be the optimal solution and partition \mathcal{N}^* into disjoint sets, $\{\mathcal{U}_n^*\}_{n=1}^a$, where each set has cardinality at most $|\mathcal{U}_a^L| = \rho$. From Proposition 4.2.3 we have:

$$\theta^* = \phi\left(\mathcal{N}^*\right) \le \phi\left(\bigcup_{n'=1}^{a-1} \mathcal{U}_{n'}^L\right) + \sum_{n=1}^a \left(\phi\left(\bigcup_{n'=1}^{a-1} \mathcal{U}_{n'}^L \cup \mathcal{U}_n^*\right) - \phi\left(\bigcup_{n'=1}^{a-1} \mathcal{U}_{n'}^L\right)\right).$$
(4.4.2)

However, since \mathcal{N}^L is a ρ -Swap local optimum, it holds that for all $n \in \{1, \ldots, a\}$

$$\delta_{a} = \phi \left(\bigcup_{n'=1}^{a} \mathcal{U}_{n'}^{L} \right) - \phi \left(\bigcup_{n'=1}^{a-1} \mathcal{U}_{n'}^{L} \right) = \phi \left(\bigcup_{n'=1}^{a-1} \mathcal{U}_{n'}^{L} \cup \mathcal{U}_{a}^{L} \right) - \phi \left(\bigcup_{n'=1}^{a-1} \mathcal{U}_{n'}^{L} \right)$$
$$\geq \phi \left(\bigcup_{n'=1}^{a-1} \mathcal{U}_{n'}^{L} \cup \mathcal{U}_{a}^{*} \right) - \phi \left(\bigcup_{n'=1}^{a-1} \mathcal{U}_{n'}^{L} \right).$$
(4.4.3)

Thus, combining Equations (4.4.1), (4.4.2), and (4.4.3) yields

$$\theta^* \le \phi \left(\bigcup_{n'=1}^{a-1} \mathcal{U}_{n'}^L \right) + a\delta_a = \theta^L + (a-1)\delta_a \le \theta^L + \frac{(a-1)}{\beta} \left(\theta^L - \phi(\emptyset) \right),$$

or equivalently,

$$\left(\beta + (a-1)\right)\left(\theta^* - \theta^L\right) \le (a-1)\left(\theta^* - \phi(\emptyset)\right).$$

We get the desired bound with the substitution $a = (p+b)/\rho$ and the definition of β :

$$\frac{\theta^* - \theta^L}{\theta^* - \phi(\emptyset)} \le \frac{(a-1)}{\beta + (a-1)} = \frac{p - \rho + b}{2p - \rho + b}.$$

In Nemhauser, Wolsey, and Fisher (1978) an Uncapacitated Location problem example is given to show that the bound is tight for b = 0.

 $^{^5 \}mathrm{See}$ also Remark 4.4.3.

We will show in Theorem 4.4.4 that for the MCLP the bound for b = 0 holds for all $\rho \in \mathbb{N}_{\geq 1}$, and thus also for b > 0. Furthermore, the bound is always tight. This improves the general bound given in Theorem 4.4.2. Such an improvement is already noted in Nemhauser, Wolsey, and Fisher (1978), but mentioned in the context of the Uncapacitated Location problem.

Remark 4.4.3. Notice that if we partition \mathcal{N}^L and \mathcal{N}^* into $a^L, a^* \in \mathbb{N}$ subsets respectively, the proof still holds under certain conditions. To be able to use the ρ -Swap local maximum conditions in Equation (4.4.3), it must hold that $|\mathcal{U}_{aL}^L| \leq \rho$ and $|\mathcal{U}_n^*| \leq |\mathcal{U}_{aL}^L|$ for all $n \in \{1, \ldots, a^*\}$. Since $|\mathcal{N}^*| = p$, we have $a^* \geq p/|\mathcal{U}_{aL}^L|$. Repeated use of Proposition 4.3.2 as in Equation (4.4.1) gives

$$\theta^L - \phi(\emptyset) = \sum_{n=1}^{a^L} \delta_n \ge \sum_{\{n: |\mathcal{U}_n^L| > |\mathcal{U}_{aL}^L|\}} \delta_{a^L} + \sum_{\{n: |\mathcal{U}_n^L| \le |\mathcal{U}_{aL}^L|\}} \frac{|\mathcal{U}_n^L|}{|\mathcal{U}_{a^L}^L|} \delta_{a^L} \equiv \beta \delta_{a^L}.$$

For fixed $|\mathcal{U}_{a^L}^L| \leq \rho$, the partitions of \mathcal{N}^L and \mathcal{N}^* that give the best bound are similar to those used in the proof. This results in $\beta = p/|\mathcal{U}_{a^L}^L|$ and $a^* = \lceil p/|\mathcal{U}_{a^L}^L| \rceil$. Consequently, we have

$$\frac{\theta^* - \theta^L}{\theta^* - \phi(\emptyset)} \le \frac{a^* - 1}{\beta + a^* - 1} = \frac{\lceil p / |\mathcal{U}_{a^L}^L| \rceil - 1}{p / |\mathcal{U}_{a^L}^L| + \lceil p / |\mathcal{U}_{a^L}^L| \rceil - 1}$$

where the bound is not necessarily minimal if $|\mathcal{U}_{a^L}^L| = \rho$. If b = 0, then $|\mathcal{U}_{a^L}^L| = \rho$ is optimal.

4.4.2 Swap Locality Gap for the MCLP

The above results for submodular functions give bounds on the locality gap for the MCLP. In Theorem 4.4.4 we show that the ρ -Swap Local Search method for MCLP has a locality gap of $\alpha = 1/2$. Its proof is based on constructing a new MCLP instance with at least equally large relative optimality gap as the original MCLP instance, but with a simplified internal structure. This process of simplification is repeated until a family of (worst-case) instances remains for which the locality gap is trivial.

Theorem 4.4.4. Consider an arbitrary MCLP instance $\omega \in \Omega$ with global maximum $\theta^*(\omega)$ and optimal solution $x^*(\omega)$. For an arbitrary initial feasible solution, let $x^L(\omega)$ be the solution found by the ρ -Swap Local Search method ($\rho \in \mathbb{N}_{\geq 1}$) with ρ -Swap local maximum $\theta^L(\omega)$, see Algorithm 4.4.1.

Suppose $x^*(\omega)$ and $x^L(\omega)$ differ in exactly $2k(\omega)$ places for some $k(\omega) \in \mathbb{N}_{>\rho}$, that is,

$$\sum_{i \in \mathcal{I}} \left| x_i^*(\omega) - x_i^L(\omega) \right| = 2k(\omega).$$

Then we can bound the relative optimality gap by

$$\frac{\theta^*(\omega) - \theta^L(\omega)}{\theta^*(\omega)} \le \frac{k(\omega) - \rho}{2k(\omega) - \rho},$$

and this bound is tight. In particular, the relative locality gap of ρ -Swap Local Search for the MCLP is $\alpha = 1/2$:

$$\sup_{\omega \in \Omega, L \in \mathcal{L}(\omega)} \left\{ \frac{\theta^*(\omega) - \theta^L(\omega)}{\theta^*(\omega)} \right\} = \frac{1}{2}.$$

We split the proof of Theorem 4.4.4 over several lemmas. For the proof we fix $\rho \in \mathbb{N}_{\geq 1}$ and $k \in \mathbb{N}$ to arbitrary values. Consider an arbitrary MCLP instance $\omega \in \Omega$ where $x^*(\omega)$ and $x^L(\omega)$ differ in exactly 2k places (k swaps). Note that k = 0 implies that the local and global maxima are equal. If $1 \leq k \leq \rho$ the global optimum is in the ρ -Swap neighbourhood and we have a contradiction regarding the ρ -Swap local optimality of $x^L(\omega)$. Thus, assume $k \geq \rho + 1$.

The proof sequentially transforms ω into new instances ω' such that the relative optimality gap of ω' is as least as large as that of ω . Each transformation preserves the following properties:

- $x^*(\omega)$ is an optimal solution for ω' (albeit projected to a lower dimension),
- $x^{L}(\omega)$ is a ρ -Swap local optimum for ω' (albeit projected to a lower dimension),
- $x^{L}(\omega)$ differs from $x^{*}(\omega)$ in exactly 2k places.

The first transformation is a projection to a lower dimension and is captured by $\phi_1 : \Omega \to \Omega$. We transform the instance ω to a new instance where some of the common coverage between the global and ρ -Swap local maxima is eliminated. We need to introduce some notation to be precise. Let $\mathcal{X}(\omega) = \{i \in \mathcal{I}(\omega) : x_i^*(\omega) \neq x_i^L(\omega)\}$ be the set of bases where $x^*(\omega)$ and $x^L(\omega)$ differ, so $|\mathcal{X}(\omega)| = 2k$. Also, set $\mathcal{I}^*(\omega) = \{i \in \mathcal{I}(\omega) : x_i^*(\omega) = 1\}$ to be the optimal set of opened bases and likewise $\mathcal{I}^L(\omega) = \{i \in \mathcal{I}(\omega) : x_i^L(\omega) = 1\}$.

Construct a new MCLP instance $\omega' = \phi_1(\omega)$ where:

- the demand points not covered by any base in $\mathcal{I}^*(\omega) \cup \mathcal{I}^L(\omega)$ are removed,
- the demand points covered by any base in $\mathcal{I}^*(\omega) \cap \mathcal{I}^L(\omega)$ are removed,
- the bases in $\mathcal{I}(\omega) \setminus \mathcal{X}(\omega)$ are removed (i.e., $\mathcal{I}(\omega') = \mathcal{X}(\omega)$).
- the number of opened bases is reduced to $p(\omega') = p(\omega) |\mathcal{I}^*(\omega) \cap \mathcal{I}^L(\omega)|.$

No other changes are performed. For an example of this transformation, see Figure 4.4.1.

The solution $x_i^*(\omega') = x_i^*(\omega)$ for $i \in \mathcal{I}(\omega')$ is optimal for ω' and $x_i^L(\omega') = x_i^L(\omega)$ for $i \in \mathcal{I}(\omega')$ is a ρ -Swap local optimum. Suppose this is not the case, then the required swaps to obtain the (local) optimum could also be applied to the corresponding solution of the original instance ω , with exactly the same difference in objective. This contradicts the (local) optimality of the original solution. As in the original instance, $x^*(\omega')$ and $x^L(\omega')$ differ in exactly 2k places.

For the new instance, it holds that $\theta^*(\omega') = \theta^*(\omega) - c(\omega)$ and $\theta^L(\omega') = \theta^L(\omega) - c(\omega)$, where $c(\omega) \in \mathbb{R}_{>0}$ is the demand covered by bases opened in both solutions of ω :

$$c(\omega) = \sum_{j \in \mathcal{J}} d_j \min \left\{ 1, \sum_{i \in \mathcal{I}} a_{ij} x_i^*(\omega) x_i^L(\omega) \right\}.$$

Consequently,

$$\frac{\theta^*(\omega') - \theta^L(\omega')}{\theta^*(\omega')} = \frac{\theta^*(\omega) - \theta^L(\omega)}{\theta^*(\omega')} \ge \frac{\theta^*(\omega) - \theta^L(\omega)}{\theta^*(\omega)},$$

that is, the relative optimality gap of ω' is as least as large as that of ω . Therefore, for worst-case instances we can focus on the bases in $\mathcal{X}(\omega)$.



(a) Original instance $\omega \in \Omega$.



(c) Second intermediate instance: all demand points covered by any base in $\mathcal{I}^*(\omega) \cap \mathcal{I}^L(\omega)$ have been removed.



(b) First intermediate instance: all demand points not covered by any base in the set $\mathcal{I}^*(\omega) \cup \mathcal{I}^L(\omega)$ have been removed.



(d) Final instance $\phi_1(\omega)$: all bases in the set $\mathcal{I}(\omega) \setminus \mathcal{X}(\omega)$ have been removed.

Figure 4.4.1: Example of the first MCLP instance transformation ϕ_1 . Bases are depicted as squares, demand points as black dots. The circles indicate the coverage of the bases. This example assumes that $\mathcal{I}(\omega) = \{A, B, C, D, E, F\}$, $p(\omega) = 3$, $k(\omega) = 2$, $\mathcal{I}^L(\omega) = \{A, B, F\}$, and $\mathcal{I}^*(\omega) = \{C, D, F\}$. Hence, we have $\mathcal{X}(\omega) = \{A, B, C, D\}$.

Consider a new arbitrary MCLP instance $\omega \in \phi_1(\Omega)$ where $x^*(\omega)$ and $x^L(\omega)$ differ in exactly 2k places. Note that $|\mathcal{I}(\omega)| = 2k$, so the two solutions have no matching elements. For the remainder of the proof, we fix the set of bases, $\mathcal{I}(\omega) = \mathcal{I}$, as well as both solutions, $x^*(\omega) = x^*$ and $x^L(\omega) = x^L$.

Each demand point can be covered by a certain subset of bases in \mathcal{I} . Thus, we can project each point to the largest subset of bases that can cover it:

$$\pi: \mathcal{J}(\omega) \to 2^{\mathcal{I}}.$$

This projection is not guaranteed an injection nor a surjection (it depends on the instance). However, we can define an inverse mapping:

$$\pi^{-1}: 2^{\mathcal{I}} \to 2^{\mathcal{J}(\omega)},$$

such that $j \in \pi^{-1}(\pi(j))$ for all $j \in \mathcal{J}(\omega)$. Elements in $2^{\mathcal{I}} \setminus \pi(\mathcal{J}(\omega))$ are mapped to the empty set. For example, the demand points in $\pi^{-1}(\{i\})$ are the points that can only be covered by base $i \in \mathcal{I}$. Demand points in $\pi^{-1}(\{i_1, i_2\})$ can be covered by exactly two bases $(i_1 \text{ and } i_2)$, etcetera. We can divide the demand points into several sets, where two points $j_1, j_2 \in \mathcal{J}(\omega)$ are in the same set if and only if both are covered by the same bases: $\pi(j_1) = \pi(j_2)$. Thus, the resulting sets are exactly the image of π^{-1} .

The second instance transformation will merge certain demand points and add new artificial demand points. Construct a new MCLP instance by merging the demand points in each set in the image of π^{-1} . Note that π is injective for this new instance. Next, add artificial demand points with zero demand to the new instance in such a way that π becomes bijective. Call this transformation $\phi_2 : \phi_1(\Omega) \to \Omega$, see also Figures 4.4.2 and 4.4.3. It is trivial that ϕ_2 does not affect feasibility and objective values.

We fix this set of demand points for the remainder of the proof, $\mathcal{J}(\omega') = \mathcal{J}$. As a result, the mapping π for $\omega' = \phi_2(\omega)$ is a bijection with inverse

$$\pi^{-1}: 2^{\mathcal{I}} \to \mathcal{J},$$

see also Lemma 4.4.5.

Lemma 4.4.5. For each MCLP instance $\omega \in \Omega$ there exists an MCLP instance $\omega' \in \Omega$ with the bijective mapping $\pi : \mathcal{J}(\omega') \to 2^{\mathcal{I}(\omega')}$, that maps each demand point to the largest subset of bases that can cover it.

For the new MCLP instance ω' , $x^*(\omega)$ is still the global optimum and $x^L(\omega)$ still a ρ -Swap local optimum, although a projection to a lower dimension can be required. Furthermore, it holds that

$$\frac{\theta^*(\omega') - \theta^L(\omega')}{\theta^*(\omega')} \ge \frac{\theta^*(\omega) - \theta^L(\omega)}{\theta^*(\omega)}$$

Therefore, the relative optimality gap of ω' is as least as large as that of ω .

Proof. The proof has been given above by taking $\omega' = \phi_2(\phi_1(\omega))$.





(a) Original instance $\omega \in \phi_1(\Omega)$.

(b) New instance $\phi_2(\omega)$: specific demand points are merged. Artificial demand points are not shown.

Figure 4.4.2: Example of the second MCLP instance transformation ϕ_2 . Bases are depicted as squares, demand points as black dots. The circles indicate the coverage of the bases. This example assumes that $\mathcal{I}(\omega) = \{A, B, C, D\}$, $p(\omega) = 2$, $\mathcal{I}^L(\omega) = \{A, B\}$, and $\mathcal{I}^*(\omega) = \{C, D\}$.



Figure 4.4.3: Example of an MCLP instance in $\phi_2(\phi_1(\Omega))$. Bases are depicted as squares, demand points as grey dots. The circles indicate the coverage of the bases. This example assumes that $\mathcal{I}(\omega) = \{A, B, C, D\}, p(\omega) = 2, \mathcal{I}^L(\omega) = \{A, B\}, \text{ and } \mathcal{I}^*(\omega) = \{C, D\}$. For clarity, we abbreviate the demand $d_{\pi^{-1}(\{A\})}(\omega)$ to d_A and similarly for the other subsets.

Let $\omega \in \phi_2(\phi_1(\Omega))$ be a new arbitrary MCLP instance, where x^* and x^L differ in exactly 2k places. Define the sets $\mathcal{I}^* = \{i \in \mathcal{I} : x_i^* = 1\}$ and $\mathcal{I}^L = \{i \in \mathcal{I} : x_i^L = 1\}$. Since the sets \mathcal{I}^* and \mathcal{I}^L of bases are disjoint, we can label the bases distinctively as follows:

$$\mathcal{I}^{L} = \{i_{1}^{L}, \dots, i_{k}^{L}\}$$
 and $\mathcal{I}^{*} = \{i_{1}^{*}, \dots, i_{k}^{*}\}.$

We can explicitly express the global and local maxima by

.

$$\theta^{L}(\omega) = \sum_{n=1}^{k} \sum_{\{\mathcal{Y}^{L} \subseteq \mathcal{I}^{L}: |\mathcal{Y}^{L}|=n\}} \sum_{m=0}^{k} \sum_{\{\mathcal{Y}^{*} \subseteq \mathcal{I}^{*}: |\mathcal{Y}^{*}|=m\}} d_{\pi^{-1}(\mathcal{Y}^{L} \cup \mathcal{Y}^{*})}(\omega), \qquad (4.4.4)$$
$$\theta^{*}(\omega) = \sum_{m=1}^{k} \sum_{\{\mathcal{Y}^{*} \subseteq \mathcal{I}^{*}: |\mathcal{Y}^{*}|=m\}} \sum_{n=0}^{k} \sum_{\{\mathcal{Y}^{L} \subseteq \mathcal{I}^{L}: |\mathcal{Y}^{L}|=n\}} d_{\pi^{-1}(\mathcal{Y}^{L} \cup \mathcal{Y}^{*})}(\omega).$$

The local maximum is equal to all demand from the points that can be covered by at least one base in \mathcal{I}^L . Likewise, for the global maximum at least one base from \mathcal{I}^* is required. For example, consider Figure 4.4.3. Omitting demand terms equal to zero, we have the local maximum

$$\theta^{L}(\omega) = \left(d_{\pi^{-1}(\{A\})}(\omega) + d_{\pi^{-1}(\{B\})}(\omega)\right) + \left(d_{\pi^{-1}(\{AC\})}(\omega) + d_{\pi^{-1}(\{BC\})}(\omega) + d_{\pi^{-1}(\{BD\})}(\omega)\right) + \left(d_{\pi^{-1}(\{AB\})}(\omega)\right) + \left(d_{\pi^{-1}(\{ABC\})}(\omega)\right) + \left(d_{\pi^{-1}(\{ABC\})}(\omega)\right).$$

Here, we have grouped the terms according to the cardinalities n and m as in Equation (4.4.4). Similarly, the global maximum is given by:

$$\theta^{*}(\omega) = \left(d_{\pi^{-1}(\{C\})}(\omega) + d_{\pi^{-1}(\{D\})}(\omega)\right) + \left(d_{\pi^{-1}(\{AC\})}(\omega) + d_{\pi^{-1}(\{BC\})}(\omega) + d_{\pi^{-1}(\{BD\})}(\omega)\right) + \left(d_{\pi^{-1}(\{ABC\})}(\omega)\right) + \left(d_{\pi^{-1}(\{BCD\})}(\omega)\right) + \left(d_{\pi^{-1}(\{BCD\})}(\omega)\right).$$

What remains is the expression of the properties of the ρ -Swap local maximum and the global maximum. Since the global maximum is equivalent to a k-Swap local maximum, we can focus on expressing Swap local maxima as constraints. Consider the ρ -Swap local optimum (x^L) and the swap where $i^L \in \mathcal{I}^L$ is replaced by $i^* \in \mathcal{I}^*$. The following demand is the net loss of this swap:

$$\sum_{m=0}^{k-1} \sum_{\{\mathcal{Y}^* \subseteq \mathcal{I}^* \setminus \{i^*\} : |\mathcal{Y}^*|=m\}} d_{\pi^{-1}(\{i^L\} \cup \mathcal{Y}^*)}(\omega).$$

This expression is equal to the demand of all points covered by base i^L and simultaneously covered by any base or multiple bases in $\mathcal{I}^* \setminus \{i^*\}$. Similarly, the following demand is the net gain:

$$\sum_{m=0}^{k-1} \sum_{\{\mathcal{Y}^* \subseteq \mathcal{I}^* \setminus \{i^*\} : |\mathcal{Y}^*|=m\}} d_{\pi^{-1}(\{i^*\} \cup \mathcal{Y}^*)}(\omega).$$

The net gain is equal to the demand of all points covered by i^* and simultaneously covered by any base or multiple bases in $\mathcal{I}^* \setminus \{i^*\}$. The net effect of each swap must be non-positive, since x^L is in particular a 1-Swap local optimum. Thus, for all $i^* \in \mathcal{I}^*$ and $i^L \in \mathcal{I}^L$:

$$\sum_{m=0}^{k-1} \sum_{\{\mathcal{Y}^* \subseteq \mathcal{I}^* \setminus \{i^*\} : |\mathcal{Y}^*| = m\}} \left(d_{\pi^{-1}(\{i^*\} \cup \mathcal{Y}^*)}(\omega) - d_{\pi^{-1}(\{i^L\} \cup \mathcal{Y}^*)}(\omega) \right) \le 0.$$

In general, consider swapping $R \in \{1, \ldots, \rho\}$ bases in \mathcal{I}^L with bases in \mathcal{I}^* . Assume we swap $\mathcal{U}^L \subseteq \mathcal{I}^L$ with $\mathcal{U}^* \subseteq \mathcal{I}^*$, both with cardinality R. The net loss is

$$\sum_{r=1}^{R} \sum_{\{\mathcal{Z}^{L} \subseteq \mathcal{U}^{L} : |\mathcal{Z}^{L}|=r\}} \sum_{m=0}^{k-R} \sum_{\{\mathcal{Y}^{*} \subseteq \mathcal{I}^{*} \setminus \mathcal{U}^{*} : |\mathcal{Y}^{*}|=m\}} d_{\pi^{-1}(\mathcal{Z}^{L} \cup \mathcal{Y}^{*})}(\omega),$$

and the net gain is

$$\sum_{r=1}^{R} \sum_{\{\mathcal{Z}^* \subseteq \mathcal{U}^* : |\mathcal{Z}^*|=r\}} \sum_{m=0}^{k-R} \sum_{\{\mathcal{Y}^* \subseteq \mathcal{I}^* \setminus \mathcal{U}^* : |\mathcal{Y}^*|=m\}} d_{\pi^{-1}(\mathcal{Z}^* \cup \mathcal{Y}^*)}(\omega).$$

Therefore, the following ρ -Swap optimality constraints must hold for all $R \in \{1, \ldots, \rho\}$, $\mathcal{U}^L \subseteq \mathcal{I}^L$ with $|\mathcal{U}^L| = R$, and $\mathcal{U}^* \subseteq \mathcal{I}^*$ with $|\mathcal{U}^*| = R$:

$$\sum_{r=1}^{R} \sum_{\{\mathcal{Z}^* \subseteq \mathcal{U}^* : |\mathcal{Z}^*|=r\}} \sum_{m=0}^{k-R} \sum_{\{\mathcal{Y}^* \subseteq \mathcal{I}^* \setminus \mathcal{U}^* : |\mathcal{Y}^*|=m\}} d_{\pi^{-1}(\mathcal{Z}^* \cup \mathcal{Y}^*)}(\omega)$$
$$- \sum_{r=1}^{R} \sum_{\{\mathcal{Z}^L \subseteq \mathcal{U}^L : |\mathcal{Z}^L|=r\}} \sum_{m=0}^{k-R} \sum_{\{\mathcal{Y}^* \subseteq \mathcal{I}^* \setminus \mathcal{U}^* : |\mathcal{Y}^*|=m\}} d_{\pi^{-1}(\mathcal{Z}^L \cup \mathcal{Y}^*)}(\omega) \le 0.$$
(4.4.5)

The constraints for the global optimum (x^*) are similar, but the roles of \mathcal{U}^L and \mathcal{U}^* are interchanged. Therefore, the following constraints must hold for all $R \in \{1, \ldots, k\}, \mathcal{U}^L \subseteq \mathcal{I}^L$ with $|\mathcal{U}^L| = R$, and $\mathcal{U}^* \subseteq \mathcal{I}^*$ with $|\mathcal{U}^*| = R$:

$$\sum_{r=1}^{R} \sum_{\{\mathcal{Z}^{L} \subseteq \mathcal{U}^{L}: |\mathcal{Z}^{L}|=r\}} \sum_{m=0}^{k-R} \sum_{\{\mathcal{Y}^{L} \subseteq \mathcal{I}^{L} \setminus \mathcal{U}^{L}: |\mathcal{Y}^{L}|=m\}} d_{\pi^{-1}(\mathcal{Z}^{L} \cup \mathcal{Y}^{L})}(\omega)$$
$$- \sum_{r=1}^{R} \sum_{\{\mathcal{Z}^{*} \subseteq \mathcal{U}^{*}: |\mathcal{Z}^{*}|=r\}} \sum_{m=0}^{k-R} \sum_{\{\mathcal{Y}^{L} \subseteq \mathcal{I}^{L} \setminus \mathcal{U}^{L}: |\mathcal{Y}^{L}|=m\}} d_{\pi^{-1}(\mathcal{Z}^{*} \cup \mathcal{Y}^{L})}(\omega) \leq 0.$$
(4.4.6)

In particular, this constraint must hold for x^* and x^L , i.e., for R = k, $\mathcal{U}^L = \mathcal{I}^L$, and $\mathcal{U}^* = \mathcal{I}^*$. The corresponding constraint can be rewritten to:

$$\theta^{*}(\omega) - \theta^{L}(\omega) = \sum_{m=1}^{k} \sum_{\{\mathcal{Y}^{*} \subseteq \mathcal{I}^{*}: |\mathcal{Y}^{*}| = m\}} d_{\pi^{-1}(\mathcal{Y}^{*})}(\omega) - \sum_{n=1}^{k} \sum_{\{\mathcal{Y}^{L} \subseteq \mathcal{I}^{L}: |\mathcal{Y}^{L}| = n\}} d_{\pi^{-1}(\mathcal{Y}^{L})}(\omega) \ge 0.$$

For example, for the instance in Figure 4.4.3 we have:

$$\theta^*(\omega) - \theta^L(\omega) = \left(d_{\pi^{-1}(\{C\})}(\omega) + d_{\pi^{-1}(\{D\})}(\omega) + d_{\pi^{-1}(\{CD\})}(\omega) \right) - \left(d_{\pi^{-1}(\{A\})}(\omega) + d_{\pi^{-1}(\{B\})}(\omega) + d_{\pi^{-1}(\{AB\})}(\omega) \right),$$

which must be non-negative.

The third instance transformation constructs a more symmetric instance and will be defined below. With this transformation we can prove the following lemma.

Lemma 4.4.6. Consider an MCLP instance $\omega \in \phi_2(\phi_1(\Omega))$ where the global optimum $x^*(\omega)$ and ρ -Swap local optimum $x^L(\omega)$ differ at 2k places. There exists a modified MCLP instance $\omega' \in \phi_2(\phi_1(\Omega))$ with certain symmetric properties, namely that for $n, m \in \{0, \ldots, k\}$

$$d_{\pi^{-1}(\mathcal{Y}^L \cup \mathcal{Y}^*)}(\omega') = d_{(n,m)}(\omega') \qquad \forall \, \mathcal{Y}^L \subseteq \mathcal{I}^L(\omega'), |\mathcal{Y}^L| = n, \mathcal{Y}^* \subseteq \mathcal{I}^*(\omega'), |\mathcal{Y}^*| = m.$$

For the new MCLP instance ω' , $x^*(\omega)$ is still the global optimum and $x^L(\omega)$ still a ρ -Swap local optimum. Furthermore, it holds that

$$\frac{\theta^*(\omega') - \theta^L(\omega')}{\theta^*(\omega')} = \frac{\theta^*(\omega) - \theta^L(\omega)}{\theta^*(\omega)}.$$

Therefore, the relative optimality gap of ω' is equal to that of ω .

Proof. Let $\omega \in \phi_2(\phi_1(\Omega))$ be an arbitrary instance where the global optimum and ρ -Swap local optimum differ at 2k places. Let $\sigma^* : \mathcal{I}^* \to \mathcal{I}^*$ be a permutation of the bases in \mathcal{I}^* and $\sigma^L : \mathcal{I}^L \to \mathcal{I}^L$ a similarly defined permutation. Note that σ^* and σ^L are disjoint. There are k! different permutations for each set, so if we combine the two permutations there are k!k! permuted instances $\sigma(\omega)$. We denote these instances by $\sigma_s(\omega)$ with $s \in \{1, \ldots, k!k!\}$. The solutions x^* and x^L are global and local optima for all permuted instances $\sigma(\omega)$, as only the labels of the bases are changed within each set \mathcal{I}^* and \mathcal{I}^L . See Figure 4.4.4 for an example.

Now construct a new MCLP instance ω' by altering the demand of each point in the following way. For $\mathcal{Y}^L \subseteq \mathcal{I}^L$ and $\mathcal{Y}^* \subseteq \mathcal{I}^*$ define

$$d_{\pi^{-1}(\mathcal{Y}^{L}\cup\mathcal{Y}^{*})}(\omega') = \frac{1}{k!k!} \sum_{s=1}^{k!k!} d_{\pi^{-1}(\mathcal{Y}^{L}\cup\mathcal{Y}^{*})}(\sigma_{s}(\omega)) = \frac{1}{k!k!} \sum_{s=1}^{k!k!} d_{\pi^{-1}(\sigma_{s}(\mathcal{Y}^{L}\cup\mathcal{Y}^{*}))}(\omega).$$

As all constraints (4.4.5) and (4.4.6) are linear in d and valid for each $\sigma_s(\omega)$, these are also valid for the new instance ω' . Furthermore, it holds that $\theta^*(\omega') = \theta^*(\omega)$ and $\theta^L(\omega') = \theta^L(\omega)$, so the relative optimality gap is the same.

Notice that for $n, m \in \{0, \ldots, k\}$, $\mathcal{Y}^L \subseteq \mathcal{I}^L$ with $|\mathcal{Y}^L| = n$, and $\mathcal{Y}^* \subseteq \mathcal{I}^*$ with $|\mathcal{Y}^*| = m$:

$$\begin{aligned} d_{\pi^{-1}(\mathcal{Y}^{L}\cup\mathcal{Y}^{*})}(\omega') &= \frac{1}{k!k!} \sum_{s=1}^{k!k!} d_{\pi^{-1}(\sigma_{s}(\mathcal{Y}^{L}\cup\mathcal{Y}^{*}))}(\omega) \\ &= \frac{(k-n)!n!(k-m)!m!}{k!k!} \sum_{\{(\mathcal{Y}^{L})'\subseteq\mathcal{I}^{L}:|(\mathcal{Y}^{L})'|=n\}} \sum_{\{(\mathcal{Y}^{*})'\subseteq\mathcal{I}^{*}:|(\mathcal{Y}^{*})'|=m\}} d_{\pi^{-1}((\mathcal{Y}^{L})'\cup(\mathcal{Y}^{*})')}(\omega) \\ &= \binom{k}{n}^{-1} \binom{k}{m}^{-1} \sum_{\{(\mathcal{Y}^{L})'\subseteq\mathcal{I}^{L}:|(\mathcal{Y}^{L})'|=n\}} \sum_{\{(\mathcal{Y}^{*})'\subseteq\mathcal{I}^{*}:|(\mathcal{Y}^{*})'|=m\}} d_{\pi^{-1}((\mathcal{Y}^{L})'\cup(\mathcal{Y}^{*})')}(\omega) \\ &\equiv d_{(n,m)}(\omega'). \end{aligned}$$

We define $d_{(n,m)}(\omega')$ as indicated, which completes the proof. We capture this transformation by $\phi_3: \phi_2(\phi_1(\Omega)) \to \Omega$.

For example, consider applying ϕ_3 to the instance in Figure 4.4.3. The new demand is:

$$d_{\pi^{-1}(\{A\})}(\omega') = d_{\pi^{-1}(\{B\})}(\omega') = d_{(1,0)}(\omega') = \frac{1}{2} \left(d_{\pi^{-1}(\{A\})}(\omega) + d_{\pi^{-1}(\{B\})}(\omega) \right),$$

$$d_{\pi^{-1}(\{AC\})}(\omega') = d_{\pi^{-1}(\{AD\})}(\omega') = d_{\pi^{-1}(\{BC\})}(\omega') = d_{\pi^{-1}(\{BD\})}(\omega') = d_{(1,1)}(\omega')$$

$$= \frac{1}{4} \left(d_{\pi^{-1}(\{AC\})}(\omega) + d_{\pi^{-1}(\{AD\})}(\omega) + d_{\pi^{-1}(\{BC\})}(\omega) + d_{\pi^{-1}(\{BD\})}(\omega) \right),$$

and similarly for the other subsets.



Figure 4.4.4: Example of permuted MCLP instances. Bases are depicted as squares, demand points as black dots. The circles indicate the coverage of the bases. This example assumes that $\mathcal{I}(\omega) = \{A, B, C, D\}, p(\omega) = 2, \mathcal{I}^{L}(\omega) = \{A, B\}, \text{ and } \mathcal{I}^{*}(\omega) = \{C, D\}$. The demand points are unaffected by the permutations.

We can simplify many of the derived expressions and constraints for $\omega \in \phi_3(\phi_2(\phi_1(\Omega)))$. The maxima are given by:

$$\theta^{L}(\omega) = \sum_{n=1}^{k} \sum_{m=0}^{k} \binom{k}{n} \binom{k}{m} d_{(n,m)}(\omega),$$
$$\theta^{*}(\omega) = \sum_{m=1}^{k} \sum_{n=0}^{k} \binom{k}{m} \binom{k}{n} d_{(n,m)}(\omega).$$

For all $R \in \{1, \ldots, k\}$, the global maximum constraints are:

$$\sum_{r=1}^{R} \binom{R}{r} \sum_{n=0}^{k-R} \binom{k-R}{n} \left(d_{(n+r,0)}(\omega) - d_{(n,r)}(\omega) \right) \le 0.$$

In particular, the global and local maxima have the following constraint:

$$\theta^*(\omega) - \theta^L(\omega) = \sum_{m=1}^k \binom{k}{m} d_{(0,m)}(\omega) - \sum_{n=1}^k \binom{k}{n} d_{(n,0)}(\omega) \ge 0.$$

The ρ -Swap local maximum constraints are as follows. For all $R \in \{1, \dots, \rho\}$ we have

$$\sum_{r=1}^{R} \binom{R}{r} \sum_{m=0}^{k-R} \binom{k-R}{m} \left(d_{(0,m+r)}(\omega) - d_{(r,m)}(\omega) \right) \le 0.$$

Remark 4.4.7. These new definitions and constraints require $(k + 1)^2$ demand values. The determination of the worst-case instance can be modelled as a Linear Programming model where the demand values are the variables. The objective is of course to maximise the relative optimality gap. By implicitly normalising the demand by θ^* , we can add the constraint that $\theta^* = 1$ and have a linear objective function. This allows us to solve the Linear Programming model, determine the worst-case relative optimality gap, and thus derive the locality gap.

The final transformation $\phi_4 : \phi_3(\phi_2(\phi_1(\Omega))) \to \Omega$ combines demand in a weighted manner. To be specific, the demand of the new instance ω' is set to

$$d_{(1,0)}(\omega') = \frac{1}{k} \sum_{n=1}^{k} \binom{k}{n} d_{(n,0)}(\omega),$$

$$d_{(1,1)}(\omega') = \frac{1}{k^2} \sum_{n=1}^{k} \sum_{m=1}^{k} \binom{k}{n} \binom{k}{m} d_{(n,m)}(\omega),$$

$$d_{(0,1)}(\omega') = \frac{1}{k} \sum_{m=1}^{k} \binom{k}{m} d_{(0,m)}(\omega),$$

and zero otherwise. No other changes are performed. Figure 4.4.5 illustrates this transformation. We prove in Lemma 4.4.8 that this transformation preserves global optimality of x^* and ρ -Swap local optimality of x^L .



(a) Original instance $\omega \in \phi_3(\phi_2(\phi_1(\Omega)))$, with demand weights $d_{(n,m)}(\omega)$ abbreviated to $d_{(n,m)}$. Note that two points, corresponding to $d_{(2,0)}$ and $d_{(0,2)}$, are not displayed.



(b) New instance $\omega' = \phi_4(\omega)$, with new demand weights $d_{(n,m)}(\omega')$ abbreviated to $d'_{(n,m)}$.

Figure 4.4.5: Example of the fourth MCLP instance transformation ϕ_4 . Bases are depicted as squares, demand points as grey dots. The circles indicate the coverage of the bases. This example assumes that $\mathcal{I}(\omega) = \{A, B, C, D\}, p(\omega) = 2, \mathcal{I}^L(\omega) = \{A, B\}, \text{ and } \mathcal{I}^*(\omega) = \{C, D\}.$

Lemma 4.4.8. Consider an MCLP instance $\omega \in \phi_3(\phi_2(\phi_1(\Omega)))$ where the global optimum $x^*(\omega)$ and ρ -Swap local optimum $x^L(\omega)$ differ at 2k places. There exists an MCLP instance $\omega' \in \phi_3(\phi_2(\phi_1(\Omega)))$ such that $x^*(\omega)$ is still the global optimum and $x^L(\omega)$ still a ρ -Swap local optimum. The maxima satisfy the following relations:

$$\theta^{L}(\omega') = kd_{(1,0)}(\omega') + k^{2}d_{(1,1)}(\omega'),$$

$$\theta^{*}(\omega') = kd_{(0,1)}(\omega') + k^{2}d_{(1,1)}(\omega'),$$

with the (necessary and sufficient) constraints

$$d_{(0,1)}(\omega') - d_{(1,0)}(\omega') \ge 0,$$

$$d_{(0,1)}(\omega') - d_{(1,0)}(\omega') - (k - \rho)d_{(1,1)}(\omega') \le 0.$$

Furthermore, the relative optimality gap of ω' is equal to that of ω :

$$\frac{\theta^*(\omega') - \theta^L(\omega')}{\theta^*(\omega')} = \frac{\theta^*(\omega) - \theta^L(\omega)}{\theta^*(\omega)}$$

Proof. Let $\omega \in \phi_3(\phi_2(\phi_1(\Omega)))$ be an arbitrary instance where the global optimum and ρ -Swap local optimum differ at 2k places. Apply the final transformation to ω : $\omega' = \phi_4(\omega)$. As a result, the objective value of x^L remains the same:

$$\begin{aligned} \theta^{L}(\omega') &= k d_{(1,0)}(\omega') + k^{2} d_{(1,1)}(\omega') \\ &= \sum_{n=1}^{k} \binom{k}{n} d_{(n,0)}(\omega) + \sum_{n=1}^{k} \sum_{m=1}^{k} \binom{k}{n} \binom{k}{m} d_{(n,m)}(\omega) = \theta^{L}(\omega). \end{aligned}$$

Similarly, the objective value for x^* is unaffected:

$$\begin{aligned} \theta^*(\omega') &= k d_{(0,1)}(\omega') + k^2 d_{(1,1)}(\omega') \\ &= \sum_{m=1}^k \binom{k}{m} d_{(0,m)}(\omega) + \sum_{n=1}^k \sum_{m=1}^k \binom{k}{n} \binom{k}{m} d_{(n,m)}(\omega) = \theta^*(\omega). \end{aligned}$$

Recall that the global maximum constraints are given by:

$$\begin{split} \sum_{r=1}^{R} \binom{R}{r} \sum_{n=0}^{k-R} \binom{k-R}{n} \left(d_{(n+r,0)}(\omega') - d_{(n,r)}(\omega') \right) \\ &= R \left(d_{(1,0)}(\omega') - d_{(0,1)}(\omega') - (k-R)d_{(1,1)}(\omega') \right) \le 0, \end{split}$$

which must hold for all $R \in \{1, ..., k\}$. Notice that the constraint for R = k is the most restricting, i.e., the global maximum constraints are satisfied if and only if

$$d_{(1,0)}(\omega') - d_{(0,1)}(\omega') \le 0$$

This constraint is indeed valid for ω' :

$$d_{(1,0)}(\omega') - d_{(0,1)}(\omega') = \frac{1}{k} \left(\theta^L(\omega') - \theta^*(\omega') \right) = \frac{1}{k} \left(\theta^L(\omega) - \theta^*(\omega) \right) \le 0.$$

Here, we have used the optimality of x^* for ω : $\theta^*(\omega) - \theta^L(\omega) \ge 0$. We conclude that x^* is still a global optimum for ω' .

The final step is to check if the ρ -Swap constraints are valid for x^L and ω' . Unfortunately, this part of the proof is somewhat cumbersome. Let $R \in \{1, \ldots, \rho\}$ and recall the expression in the ρ -Swap constraints:

$$\begin{split} \sum_{r=1}^{R} \binom{R}{r} \sum_{m=0}^{k-R} \binom{k-R}{m} \left(d_{(0,m+r)}(\omega') - d_{(r,m)}(\omega') \right) \\ &= R \left(d_{(0,1)}(\omega') - d_{(1,0)}(\omega') - (k-R)d_{(1,1)}(\omega') \right) \\ &= \frac{R}{k} \sum_{m=1}^{k} \binom{k}{m} d_{(0,m)}(\omega) - \frac{R}{k} \sum_{n=1}^{k} \binom{k}{n} d_{(n,0)}(\omega) - \frac{R(k-R)}{k^2} \sum_{n=1}^{k} \sum_{m=1}^{k} \binom{k}{n} d_{(n,m)}(\omega), \end{split}$$

which must be non-positive. First, we focus on the two negative terms:

$$-\frac{R}{k}\sum_{n=1}^{k}\binom{k}{n}d_{(n,0)}(\omega) - \frac{R(k-R)}{k^{2}}\sum_{n=1}^{k}\sum_{m=1}^{k}\binom{k}{n}\binom{k}{m}d_{(n,m)}(\omega)$$

$$\leq -\frac{R}{k}\sum_{n=1}^{R}\binom{k}{n}d_{(n,0)}(\omega) - \frac{R(k-R)}{k^{2}}\sum_{n=1}^{R}\sum_{m=1}^{k-R}\binom{k}{n}\binom{k}{m}d_{(n,m)}(\omega)$$

$$\leq -\sum_{n=1}^{R}\binom{R}{n}d_{(n,0)}(\omega) - \sum_{n=1}^{R}\sum_{m=1}^{k-R}\binom{R}{n}\binom{k-R}{m}d_{(n,m)}(\omega)$$

$$= -\sum_{r=1}^{R}\binom{R}{r}\sum_{m=0}^{k-R}\binom{k-R}{m}d_{(r,m)}(\omega).$$
(4.4.7)

For the second inequality we have used that

$$\frac{R}{k}\binom{k}{n} = R\frac{(k-1)\cdots(k-n+1)}{n!} \ge R\frac{(R-1)\cdots(R-n+1)}{n!} = \binom{R}{n},$$

and

$$\begin{aligned} \frac{k-R}{k} \binom{k}{m} &= (k-R) \frac{(k-1)\cdots(k-m+1)}{m!} \\ &\geq (k-R) \frac{(k-R-1)\cdots(k-R-m+1)}{m!} = \binom{k-R}{m}. \end{aligned}$$

Second, suppose we can bound the positive term of the ρ -Swap constraint as follows:

$$\frac{R}{k} \sum_{m=1}^{k} \binom{k}{m} d_{(0,m)}(\omega) \le \sum_{r=1}^{R} \binom{R}{r} \sum_{m=0}^{k-R} \binom{k-R}{m} d_{(0,m+r)}(\omega).$$
(4.4.8)

By combining Equations (4.4.7) and (4.4.8), we can prove that the ρ -Swap constraints are valid:

$$\sum_{r=1}^{R} {\binom{R}{r}} \sum_{m=0}^{k-R} {\binom{k-R}{m}} \left(d_{(0,m+r)}(\omega') - d_{(r,m)}(\omega') \right)$$
$$\leq \sum_{r=1}^{R} {\binom{R}{r}} \sum_{m=0}^{k-R} {\binom{k-R}{m}} \left(d_{(0,m+r)}(\omega) - d_{(r,m)}(\omega) \right) \leq 0.$$

Thus, only the proof of Equation (4.4.8) remains. First apply an identity, also known as the Chu-Vandermonde identity:

$$\frac{R}{k}\sum_{m=1}^{k} \binom{k}{m} d_{(0,m)}(\omega) = \frac{R}{k}\sum_{m=1}^{k}\sum_{r=0}^{m} \binom{R}{r} \binom{k-R}{m-r} d_{(0,m)}(\omega).$$
(4.4.9)

We can switch the order of summation and delete terms with binomial coefficients equal zero. Consequently, Equation (4.4.9) is equal to:

$$\frac{R}{k} \sum_{m=1}^{k} \binom{k-R}{m} d_{(0,m)}(\omega) + \frac{R}{k} \sum_{r=1}^{k} \sum_{m=r}^{k} \binom{R}{r} \binom{k-R}{m-r} d_{(0,m)}(\omega)
= \frac{R}{k} \sum_{m=1}^{k-R} \binom{k-R}{m} d_{(0,m)}(\omega) + \frac{R}{k} \sum_{r=1}^{R} \sum_{m=r}^{k-R+r} \binom{R}{r} \binom{k-R}{m-r} d_{(0,m)}(\omega)
= \frac{R}{k} \sum_{m=1}^{k-R} \binom{k-R}{m} d_{(0,m)}(\omega) + \left(1 - \frac{k-R}{k}\right) \sum_{r=1}^{R} \sum_{m=0}^{k-R} \binom{R}{r} \binom{k-R}{m} d_{(0,m+r)}(\omega).$$

Notice that we have been able to obtain the right-hand side of Equation (4.4.8), but there are some additional terms. Therefore, consider these additional terms:

$$\frac{R}{k} \sum_{m=1}^{k-R} \binom{k-R}{m} d_{(0,m)}(\omega) - \frac{k-R}{k} \sum_{r=1}^{R} \sum_{m=0}^{k-R} \binom{R}{r} \binom{k-R}{m} d_{(0,m+r)}(\omega)$$

$$\leq \frac{R}{k} \sum_{m=1}^{k-R} \binom{k-R}{m} d_{(0,m)}(\omega) - \frac{k-R}{k} \sum_{m=1}^{k-R} \binom{k-R}{m-1} d_{(0,m)}(\omega)$$

$$\leq \frac{R}{k} \sum_{m=1}^{k-R} \binom{k-R}{m} - (k-R)\binom{k-R}{m-1} d_{(0,m)}(\omega) \leq 0.$$

The non-positivity follows from the estimation

$$\begin{split} (k-R)\binom{k-R}{m-1} &= (k-R)\frac{(k-R)\cdots(k-R-m+2)}{(m-1)!} \\ &\geq \frac{(k-R)\cdots(k-R-m+1)}{(m-1)!} \geq \binom{k-R}{m}. \end{split}$$

As the sum of these extra terms is non-positive, we have shown that Equation (4.4.8) is valid. Hence, the ρ -Swap optimality conditions hold for x^L and ω' .

To conclude, we have a final instance ω' with

$$\begin{aligned} \theta^{L}(\omega') &= k d_{(1,0)}(\omega') + k^{2} d_{(1,1)}(\omega'), \\ \theta^{*}(\omega') &= k d_{(0,1)}(\omega') + k^{2} d_{(1,1)}(\omega'), \end{aligned}$$

and the constraints

$$d_{(0,1)}(\omega') - d_{(1,0)}(\omega') \ge 0,$$

$$d_{(0,1)}(\omega') - d_{(1,0)}(\omega') - (k - \rho)d_{(1,1)}(\omega') \le 0.$$

Note that only the most restricting global maximum and $\rho\mbox{-Swap}$ local maximum constraints need to be included. $\hfill \Box$

We can now prove Theorem 4.4.4 by using the four instance transformations and the three lemmas. In the proof we derive the maximum relative optimality gap of a ρ -Swap local maximum for given parameters k and ρ .

Proof of Theorem 4.4.4. Let $\omega \in \Omega$ be an arbitrary MCLP instance with global maximum $\theta^*(\omega)$ and optimal solution $x^*(\omega)$. Let $x^L(\omega)$ be a solution found by the ρ -Swap Local Search method with ρ -Swap local maximum $\theta^L(\omega)$. Suppose $x^L(\omega)$ differs from $x^*(\omega)$ in exactly $2k(\omega)$ places with $k(\omega) \in \mathbb{N}_{>\rho}$.

Sequentially apply the above described instance transformations ϕ_1, \ldots, ϕ_4 to ω , resulting in a new instance ω' . We have shown that each transformation preserves the following properties:

- $x^*(\omega)$ is an optimal solution for ω' (albeit projected to a lower dimension by ϕ_1),
- $x^{L}(\omega)$ is a ρ -Swap local optimum for ω' (albeit projected to a lower dimension by ϕ_1),
- $x^{L}(\omega)$ differs from $x^{*}(\omega)$ in exactly $2k(\omega)$ places,

and the relative optimality gap of ω' is as least as large as that of ω .

The transformation resulted in a family of worst-case instances with symmetric properties. Using Lemma 4.4.8, we can describe this family by the (abstract) relations:

$$\begin{aligned} \theta^{L}(\omega') &= k(\omega)d_{(1,0)}(\omega') + (k(\omega))^{2}d_{(1,1)}(\omega'), \\ \theta^{*}(\omega') &= k(\omega)d_{(0,1)}(\omega') + (k(\omega))^{2}d_{(1,1)}(\omega'), \\ d_{(0,1)}(\omega') - d_{(1,0)}(\omega') &\geq 0, \\ d_{(0,1)}(\omega') - (k(\omega) - \rho)d_{(1,1)}(\omega') &\leq 0. \end{aligned}$$

Since the relative optimality gap is given by

$$\frac{\theta^*(\omega') - \theta^L(\omega')}{\theta^*(\omega')} = \frac{d_{(0,1)}(\omega') - d_{(1,0)}(\omega')}{d_{(0,1)}(\omega') + k(\omega)d_{(1,1)}(\omega')},$$

we are interested in the following optimisation model:

Maximise

$$f(u, v, w) = \frac{u - v}{u + k(\omega)u}$$

subject to

$$\begin{aligned} u-v &\geq 0, \\ u-v-(k(\omega)-\rho)w &\leq 0, \\ u,v,w &\in \mathbb{R}_{\geq 0}. \end{aligned}$$

The decision variable u corresponds to demand $d_{(0,1)}(\omega')$, v to demand $d_{(1,0)}(\omega')$, and w to demand $d_{(1,1)}(\omega')$. The objective is to maximise the relative optimality gap, whilst satisfying the global and ρ -Swap local maxima constraints. The resulting optimal objective value is the maximum relative optimality gap for the parameters $k(\omega)$ and ρ .

Clearly, it is optimal to set w as small as possible:

$$w = \frac{u - v}{k(\omega) - \rho}.$$

The objective simplifies to

$$f(u,v) = \frac{(k(\omega) - \rho)(u - v)}{(k(\omega) - \rho)u + k(\omega)(u - v)} = \frac{(k(\omega) - \rho)(u - v)}{(2k(\omega) - \rho)u - k(\omega)v},$$

with

$$\frac{\partial f}{\partial v}(u,v) = -\frac{(k(\omega)-\rho)^2 u}{((2k(\omega)-\rho)u-k(\omega)v)^2} \le 0.$$

Since the partial derivative of f with respect to v is non-positive, it is optimal to set v = 0. Hence, the new objective is:

$$f(u) = \frac{(k(\omega) - \rho)u}{(2k(\omega) - \rho)u} = \frac{k(\omega) - \rho}{2k(\omega) - \rho}.$$

In conclusion, we can bound the relative optimality gap by

$$\frac{\theta^*(\omega) - \theta^L(\omega)}{\theta^*(\omega)} \le \frac{k(\omega) - \rho}{2k(\omega) - \rho}$$

Since we do not know the exact value of $k(\omega)$ (although $k(\omega) \in \{0, \ldots, p(\omega)\}$), we have to consider the worst-case. Note that the family of worst-case instances is given by a fixed $d_{(0,1)} \ge 0$ and $d_{(1,1)} = d_{(0,1)}/(k-\rho)$ with $k \in \mathbb{N}$ ($k \ge \rho+1$). The relative optimality gap of this family converges in the following sense:

$$\lim_{k(\omega)\to\infty}\frac{\theta^*(\omega)-\theta^L(\omega)}{\theta^*(\omega)}\leq \lim_{k(\omega)\to\infty}\frac{k(\omega)-\rho}{2k(\omega)-\rho}=\frac{1}{2}.$$

Thus, the relative locality gap of ρ -Swap Local Search for the MCLP is:

$$\sup_{\omega\in\Omega, L\in\mathcal{L}(\omega)}\left\{\frac{\theta^*(\omega)-\theta^L(\omega)}{\theta^*(\omega)}\right\}=\frac{1}{2}.$$

We have shown how to construct a family of worst-case instances with a relative optimality gap converging to the relative locality gap. $\hfill \Box$

Although the proof of Theorem 4.4.4 describes how the worst-case MCLP instances can be constructed, it is useful to give some explicit examples. For $\rho \in \mathbb{N}_{\geq 1}$ and $k \in \mathbb{N}$ with $k \geq \rho + 1$, the worst-case MCLP instance has the following parameters. The number of opened bases p is equal to k and there are in total $|\mathcal{I}| = 2p = 2k$ bases. The first p bases correspond to a ρ -Swap Local Search optimum ($\mathcal{I}^L = \{1, \ldots, p\}$) and the last p bases to the optimal solution ($\mathcal{I}^* = \{p+1, \ldots, 2p\}$). There are p(p+1) demand points, where the first p^2 points are covered by exactly one ρ -Swap base and one optimal base. These points have demand ($k - \rho$)⁻¹. The final p demand points are covered by exactly one optimal base and have demand 1. This construction is permutation invariant between the two types of bases.

In Figure 4.4.6 we give three examples for $\rho = 1$ and $k \in \{2, 3\}$, followed by $\rho = 2$ and k = 3. The parameter $\epsilon > 0$ is insignificantly small. We denote the adjacency parameter by $a_{ij}^{(k)}$ and the demand by $d_j^{(\rho)}$. It is straightforward to verify that the first p bases form a ρ -Swap Local Search optimum and the last p bases are the optimal solution. The relative optimality gap matches the bound in Theorem 4.4.4 (as ϵ decreases to zero). It is interesting to note that the Greedy Search would find the optimal solution for these instances.

As mentioned in the proof of Theorem 4.4.2, Nemhauser, Wolsey, and Fisher (1978) provide a family of worst-case Uncapacitated Location problem instances. Using the transformation shown in Remark 4.3.5, it turns out that after rescaling this gives the same family of worst-case MCLP instances as found in the proof of Theorem 4.4.4. Nevertheless, our proof gradually builds a description of worst-case instances, which has the potential to be used for other optimisation problems as well. In particular, we note that we reduce the number of demand parameters from $2^{|\mathcal{I}|}$ to $(k+1)^2$ by symmetry, followed by another reduction to 3 demand parameters, and finally only one parameter remains.

The computational time of the ρ -Swap Local Search is not polynomial. Each iteration requires the evaluation of $\mathcal{O}(\binom{|\mathcal{I}|}{\rho}) = \mathcal{O}(|\mathcal{I}|^{\rho})$ subsets of bases, each taking at most $\mathcal{O}(\rho|\mathcal{J}|)$ time. In practice, the submodularity of the objective can be used to implement a lazy evaluation (updating the value of a subset only when required). This would improve the average computational time of an iteration. However, the main issue is the number of required iterations, which can be exponentially many (see for example Nemhauser, Wolsey, and Fisher (1978)).

When comparing the performance guarantees of the Greedy Search and the Swap Local Search, it is remarkable that the Greedy Search outperforms the Swap heuristic with respect to worst-case performance. In Section 4.5 we compare both heuristics using realistic and randomly generated instances. These results give an indication on the average performance of both heuristics.






(b) Example for $\rho = 1$ and k = 3.



(c) Example for $\rho = 2$ and k = 3.

Figure 4.4.6: Worst-case examples for Swap Local Search. Bases are depicted as squares, demand points as circles. Points are connected to adjacent bases and their demand is shown in brackets.

4.5 Numerical Results

We have discussed several theoretical results on the worst-case performance of approximation algorithms for the MCLP. As these worst-case instances rarely appear in real-life decision problems, it is useful to empirically evaluate the performance for realistic instances. In order to do so, we have two sets of instances. The first set consists of the realistic instances based on the 24 Regional Ambulance Services (RAVs) of The Netherlands, as described in Section 3.4.1. The second set consists of randomly generated instances based on Euclidean distances, which will be described in Section 4.5.3.

At our disposal are four Greedy Search variations (solution construction methods) and the Swap Local Search (a solution improvement method). The variations of the Greedy Search are:

- the Greedy Search, which opens bases and maximises coverage,
- the Reverse Greedy Search, which closes bases and maximises coverage,
- the Anti Greedy Search, which opens bases and minimises coverage,
- the Revised Anti Greedy Search, which opens non-adjacent bases and minimises coverage.

The Greedy Search and Reverse Greedy Search have been introduced in Section 4.3 and 4.3.3, respectively. The Anti Greedy Search is similar to the Greedy Search, except it minimises coverage at each iteration. It will be used to create an initial solution with low coverage for the Swap Local Search. The Revised Anti Greedy Search is a modification of the Anti Greedy Search: at each iteration it can only open bases that are non-adjacent. That is, a selected base is not allowed to be covered by a different opened base. This leads to base locations that are more spread out. In case of full coverage, arbitrary additional bases are opened if required.

Thus, we have four⁶ different ways to construct the initial solutions for the Swap Local Search: two initial solutions will have high coverage and two low coverage. This allows us to determine how robust the Swap Local Search method is with respect to the initial solution. For the details on the implementation, see Appendix B.

Both the Greedy Search and the Swap Local Search require the specification of the search parameter $\rho \in \mathbb{N}$. We consider the 1-Greedy Search and the 1-Swap Local Search as the reference solution methods. Furthermore, we have two additional settings to determine the effect of increasing ρ : 2-Greedy with 1-Swap, and 1-Greedy with 2-Swap. We change only one parameter with respect to the reference to be able to discern its effect. Therefore, we do not consider $\rho = 2$ for both methods simultaneously.

We start with the realistic set of RAV instances in Sections 4.5.1 and 4.5.2. In Section 4.5.3 the random generation of the instances is discussed, followed by the solution method performances. Similar to the sensitivity analysis in Section 3.4, we restrict the analysis to attainable optimality gaps and do not discuss the actual solutions. In Section 4.6 we go into the details of the results for two RAV regions, namely RAV14 (Gooi- en Vechtstreek) and RAV23 (Limburg-Noord). These two regions have been discussed in detail in the sensitivity analysis.

 $^{^{6}\}mathrm{We}$ have also evaluated the Reverse variant of the Anti Greedy Search. The results are omitted, as they lead to similar conclusions.

4.5.1 Greedy Search Performance for Realistic Instances

As mentioned before, we have four solution construction methods, all based on the Greedy Search. Furthermore, each construction method is applied in two ways: with search parameter ρ equal to 1 or 2. This leads to eight different solutions for the RAV regions. In Theorem 4.3.3 we have determined bounds for the worst-case performance of the Greedy Search. For $p = a\rho - b$ with $a \in \mathbb{N}_{\geq 1}$ and $b \in \{0, \ldots, \rho - 1\}$, the following bound holds for the ρ -Greedy maximum θ^G and the global maximum θ^* :

$$\frac{\theta^* - \theta^G}{\theta^*} \leq \left(\frac{a - \frac{1}{\rho}(\rho - b)}{a}\right) \left(\frac{a - 1}{a}\right)^{a - 1}.$$

Table 4.5.1 gives the resulting theoretical bounds on the relative optimality gap for each of the RAV regions, where the average bound for the RAV regions is approximately 35%.

Fortunately, the achieved relative optimality gaps are much smaller, see the performance of the Greedy Search and the other construction methods in Table 4.5.2. For each RAV and construction method the relative optimality gap (rounded to four decimals) is given. Furthermore, the performance of each method is summarised by the average and standard deviation of the relative optimality gaps. Entries highlighted in green correspond to the best solution.

Consider only the solutions for $\rho = 1$. The 1-Greedy Search method has the best result of all four construction methods for all but one RAV region (RAV20). For eight RAV regions the 1-Greedy solution is optimal and the average relative optimality gap is 0.43%. Naturally, the 1-Greedy Search outperforms the (Revised) Anti Greedy Search.

Furthermore, we have shown in Section 4.3.3 that the Reverse Greedy Search has a worst-case relative optimality gap of 1. Therefore, it is not surprising that the Greedy Search outperforms the Reverse Greedy Search as well. In fact, the average relative optimality gap of the Reverse 1-Greedy Search is 1.62%, four times larger than that of the 1-Greedy Search. Do note that the relative optimality gaps of the Reverse Greedy Search are nevertheless decent.

When comparing the results for the two search parameters ρ , we notice that the 2-Greedy Search only outperforms the 1-Greedy Search for 4 RAV regions. For the remaining 20 regions both methods have the same optimality gap. The average relative optimality gap of the 2-Greedy Search is 0.39%, approximately a 10% relative improvement of the 1-Greedy Search. Remarkably, the Reverse 2-Greedy Search finds the optimal solution for RAV20, whereas it does not improve the Reverse 1-Greedy Search for the other RAVs. Consequently, the average performance of the Reverse 2-Greedy Search is almost the same as the Reverse 1-Greedy Search.

Since the Anti methods are only used to construct bad initial solutions for the Swap Local Search, a large relative optimality gap is desired. Overall, the Anti Greedy Search constructs solutions with a larger gap than the Revised Anti Greedy Search. Increasing search parameter ρ to 2 does not necessarily lead to a solution with a larger gap. In fact, the average relative gap of the Anti 2-Greedy Search is 87.95%, only slightly larger than that of the Anti 1-Greedy Search (87.01%). For the Revised Anti Greedy Search the difference between $\rho = 1$ and $\rho = 2$ is also small (60.70% and 61.27%, respectively). Notice that the Revised Anti method has difficulty constructing a bad solution for RAV14, the smallest region, and finds the optimal solution for RAV15.

We do note that most RAV regions have a surplus of bases. This implies that the Greedy Search can use the extra capacity (the surplus bases) to compensate for a suboptimal selection of bases during the search. Consequently, the low average optimality gap was expected. However, the relative optimality gaps for regions with a shortage of bases (RAV2, RAV7, and RAV23) are also small (0.89%, 0.76%, and 2.71%, respectively).

To conclude, the Greedy Search provides the best solutions. Reversing the search is counterproductive in general. For the Greedy Search it is beneficial to increase the search parameter ρ to 2, although the improvement is small. Recall that the 1-Greedy Search has a running time of $\mathcal{O}(p|\mathcal{I}|^2)$, whereas the 2-Greedy requires $\mathcal{O}(p|\mathcal{I}|^3)$ time. With our implementation there is a significant (but manageable) increase in computational time. Given the small improvement in the average relative optimality gap, the additional computational time could be better used to apply a solution improvement method, such as the Swap Local Search.

	Worst-Case Rela	tive Optimality Gap
RAV	of ρ -Gre	edy Search
Region	ho=1	ho=2
1	0.3533	0.3682
2	0.3464	0.3686
3	0.3464	0.3686
4	0.3399	0.3691
5	0.2963	0.3750
6	0.3436	0.3164
7	0.3487	0.3277
8	0.3487	0.3277
9	0.3349	0.2963
10	0.3505	0.3684
11	0.3580	0.3680
12	0.3533	0.3682
14	0.3399	0.3691
15	0.3399	0.3691
16	0.3399	0.3691
17	0.3164	0.2500
18	0.3349	0.2963
19	0.3533	0.3682
20	0.3487	0.3277
21	0.3464	0.3686
22	0.3533	0.3682
23	0.3399	0.3691
24	0.3505	0.3684
25	0.3505	0.3684

Table 4.5.1: Theoretical bounds for the relative optimality gap of ρ -Greedy Search for the RAV regions.

	Relative Optimality Gap of ρ -Greedy Search										
RAV	Nor	mal	Rev	rerse	A	nti	Revised Anti				
Region	$\rho = 1$	ho=2	ho = 1	ho=2	$\rho = 1$	ho=2	ho = 1	ho=2			
1	0.0017	0.0017	0.0020	0.0020	0.9411	0.9411	0.7260	0.7355			
2	0.0089	0.0089	0.0240	0.0240	0.9783	0.9783	0.8494	0.8494			
3	0.0063	0.0046	0.0187	0.0187	0.9410	0.9410	0.6667	0.6762			
4	0.0032	0.0032	0.0099	0.0099	0.9458	0.9458	0.7403	0.7047			
5	0	0	0	0	0.8680	0.8899	0.4303	0.4187			
6	0.0017	0.0017	0.0203	0.0203	0.8507	0.8803	0.6111	0.6281			
7	0.0076	0.0040	0.0114	0.0114	0.9138	0.9138	0.7150	0.7150			
8	0	0	0.0001	0.0001	0.8711	0.8450	0.4957	0.5498			
9	0.0021	0.0021	0.0044	0.0044	0.9183	0.9438	0.6610	0.6801			
10	0	0	0.0143	0.0143	0.9522	0.9522	0.6435	0.6834			
11	0	0	0	0	0.8921	0.8921	0.6746	0.6746			
12	0	0	0	0	0.7484	0.7519	0.3351	0.3184			
14	0	0	0	0	0.4369	0.4369	0.1464	0.1464			
15	0	0	0	0	0.6194	0.6621	0	0			
16	0	0	0	0	0.8984	0.8984	0.4903	0.5068			
17	0.0044	0.0044	0.0067	0.0067	0.9547	0.9547	0.8380	0.8380			
18	0.0042	0.0042	0.0164	0.0164	0.8750	0.8750	0.6926	0.7137			
19	0.0106	0.0080	0.0262	0.0262	0.8338	0.8759	0.7443	0.7443			
20	0.0022	0.0011	0.0013	0	0.8460	0.8946	0.6606	0.6606			
21	0.0004	0.0004	0.0262	0.0262	0.9365	0.9365	0.6288	0.6288			
22	0.0143	0.0143	0.0638	0.0638	0.9093	0.9093	0.7623	0.7623			
23	0.0271	0.0271	0.0672	0.0672	0.9439	0.9439	0.7377	0.7377			
24	0.0052	0.0052	0.0498	0.0498	0.9164	0.9164	0.6466	0.6597			
25	0.0038	0.0038	0.0265	0.0265	0.8920	0.9295	0.6716	0.6716			
Average	0.0043	0.0039	0.0162	0.0162	0.8701	0.8795	0.6070	0.6127			
Standard Deviation	0.0062	0.0061	0.0197	0.0198	0.1195	0.1164	0.2039	0.2048			

Table 4.5.2: Relative optimality gap of ρ -Greedy Search for the RAVs. The solutions with the smallest gaps are highlighted in green.

4.5.2 Swap Local Search Performance for Realistic Instances

For an arbitrary initial solution, the worst-case relative optimality gap of the ρ -Swap Local Search is at most 50%, as shown in Theorem 4.4.4. In particular, for global maximum θ^* and the resulting ρ -Swap coverage θ^L it holds that

$$\frac{\theta^* - \theta^L}{\theta^*} \le \frac{p - \rho}{2p - \rho}.$$

The corresponding theoretical bounds on the relative optimality gaps are given in Table 4.5.3. The average bound is approximately 45%. However, for a given initial solution, we know that the Swap Local Search can only improve it. To determine how robust the Swap Local Search is with respect to the initial solution, we evaluate its performance using the eight constructed solutions of Section 4.5.1 as initial solutions. Recall that we apply the 1-Swap Local Search to all eight initial solutions, and the 2-Swap Local Search only to the four 1-Greedy variants. Thus, in total we have 12 different Swap Local Search solutions for each RAV.

The relative optimality gaps for the Swap Local Search solutions are given in Table 4.5.4. As usual, the results are rounded to four decimals (note that 0 implies optimality and 0.0000 a very small gap). The solutions are distinguished with respect to the initial Greedy solution and the search parameters. For example, the column (2, 1) of the Reverse Greedy initial solution corresponds to the Reverse 2-Greedy Search followed by the 1-Swap Local Search. Entries highlighted in green are the best solutions for the RAV region. The required number of Swap iterations for convergence to a ρ -Swap local maximum are shown in Table 4.5.5. Therefore, if only one iteration was used, the initial solution was already a ρ -Swap local maximum.

Let us first only consider the 1-Greedy variants and the 1-Swap (columns (1,1)). The average relative optimality gap of 1-Greedy solution is halved by the 1-Swap Local Search and the improvement is even greater for the other three initial solutions. Notice that the 1-Greedy and the Anti 1-Greedy result in the same average relative gap of 0.19%. The Reverse 1-Greedy and the Revised Anti 1-Greedy initial solutions lead to worse averages of 0.29% and 0.23%, although still very small average gaps. The four initial solutions lead to an optimal solution for almost the same RAV regions. This indicates that the Swap Local Search is robust with respect to the initial solution, although the (Revised) Anti Greedy initial solutions need more Swap iterations.

The 1-Swap Local Search solutions differ more for the 2-Greedy variants as initial solutions (columns (2,1)). The average relative optimality gaps for the Greedy, Reverse Greedy, Anti Greedy, and Revised Anti Greedy initial solutions are 0.14%, 0.29%, 0.17%, and 0.31%, respectively. Although the 1-Swap Local Search performs very well for all initial solutions, the relatively large difference in average gap indicates that the initial solution is an important factor for the performance. Note that the Anti Greedy initial solutions lead to a smaller 1-Swap average gap than the Revised Greedy solutions, even though the Revised Greedy initial solutions are much better than those of the Anti Greedy Search.

The best improvement can be achieved by applying a 2-Swap Local Search: between 18 and 20 RAV regions are solved to optimality, depending on the initial solution. Furthermore, the average relative optimality gaps are 0.03% (Greedy and Reverse Greedy) and 0.02% (Anti Greedy and Revised Anti Greedy). The maximum relative gap is 0.44%. We can therefore conclude that the 2-Swap Local Search is robust with respect to the initial solution.

Of course, the relative optimality gap is not the only performance measure: the computational time should also be considered. As can be seen in Table 4.5.5, the Greedy Search initial solution leads to the smallest number of required ρ -Swap iterations on average. For example, using the Anti Greedy solutions requires at least twice as many iterations. Therefore, the Greedy Search provides superior initial solutions in general.

The iterations of both the 2-Greedy Search and the 2-Swap Local Search require significantly (but manageably) more computational time. Thus, considering the evaluated heuristics, we advise to apply the 1-Greedy Search followed by either the 1- or 2-Swap Local Search (depending on the available computational time).

RAV	Worst-Case Relative Optimality Gap of ρ -Swap Local Search					
Region	$\rho = 1$	ho=2				
1	0.4800	0.4583				
2	0.4706	0.4375				
3	0.4706	0.4375				
4	0.4615	0.4167				
5	0.4000	0.2500				
6	0.4667	0.4286				
7	0.4737	0.4444				
8	0.4737	0.4444				
9	0.4545	0.4000				
10	0.4762	0.4500				
11	0.4865	0.4722				
12	0.4800	0.4583				
14	0.4615	0.4167				
15	0.4615	0.4167				
16	0.4615	0.4167				
17	0.4286	0.3333				
18	0.4545	0.4000				
19	0.4800	0.4583				
20	0.4737	0.4444				
21	0.4706	0.4375				
22	0.4800	0.4583				
23	0.4615	0.4167				
24	0.4762	0.4500				
25	0.4762	0.4500				

Table 4.5.3: Theoretical bounds for the relative optimality gap of ρ -Swap Local Search for the RAV regions.

Relative Optimality Gap of ρ^S -Swap Local Search after ρ^G -Greedy Search for (ρ^G, ρ^S)												
RAV	Noi	rmal Gre	edy	Rev	erse Gre	edy	Anti Greedy			Revised Anti Greedy		
Region	(1, 1)	(2 , 1)	(1 , 2)	$({f 1},{f 1})$	(2 , 1)	(1 , 2)	(1, 1)	(2 , 1)	(1 , 2)	(1, 1)	(2 , 1)	(1 , 2)
1	0.0005	0.0005	0	0.0020	0.0020	0	0.0007	0.0007	0	0.0008	0.0013	0
2	0.0051	0.0051	0.0011	0.0072	0.0072	0.0004	0.0033	0.0033	0.0011	0.0061	0.0061	0.0022
3	0.0002	0.0028	0	0.0023	0.0023	0	0.0031	0.0031	0	0	0	0
4	0.0012	0.0012	0	0.0026	0.0026	0	0.0012	0.0012	0	0.0012	0.0036	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0.0017	0.0017	0.0001	0.0010	0.0010	0.0001	0.0012	0.0011	0.0003	0.0017	0.0014	0.0001
7	0.0003	0.0003	0	0.0090	0.0090	0	0.0039	0.0039	0	0.0023	0.0023	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0.0001	0.0001	0	0.0042	0.0042	0.0013	0.0013	0.0001	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0.0001	0.0001	0
12	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0
17	0.0031	0	0	0.0000	0.0000	0.0000	0.0044	0.0044	0	0	0	0.0000
18	0.0042	0.0042	0	0	0	0	0.0042	0.0042	0	0	0.0042	0
19	0.0106	0	0.0021	0.0185	0.0185	0	0.0021	0.0021	0.0021	0.0069	0.0069	0
20	0.0001	0.0001	0	0.0009	0	0	0.0001	0.0001	0	0.0022	0.0022	0
21	0	0	0	0.0021	0.0021	0	0	0	0	0	0	0
22	0.0088	0.0088	0.0028	0.0128	0.0128	0.0044	0.0112	0.0112	0	0.0112	0.0112	0
23	0.0047	0.0047	0	0.0024	0.0024	0.0016	0.0047	0.0047	0.0016	0.0189	0.0189	0.0016
24	0.0052	0.0052	0.0021	0.0052	0.0052	0	0	0	0	0	0.0125	0
25	0	0	0	0	0	0	0.0038	0	0	0.0038	0.0038	0
Average	0.0019	0.0014	0.0003	0.0029	0.0029	0.0003	0.0019	0.0017	0.0002	0.0023	0.0031	0.0002
Standard Deviation	0.0030	0.0024	0.0008	0.0047	0.0047	0.0010	0.0026	0.0026	0.0006	0.0045	0.0049	0.0005

Table 4.5.4: Relative optimality gap of ρ -Swap Local Search for the RAV regions. The smallest gaps are highlighted in green.

$ ho^{S}$ -Swap Local Search Iterations after $ ho^{G}$ -Greedy Search for (ho^{G}, ho^{S})												
RAV	No	rmal Gre	edy	Rev	verse Gre	edy	Α	nti Gree	dy	Revised Anti Greedy		
Region	$({\bf 1},{\bf 1})$	(2 , 1)	$({\bf 1},{\bf 2})$	(1, 1)	(2 , 1)	$({\bf 1},{\bf 2})$	(1, 1)	(2 , 1)	$({\bf 1},{\bf 2})$	(1, 1)	(2 , 1)	(1 , 2)
1	2	2	3	2	2	3	13	13	8	13	11	7
2	5	5	6	9	9	14	22	22	14	26	26	14
3	6	2	3	5	5	5	15	15	9	16	11	9
4	4	4	3	5	5	4	12	12	7	11	10	7
5	1	1	1	1	1	1	6	6	4	4	4	2
6	1	1	5	4	4	4	12	15	8	11	12	9
7	3	2	3	2	2	3	7	7	5	9	9	7
8	1	1	1	2	2	2	9	8	5	4	4	3
9	3	3	3	2	2	3	13	13	7	11	10	6
10	1	1	1	4	4	3	7	7	5	8	7	5
11	1	1	1	1	1	1	6	6	4	5	5	3
12	1	1	1	1	1	1	4	4	2	2	2	2
14	1	1	1	1	1	1	2	2	2	2	2	2
15	1	1	1	1	1	1	4	4	2	1	1	1
16	1	1	1	1	1	1	7	7	4	5	5	3
17	2	3	2	2	2	2	11	11	7	7	7	5
18	1	1	2	3	3	2	5	5	4	7	6	4
19	1	2	3	3	3	5	10	11	5	11	11	7
20	2	2	2	2	1	2	11	15	6	9	9	6
21	2	2	2	3	3	3	8	8	5	9	9	4
22	2	2	3	6	6	3	7	7	5	8	8	6
23	6	6	5	9	9	5	12	12	7	11	11	8
24	1	1	2	4	4	3	7	7	3	5	5	3
25	2	2	2	3	3	2	6	7	4	4	4	4
Average	2.1250	2.0000	2.3750	3.1667	3.1250	3.0833	9.0000	9.3333	5.5000	8.2917	7.8750	5.2917
Standard Deviation	1.5965	1.3513	1.4084	2.2968	2.3278	2.6689	4.3539	4.6219	2.6540	5.3445	5.0589	2.9559

Table 4.5.5: Number of ρ -Swap Local Search iterations for the RAV regions.

4.5.3 Generated Instances

The RAV regions are suitable for an empirical analysis of the solution quality of the heuristics for realistic instances. However, due to the variations in the size of the RAVs, in the number of opened bases, and in attainable coverage (surplus capacity), it is difficult to single out the causes for the difference in performance. Therefore, we randomly generate a set of additional instances with common properties. The generation is motivated by the work of Vohra and Hall (1993), although our generation differs.

We denote the generated type of instances as 'Euclidean'. The Euclidean instances use the Euclidean distance between bases and points as travel time. The bases and points lie in the unit square $[0,1]^2$, where the coordinates are independently sampled from the Uniform distribution (with range (0,1)). For all instances we assume that the set of bases is equal to the set of points, similar to the RAV regions. Furthermore, the demand weights of the points are independently sampled from the Uniform distribution (with range (0,1)).

There are two sizes for the instances: 125 and 250 demand points, respectively. We generate 50 instances for each set, for a total of 100 'master' instances. For each of these 'master' instances we analyse the effect of increasing the number of bases to open and increasing the maximum attainable coverage. We compare opening 5, 10, and 15 bases and attaining 70%, 80%, 90%, and 100% maximum coverage. That is, we derive 12 MCLP instances from each 'master' instance for a total of 1200 MCLP instances. Since most RAV regions have full coverage and surplus capacity, we mimic this situation by having surplus capacity in the case of 100% coverage.

First, we set the number of bases to open to 5, 10, and 15 bases. Given the number of bases to open, we determine the minimum travel time threshold such that the desired maximum coverage is attained. In Appendix B we describe exactly how we determine the thresholds by using a bisection method. The exception is when the desired coverage is 100%, as these instances should have surplus capacity. We achieve this by setting the threshold such that full coverage can be attained with one base less (i.e., 4, 9, and 14 bases, respectively). We have chosen for a fixed surplus of one base as this is the rounded average surplus of the RAVs. The resulting travel time thresholds are shown in Table C.1.1.

To conclude, we have 24 classes of MCLP instances which are divided according to: size (125 or 250), number of opened bases (5, 10, or 15), and maximum attainable coverage (70%, 80%, 90%, or 100%). Each class has 50 instances for a total of 1200 generated MCLP instances.

Since many properties of the instances need to be specified for the generation, we have not included the generated instances in the sensitivity analysis of Chapter 3, as the results would be biased. However, with the chosen generation there is also a danger of creating Euclidean instances with a bias in favour of the Greedy Search. If the desired maximum coverage is low (e.g., 50%) the resulting travel time threshold can be so small that many points are only covered by themselves. In the extreme case, this leads to disjoint coverage, for which the Greedy Search is optimal. It is up to debate how many points can be isolated (covered only by itself) without an instance being biased. Up to 5% of all points should not cause any problems.

Almost all Euclidean instances have less than 5% isolated points, except for three classes of instances. These are Euclidean instances with 125 points and 70% coverage with 10 or 15 bases, or 80% coverage with 15 bases. Table 4.5.6 states the number of instances for which the number of isolated points lies within the shown ranges.

For instance, there are two Euclidean instances with 125 points, 10 bases, and 70% coverage that have between 6 and 10 isolated points. Primarily the set of Euclidean instances with 125 points, 15 bases, and 70% coverage raises concerns. When drawing conclusions, this class should be omitted in the reasoning to prevent any bias.

We note that the RAV regions have very few isolated points: RAV2 (Friesland) has two, RAV10 (Noord-Holland-Noord) only one, and RAV18 (Zuid-Holland-Zuid) also only one. Therefore, we have not mentioned the issue of isolated points before.

	Opened	Number of Instances withMaximumNumber of Isolated Points in Range					
Size	Bases	Coverage	[0,5]	(5, 10]	(10, 15]	(15,20]	$(20,\infty)$
125	10	70%	48	2	0	0	0
125	15	70%	12	30	6	2	0
125	15	80%	45	5	0	0	0

Table 4.5.6: Number of isolated demand points for the generated Euclidean instance classes. Instance classes where all instances have at most 5 isolated points are omitted.

4.5.4 Performance for Generated Instances

As we have 1200 different MCLP instances, displaying all individual results is impractical. Furthermore, the goal of the generated instances is to discern general properties of the solution methods, preferably agreeing to the observations for the RAV regions. In Appendix C the averages and standard deviations of the relative optimality gaps of all solution methods are given for each Euclidean instance class. The results include those of combined classes, e.g., all instances with 125 points and 70% coverage. Note that the combined results are not averages of the individual classes. For the Swap Local Search also the number of used iterations (until convergence) are given. Table 4.5.7 summarises the performance of each method.

Similar to the results for the 24 RAV regions, the ρ -Greedy Search outperforms the Reverse ρ -Greedy Search on average for the generated instances. The average relative optimality gap of the Reverse Greedy Search (10%) is five times as large as the gap of the Greedy Search (2%). The Anti Greedy Search outperforms its Revised variant (as these solutions should have a large optimality gap). This is expected, since the Revised Anti Greedy Search is only allowed to open non-adjacent bases. The average relative optimality gaps are 87% (Anti) and 68% (Revised Anti).

The detailed results of the Greedy variants are given in Tables C.2.1 to C.2.8. We will focus on the Greedy Search results, as it gives the best solutions of the Greedy variants. Table C.2.1 shows that the performance of the Greedy Search decreases as the attainable coverage increases: the average relative optimality gap for 90% maximum coverage is twice as large as that for 70%. Thus, it is more difficult to attain high coverage as deviations from the optimal selection of bases cannot be corrected during the search. However, when there is surplus capacity the performance of the Greedy Search improves again (deviations can be corrected). This explains the good results for the RAV regions, as most have surplus capacity. The Greedy Search performs worse for larger instances: the average relative optimality gap for instances with 125 points is 1.89%, whereas for instances with 250 points it is 2.41%. For most instance classes, having more bases to open also implies a better performance. For instance, consider the instance classes with 250 points and 80% coverage. The average relative optimality gaps are: 2.95% (5 bases), 2.81% (10 bases), and 2.56% (15 bases). This corresponds to the observation that the ability to correct for suboptimal selection of bases during the search is beneficial for the Greedy Search.

Comparing the 1- and 2-Greedy Search, we see that the 2-Greedy Search primarily improves the results for instances with 5 bases to open. For 125 demand points with 5 bases, the average relative gaps are 2.21% ($\rho = 1$) and 1.94% ($\rho = 2$). For 250 points with 5 bases the gaps are 2.45% and 2.12%, respectively. On average the 2-Greedy Search leads to gaps that are a 6.5% relative improvement of the 1-Greedy Search.

The results of the Reverse Greedy Search lead to similar conclusions as the Greedy Search, but there are two exceptions. First, we cannot discern a general pattern when increasing the attainable coverage from 70% to 90%. Second, the improvement of the Reverse 2-Greedy Search is insignificantly small.

For the RAV regions the Swap Local Search is robust with respect to the initial solution. The results in Table 4.5.7 show that the Swap Local Search is also robust for Euclidean instances. Consider the 1-Swap Local Search, the average relative optimality gaps are: 1.02% (Greedy), 1.37% (Reverse Greedy), 1.11% (Anti Greedy), and 1.18% (Revised Anti Greedy). For the 2-Swap Local Search the optimality gap are even smaller: all are approximately 0.20%.

Detailed results of the Swap Local Search are given in Tables C.3.1 to C.3.12. For all instance classes the Swap Local Search leads to a great improvement of the solution. For example, the average relative gap of the Greedy initial solutions is halved by the 1-Swap Local Search. The improvement for other initial solutions or for the 2-Swap Local Search is even greater. Extremely good performance is achieved for instances with 5 opened bases and 100% coverage: all are solved to optimality with 2-Swap Local Search (for most initial solutions).

In general, the optimality gap increases when the attainable coverage increases to 80% and 90%, and decreases again when there is surplus capacity. This observation holds for all eight initial solutions. To be more specific, the corresponding average relative gaps for the 1-Swap Local Search are approximately: 1.02% (70%), 1.44% (80%), 1.71% (90%), and 0.51% (100%). For the 2-Swap the gaps are: 0.11% (70%), 0.27% (80%), 0.37% (90%), and 0.07% (100%).

Furthermore, the gaps for instances with 250 demand points are generally larger than for instances with 125 points. For 1-Swap Local Search the average relative gaps are approximately 1.07% (125) and 1.27% (250). Similarly, for 2-Swap we have 0.16% (125) and 0.25% (250). When comparing instances with 5, 10, or 15 bases we could not discern a general pattern. The average required number of Swap iterations behaves in a similar manner as the average optimality gap. However, the iterations increase when there are more bases opened.

Taking both the optimality gaps and the required number of iterations into account, we come to the same conclusion as with the RAV regions with regards to the considered heuristics: we advise to apply the 1-Greedy Search followed by either the 1- or 2-Swap Local Search, depending on the available computational time. The resulting solution is expected to be of very high quality: on average a relative optimality gap of 1% (1-Swap) or 0.2% (2-Swap).

		Relative Op	Relative Optimality Gap		terations
Method	ρ	Average	Standard Deviation	Average	Standard Deviation
ρ -Greedy	1	0.0215	0.0171		
Reverse ρ -Greedy	1	0.1046	0.0103		
Anti e Greedy	$\frac{2}{1}$	$0.1041 \\ 0.8672$	0.0508 0.0381		
De la la train Carl	$\frac{2}{1}$	$0.8785 \\ 0.6836$	$0.0326 \\ 0.1125$		
Revised Anti ρ -Greedy	2	0.6914	0.1088	0 5017	1 5616
ρ -Swap Local Search after 1-Greedy	$\frac{1}{2}$	0.0102	0.0120	3.0342	1.3010 1.4691
ho-Swap Local Search after 2-Greedy	1	0.0101	0.0118	2.4242	1.5560
ρ -Swap Local Search after Reverse 1-Greedy	$\frac{1}{2}$	$0.0137 \\ 0.0022$	$0.0153 \\ 0.0050$	7.3233 5.3533	$3.1962 \\ 2.1317$
ρ -Swap Local Search after Reverse 2-Greedy	1	0.0137	0.0153	7.3000	3.1791
ρ -Swap Local Search after Anti 1-Greedy	$\frac{1}{2}$	$0.0113 \\ 0.0020$	$0.0134 \\ 0.0051$	$\frac{12.2583}{7.8025}$	$4.4508 \\ 2.8662$
ρ -Swap Local Search after Anti 2-Greedy	1	0.0108	0.0129	12.2625	4.4681
ρ -Swap Local Search after Revised Anti 1-Greedy	$\frac{1}{2}$	$0.0118 \\ 0.0021$	$0.0129 \\ 0.0048$	$\frac{12.0275}{7.6225}$	$\begin{array}{c} 4.4181 \\ 2.7970 \end{array}$
$\rho\text{-}\mathrm{Swap}$ Local Search after Revised Anti 2-Greedy	1	0.0119	0.0130	12.0717	4.4224

Table 4.5.7: Performance of the heuristic solution methods.

4.6 Two Dutch Regional Ambulance Services in Detail

As mentioned previously, we discuss the heuristic solutions in detail for two RAV regions. The two regions are RAV14 (Gooi- en Vechtstreek) and RAV23 (Limburg-Noord), the same regions as in the sensitivity analysis in Section 3.5. RAV14 can be solved to optimality with many heuristic solution methods, as indicated by the results in Section 4.5. In particular, applying 1-Swap Local Search is sufficient to find an optimal solution for all evaluated initial solutions. This is not the case for RAV23, which is more difficult to solve to optimality by the heuristics. For example, the Greedy Search has the worst performance for RAV23, compared to the other RAV regions. However, applying the 2-Swap Local Search afterwards leads to an optimal solution.

RAV14 and RAV23 are highlighted in Figure 4.6.1. For the detailed information of the two considered RAVs we refer to Section 3.5. The heuristic solutions of RAV14 are discussed in Section 4.6.1 and those for RAV23 in Section 4.6.2.



(a) Gooi- en Vechtstreek (RAV14). (b) Limburg-Noord (RAV23).

Figure 4.6.1: RAV14 and RAV23 of The Netherlands.

4.6.1 Gooi- en Vechtstreek

Recall that RAV14 has 16 optimal solutions with full coverage if only two bases are opened. Consequently, many more optimal solutions exist if three bases are available. As the region does not contain many isolated demand points, it can be expected that the Greedy Search performs well. Figure 4.6.2a shows which bases are opened by the Greedy Search and in what order. Applying 1- or 2-Greedy Search results in the same base locations. The Greedy solution has full coverage and is therefore optimal. In fact, the first two bases opened by the Greedy Search are sufficient for complete coverage (recall Figure 3.5.5).

The Reverse Greedy Search solution is given in Figure 4.6.2b. Again, there is no difference between Reverse 1- and 2-Greedy. Note that the behaviour of the Reverse Greedy Search is somewhat arbitrary. As Reverse Greedy iteratively closes bases, there is no natural order in the remaining set of opened bases. Therefore, we do not number the bases. The Reverse Greedy solution is optimal, but it does require all three bases. Similar to the Greedy Search, there is no difference in using search parameter $\rho = 1$ or $\rho = 2$ for the Anti ρ -Greedy and the Revised Anti ρ -Greedy Search. By design, these methods try to construct a solution with minimal coverage. For RAV14 the Anti Greedy Search leads to a clustered set of opened bases, see Figure 4.6.2c. The Anti Greedy Search opens the first base in Breukeleveen, the smallest and most isolated demand point. The other bases are clustered nearby in order to minimise the coverage. The base locations of the Revised Anti Greedy Search are more spread out, as shown in Figure 4.6.2d.

By repositioning the third base opened by the Anti Greedy Search to the north, an optimal solution can be constructed. This is exactly the swap performed by the Swap Local Search, see Figure 4.6.2e. Similarly, an optimal solution can be constructed by repositioning the third base opened by the Revised Anti Greedy Search. The result of the Swap Local Search is given in Figure 4.6.2f.

To conclude, RAV14 has a surplus capacity of one base and one centrally located base covers almost the entire region. Furthermore, there are few isolated demand points (e.g., Breukeleveen in the south). As a result, the (Reverse) Greedy Search and the Swap Local Search (with any of the evaluated initial solutions) are optimal for RAV14.



(e) Swap Local Search after Anti Greedy.

(f) Swap Local Search after Revised Anti Greedy.

Figure 4.6.2: Heuristic base locations (blue) and covered (green) or uncovered (red) demand points for RAV14.

4.6.2 Limburg-Noord

None of the Greedy construction methods are able to construct an optimal solution for RAV23. In fact, the relative optimality gap of the 1- and 2-Greedy Search solution is 2.71%, which is the largest of all RAV regions. The reason for this relatively bad performance can easily be determined from Figure 4.6.3a, which shows the order of the opened bases by the Greedy Search. The second base has become superfluous due to bases 4, 5, and 7. This problem occurs for both the 1- and 2-Greedy Search, which consequently have the same solution.

To improve the Greedy Search solution the second base should be repositioned more to the north. This swap is indeed performed by the 1- and 2-Swap Local Search, see Figures 4.6.3b and 4.6.3c. After applying minor adjustments to the southern bases (4, 5, and 7), the 1-Swap Local Search converges to a 1-Swap local maximum. Further improvement can be achieved by placing base 3 more to the west and base 1 more north. However, the intermediate solution would have a lower coverage and such swaps are not allowed⁷ by the 1-Swap Local Search. The 2-Swap Local Search is allowed to swap these bases and the resulting solution is optimal.

Even though the behaviour of the Reverse 1- and 2-Greedy Search is somewhat arbitrary, the solution resembles that of the Greedy Search. Both Reverse Greedy methods give the same solution, see Figure 4.6.3d. The corresponding 1- and 2-Swap Local Search solutions are given in Figures 4.6.3e and 4.6.3f, respectively. The performed swaps are similar to those applied to the Greedy solution, except that two simultaneous swaps are insufficient to construct the optimal solution.

The Anti 1- and 2-Greedy Search construct the same solution, which is depicted in Figure 4.6.4a. All bases are clustered in the isolated region in the north. Consequently, the 1-Swap Local Search starts by spreading the bases, almost in a 1-Greedy Search way. This is followed by the usual 1-Swap neighbourhood search. The required number of iterations in Table 4.5.5 confirm this statement: the 1-Swap Local Search needs 12 iterations for the Anti Greedy solution and 6 for the Greedy solution. Do note that base 7 of the Anti Greedy solution is already positioned at the optimal location. This also explains the similar average performance of the Swap Local Search with either the Greedy or Anti Greedy Search as initial solution.

The major difference of the Revised Anti Greedy solution is that the bases are not clustered in an isolated region (Figure 4.6.4d), which alters the performed swaps by the Swap Local Search. Consequently, using the Revised Anti Greedy Search as initial solution for the 1-Swap Local Search results in the worst optimality gap of all initial solutions. The Swap Local Search solutions are given in Figures 4.6.4e and 4.6.4f.

We conclude that the Swap Local Search is robust with respect to the initial solution, in particular if a 2-Swap neighbourhood is used. Furthermore, the 1- and 2-Swap Local Search solutions have many base locations in common.

 $^{^7\}mathrm{This}$ is an example where the 1-Swap Local Search can be improved by temporarily allowing swaps that lead to worse coverage.



Figure 4.6.3: Heuristic base locations (blue) and covered (green) or uncovered (red) demand points for RAV23 (1 of 2).



Figure 4.6.4: Heuristic base locations (blue) and covered (green) or uncovered (red) demand points for RAV23 (2 of 2).

4.7 Conclusion

From a worst-case point of view, most emergency medical service (EMS) optimisation models cannot be solved to optimality efficiently. This seems to contradict their use to evaluate multiple scenarios within reasonable time. However, the required computational time for realistic problem instances is usually manageable. In particular, the Maximal Covering Location problem (MCLP) instances based on the 24 Regional Ambulance Services of The Netherlands can be solved to optimality within seconds (with standard hardware).

We do note that the MCLP is one the most basic EMS facility location models. The computational time will increase as the model complexity increases. Independent of the EMS model, solving very large instances to optimality will require an impractical amount of computational time. One is then forced to use approximation algorithms or heuristic solution methods.

For the MCLP we have considered two heuristic solution methods: the Greedy Search and the Swap Local Search. By using a framework of submodular functions, we can give performance guarantees for the two methods. We have constructed families of worst-case instances for both methods, proving that these bounds are tight: the maximum relative optimality gap is e^{-1} for 1-Greedy Search and 1/2 for ρ -Swap Local Search. In fact, the guarantee for the Swap Local Search follows directly from our constructive proof, which explicitly derives the worst-case MCLP instances. These worst-case instances have a certain symmetry. It is therefore interesting to investigate the effect to the worst-case performance of applying small perturbations to this symmetry.

We have numerically evaluated the Greedy Search and the Swap Local Search for the MCLP, using realistic and randomly generated instances. The empirical performance far exceeds the guarantees for both methods: the Greedy Search has an average relative optimality gap of 2%, and the Swap Local Search 1% (1-Swap neighbourhood) or 0.2% (2-Swap neighbourhood). Considering the evaluated methods and parameters (including different initial solutions), we conclude that the 1-Greedy Search followed by either the 1- or 2-Swap Local Search is preferable, depending on the available computational time.

Of course, there are many more heuristic solution methods possible for the MCLP, in particular methods based on the Linear Programming relaxation. A solution method (or framework) that can be applied to multiple EMS models (e.g., extensions of the MCLP) would be a natural choice, as one can then easily switch between models.

Chapter 5

Conclusions and Future Work

Contents

5.1 Con	nclusions	
5.2 Fut	ure Work	
5.2.1	Numerical Analysis	
5.2.2	Other EMS Models	
5.2.3	Other Solution Methods	

5.1 Conclusions

The first facility location optimisation models applied to emergency medical service (EMS) only incorporate the most basic aspects of EMS: demand points (origins of emergency calls), possible ambulance base locations, and coverage. The models are static: all parameters are deterministic and any uncertainty is not (explicitly) included. The coverage parameters satisfy an 'all-ornothing' relation: a demand point is either covered (reachable in time) or uncovered by a base. These basic models capture the essence of facility location, but lack more realistic features.

In the past forty years, additional features have been added to existing EMS models in order to include desired aspects of EMS planning. Examples are the inclusion of: regional coverage constraints, backup coverage, survival objective functions, busy fractions of ambulances, reliability of service, stochastic response times, multiple vehicle types, and multiple time periods.

We have considered the Maximal Covering Location problem (MCLP), one of the basic EMS models, and applied it to the 24 Regional Ambulance Services of The Netherlands. In 21 regions full coverage can be attained according to the MCLP. The other three regions have near-complete coverage (around 99.5% covered). The objective function of the MCLP only considers the covered demand. Consequently, the regions have many optimal MCLP solutions. Although all these base locations are 'optimal', some are more preferred than others.

For example, the coverage of an optimal solution can be more sensitive to changes in the travel time threshold (the adjacency parameters). Selecting a more robust optimal solution would be preferred in practice. This can be achieved by modifying the objective function or by applying a multi-stage optimisation procedure. We have chosen for a two-stage approach: in the first stage the MCLP is solved, in the second stage the most robust optimal MCLP solution is selected. To be more specific, the second stage minimises the maximum travel time between the demand points and their nearest opened bases.

This example shows but one of many ways to modify the MCLP. Another modification is the Robust Counterpart MCLP, which determines an MCLP solution that is robust to perturbations in demand. That is, the Robust Counterpart optimum has the best coverage in case of a worst-case realisation of demand.

To evaluate the robustness of the MCLP with respect to uncertain demand, we have considered its Robust Counterpart with two types of perturbations. The first type is a fixed relative perturbation, e.g., the demand can deviate 5% from its estimated value. The second type relates to Poisson arrivals and uses a perturbation equal to a multiple of the standard deviation of the estimated demand. For the first type both MCLP models are equivalent. For the second type both models give the same solution for almost all evaluated instances. If the robust and non-robust solutions do differ, the difference in coverage is insignificantly small.

The considered MCLP instances could be solved to optimality within seconds (using standard hardware and commercial software). Such computational efficiency cannot be guaranteed in general. It is therefore useful to analyse the performance of heuristic solution methods. We have focussed on two heuristics: the Greedy Search and the Swap Local Search. Both solution methods have been analysed theoretically and numerically, where we have used realistic and randomly generated MCLP instances.

The Greedy Search constructs a feasible solution by iteratively opening bases in a greedy and myopic way. A tight performance guarantee can be derived for the 1-Greedy Search, which turns out to be the best possible guarantee for efficient solution methods: the relative optimality gap is at most e^{-1} . Furthermore, the method achieves an average relative optimality gap of 2% for the generated instances.

The Swap Local Search tries to improve an initial solution by iteratively closing and opening bases. We have proved that any final Swap Local Search solution has a relative optimality gap of at most 50%. From the numerical analysis we can conclude that the Swap Local Search is robust with respect to the initial solution. The average relative optimality gap for the generated instances is 1% (1-Swap neighbourhood) or 0.2% (2-Swap neighbourhood), much smaller than the theoretical guarantee.

Considering the evaluated parameters of the Greedy Search and the Swap Local Search, we advise to use 1-Greedy Search followed by either the 1- or 2-Swap Local Search. The 2-Greedy Search leads to insignificant improvements, while requiring additional computational time. Of course, any decision support tool is not limited to these two methods: other (more advanced) heuristic solution methods exist (e.g., based on the Linear Programming relaxation).

To conclude, we have discussed several EMS optimisation models, each incorporating a different set of features. We have focussed on the MCLP and noticed that care should be taken when using the MCLP for the Regional Ambulance Services in The Netherlands. For 'optimal' base locations a two-stage optimisation approach should be used, or a more complex model has to be considered. Nevertheless, the MCLP can be used as a preliminary model to quickly assess potentially uncovered regions, allowing more advanced models to focus on smaller regions.

5.2 Future Work

In the previous chapters we have limited the analysis and made certain assumptions. With these assumptions in mind, we suggest future research in Sections 5.2.1 and 5.2.2. Moreover, there are other solution approaches that could lead to approximation algorithms for the MCLP. In Section 5.2.3 we discuss additional heuristics, Linear Programming relaxation approaches, and Primal-Dual methods.

5.2.1 Numerical Analysis

For the numerical results, we have used MCLP instances based on the 24 Regional Ambulance Services of The Netherlands. The regions are discretised and aggregated into 4-digit postal code areas. This aggregation leads to a loss in accuracy in travel times and therefore in attainable coverage. For The Netherlands, 6-digit postal code areas would provide more than sufficient accuracy, as these areas are relatively small. However, more detailed regions also require more (estimated) data. Furthermore, the size of the MCLP instances (and thus the computational time) increases significantly. In The Netherlands, there are approximately 4000 4-digit postal codes, whereas there are more than 400 000 6-digit areas. That is, a '6-digit' MCLP instance has on average 100 times more demand points than a '4-digit' instance.

We have assumed that all considered postal codes are suitable for an ambulance base. However, recall that the provided data does not contain postal code areas without any inhabitants. Restrictions on the possible base locations can be researched, in particular in combination with 6-digit postal code areas.

Using the number of inhabitants as objective coefficients for the MCLP is common in EMS literature. It would be interesting to see if historical data on the emergency calls can provide additional insights, perhaps leading to modifications to the objective coefficients. This can also affect the demand uncertainty structure for the Robust Counterpart MCLP.

In general, whether the MCLP is robust to demand perturbations can be determined for other MCLP instances and other uncertainty structures. Furthermore, we have only considered a worst-case approach (the Robust Counterpart) of Ben-Tal, El Ghaoui, et al. (2009), but other robust optimisation approaches exist (see Bertsimas et al. (2011)).

The performance of the Greedy Search and the Swap Local Search can be further evaluated by using differently generated instances, for example by introducing spatial patterns (e.g., clusters) or by using non-Euclidean travel times. In particular, very large instances are of interest, as exact solution methods would require an unreasonable amount of computational time.

5.2.2 Other EMS Models

The overview of EMS models is not complete, as we have focussed on coverage models. Other types of EMS models include derivatives of the *p*-median and *p*-centre models. Furthermore, the Facility Location problem is applied to other (non-medical) research areas as well. Advancements in those areas can be useful for EMS models.

We have primarily discussed static optimisation models for strategic or tactical decision support. On the operational level dynamic ambulance management is becoming more relevant. Dynamic ambulance management (DAM) models explicitly include the ability to reposition ambulances in real time, see for instance Gendreau et al. (2001) and Restrepo (2008). Thus, conditional ambulance relocation rules can be considered. The models with multiple time periods and Stochastic Programming can be seen as a step towards dynamic management. Do note that DAM models (operational level) can be used to support decisions on a strategic or tactical level.

An extension of the performed analyses is to use a more complex EMS model (if possible). In particular, the Maximum Expected Covering Location problem (MEXCLP) by Daskin (1983) is an appropriate extension, as it adds the assignment and unavailability of ambulances to the MCLP. For easy reference, we have restated the MEXCLP in Model 5.2.1. For example, the Robust Counterpart of the MEXCLP is already mentioned in Section 3.2.5. Note that properties such as in Section 3.2.4.1 do not need to hold for the Robust Counterpart of the MEXCLP, since the new robust decision variables $(u, v_j^+, \text{ and } v_j^-)$ are not necessarily binary.

Maximise	
$\sum_{j \in \mathcal{J}} d_j \sum_{k=1}^q ($	$(1-\rho)\rho^{k-1}z_j^k$
subject to	
$\sum_{i \in \mathcal{I}} x_i \le p,$	
$\sum_{i\in\mathcal{I}}y_i\leq q,$	
$y_i \le q_i x_i$	$\forall i \in \mathcal{I},$
$\sum_{i \in \mathcal{I}} a_{ij} y_i \ge \sum_{k=1}^q z_j^k$	$\forall j \in \mathcal{J},$
$x_i \in \mathbb{B}$	$\forall i \in \mathcal{I},$
$y_i \in \mathbb{N}$	$\forall i \in \mathcal{I},$
$z_j^k \in \mathbb{B}$	$\forall j \in \mathcal{J}, k \in \{1, \dots, q\}.$

Model 5.2.1: Maximum Expected Covering Location problem (MEXCLP).

If the MEXCLP can be formulated as a Maximum k-Coverage problem with a non-decreasing submodular function, the performance guarantees of the Greedy Search (Section 4.3.1) and the Swap Local Search (Section 4.4.1) for general submodular functions can be used. Recall that the Maximum k-Coverage problem for $k \in \mathbb{N}$ is given by the optimisation problem

$$\max\left\{\phi(\mathcal{U}): |\mathcal{U}| \le k, \mathcal{U} \subseteq \mathcal{N}\right\},\$$

where $\phi: 2^{\mathcal{N}} \to \mathbb{R}$ is a non-decreasing submodular function and \mathcal{N} a finite discrete set.

Consider the following transformation of the MEXCLP to the context of submodular functions. Set k = q (the number of available ambulances) and define the finite set \mathcal{N} as

$$\mathcal{N} = \{i_1, \ldots, i_{q_i} : i \in \mathcal{I}\}.$$

Any subset $\mathcal{U} \subseteq \mathcal{N}$ corresponds to the MEXCLP solution where at each base $i \in \mathcal{I}$ exactly $|\{i_1, \ldots, i_{q_i}\} \cap \mathcal{U}|$ ambulances are stationed. A base is opened if at least one ambulance is assigned to it. Define $\phi : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ such that it corresponds to the resulting coverage. This is a non-decreasing submodular function.

The only flaw in this formulation is that more than p bases can be opened (which is relevant if p < q). A possible solution is to incorporate a Branch-and-Bound procedure or some variant of Lagrangian relaxation. However, this complicates the analysis for performance guarantees.

Other approaches are also possible: for example, one can set $\phi(\mathcal{U}) = 0$ if more than p bases are opened with \mathcal{U} . Unfortunately, ϕ is non-monotone in this case and we cannot use the discussed guarantees. An applicable performance guarantee is given in Nemhauser, Wolsey, and Fisher (1978) for the 1-Greedy Search. Another example is to set $\phi(\mathcal{U}) = 0$ if more than p bases are opened or more than q ambulances are used. In this case we do not need the capacity constraint, i.e., we are maximising an unconstrained non-monotone submodular function. See Feige et al. (2007) for a constant factor approximation algorithm (a Local Search algorithm).

Modifying the objective function seems to be a crude way to implement the MEXCLP constraints, as it leads to non-monotone submodular functions. Perhaps the use of additional knapsack or matroid constraints (see J. Lee et al. (2009)) or other definitions of \mathcal{N} can be successful. Do note that a modified Maximum k-Coverage problem would most likely require different heuristic solution methods. That is, the Greedy Search and Swap Local Search frameworks have to be adjusted. Consequently, the discussed performance guarantees are not applicable.

5.2.3 Other Solution Methods

In Section 4.6 we have seen that bases opened by a Greedy Search iteration can become redundant in later iterations. To improve the Greedy Search, we can use a hybrid 'Greedy Swap Search': if the coverage of some bases overlap significantly after opening an additional base, apply a certain number of Swap Local Search iterations to the bases in question. If a base is redundant, the Swap Local Search iteration will relocate it.

We have also noticed that the 1-Swap Local Search can be improved if it is sometimes allowed to temporarily perform swaps that result in worse coverage. Such a modification is often used in Local Search methods to 'escape' from local extrema. In particular, the suggested modification resembles Tabu Search, see also Glover and Laguna (1997).

These modifications lead to the field of meta-heuristics (see Voß (2001) and Voß et al. (1999)): advanced solution methods that combine and guide several heuristics. Meta-heuristics have been successfully applied to several research areas (e.g., routing and scheduling). We do not go into the details on meta-heuristics, instead we will discuss Linear Programming and Primal-Dual approaches in the next sections. There are strong indications that rounding schemes for Linear Programming solutions have good performance.

5.2.3.1 Linear Programming Relaxation and Rounding

The Linear Programming (LP) relaxation of a (Mixed) Integer Programming model relaxes the integrality constraints on the decision variables, allowing the variables to take on non-integral values. As the resulting model is an LP model, it can be solved efficiently (in polynomial time). The LP relaxation of the MCLP is given in Model 5.2.2. Note that requiring $x_i \in \mathbb{R}_{\geq 0}$ is sufficient.

$$\sum_{j \in \mathcal{J}} d_j z_j$$

subject to

Maximise

$$\sum_{i \in \mathcal{I}} x_i \leq p,$$

$$\sum_{i \in \mathcal{I}} a_{ij} x_i \geq z_j \qquad \forall j \in \mathcal{J},$$

$$x_i \in [0, 1] \qquad \forall i \in \mathcal{I},$$

$$z_j \in [0, 1] \qquad \forall j \in \mathcal{J}.$$

Model 5.2.2: Linear Programming relaxation of the MCLP.

Consider an arbitrary MCLP instance, denoted by $\omega \in \Omega$. Let $\theta^{LP}(\omega)$ be the maximum of the LP relaxation and $\theta^*(\omega)$ the maximum for the (integer) MCLP. The LP provides¹ an upper bound for the MCLP: it holds that $\theta^{LP}(\omega) \geq \theta^*(\omega)$. Therefore, we can consider the gap $\theta^{LP}(\omega)/\theta^*(\omega)$ between the two maxima, called the integrality gap of the instance. Likewise,

$$\sup_{\omega\in\Omega}\left\{\frac{\theta^{LP}(\omega)}{\theta^*(\omega)}\right\}$$

is the integrality gap for the MCLP (in general). It is similar to the locality gap of the Swap Local Search.

The optimal LP solution x^{LP} can be used to construct feasible solutions for the MCLP (e.g., see Williamson and Shmoys (2011)). For example, we can use the Greedy Search in the following way. If x^{LP} is not integer, then more than p bases are 'opened': $|\{x_i^{LP} > 0 : i \in \mathcal{I}\}| > p$. We can apply the ρ -Greedy Search on the set of bases with non-integer x_i^{LP} (the set $\{i \in \mathcal{I} : x_i^{LP} \in (0, 1)\}$) to construct an integer solution. Let θ^H be the resulting coverage. Using Proposition 4.3.2 we can show that

$$\theta^H \ge \frac{\rho}{|\{i \in \mathcal{I} : x_i^{LP} \in (0,1)\}|} \theta^*,$$

where $1 \le \rho \le p - |\{i \in \mathcal{I} : x_i^{LP} = 1\}| < |\{i \in \mathcal{I} : x_i^{LP} \in (0, 1)\}|$. The performance guarantee goes to zero if there are many non-integer x_i^{LP} . Note that possibly a better guarantee exists.

¹For some optimisation problems there always exists an integer optimal solution for the LP relaxation, e.g., if the system of constraints is totally dual integral. That is, the LP relaxation directly gives an optimal solution for the original problem. This is not the case for the MCLP.

Ageev and Sviridenko (1999) give an approximation algorithm that rounds the LP solution with the best possible performance guarantee. Their rounding algorithm is as follows. Let $i_1, i_2 \in \{i \in \mathcal{I} : x_{i_1}^{LP} \in \{0, 1\}\}$ be two arbitrary bases with non-integer LP values. Redistribute the total value $x_{i_1}^{LP} + x_{i_2}^{LP}$ among x_{i_1} and x_{i_2} such that at least one is integer. As there could be multiple ways to achieve this, the one leading to the highest coverage is chosen. The resulting solution is feasible for the LP and has a smaller number of non-integer values. The procedure is repeated until a feasible MCLP solution has been constructed, which requires polynomial time in total.

It can be shown that the final coverage θ^R of the rounding procedure satisfies

$$\frac{\theta^* - \theta^R}{\theta^*} \le \left(\frac{k-1}{k}\right)^k,$$

where $k \in \mathbb{N}$ is the maximum number of adjacent bases:

$$k = \max\left\{\sum_{i\in\mathcal{I}}a_{ij}: j\in\mathcal{J}\right\}.$$

We refer to Ageev and Sviridenko (1999) for the proof. Recall that Theorem 4.1.1 implies that this is the best possible performance guarantee in polynomial time, unless P = NP. In fact, the proof actually shows that

$$\theta^R \ge \left(1 - \left(\frac{k-1}{k}\right)^k\right) \theta^{LP}.$$

Since $\theta^* \ge \theta^R$, we have a bound for the integrality gap:

$$\frac{\theta^{LP}}{\theta^*} \le \left(1 - \left(\frac{k-1}{k}\right)^k\right)^{-1} \le \left(\frac{e}{e-1}\right).$$

A family of MCLP instances exists for which the integrality gap of the instance is arbitrarily close this bound (see Ageev and Sviridenko (1999)). Thus, the integrality gap for the MCLP is e/(e-1). Consequently, the described rounding approximation algorithm is the best possible rounding procedure (with respect to worst-case performance). Of course, the average performance (including required computational time) for realistic instances is often more important. Future research can focus on developing rounding procedures with better average performance.

Vohra and Hall (1993) discuss the expected integrality gap for three types of randomly generated MCLP instances. One type corresponds to the generated instances in Section 4.5.3. They show that the expected relative gap between the LP solution and the optimal MCLP solution approaches zero as the dimension of the instance increases. For our evaluated instances the LP solution was integer in most cases. If the LP solution was non-integer, the integrality gap was small. This resulted in few required Branch-and-Bound iterations to solve the MCLP instances to optimality. Thus, the LP relaxation should be very useful for (heuristic) solution methods.

However, we have noticed that generated instances where the adjacency parameters are sampled from the Bernoulli distribution require significantly more Branch-and-Bound iterations. Perhaps this type of instances has more non-integer optimal LP solutions in general. Finally, Integer Programming models often have weaker and stronger LP relaxations. That is, stronger LP formulations have tighter constraints and therefore most likely a smaller integrality gap. Furthermore, an optimisation problem can be translated to multiple Integer Programming models. For example, the MCLP can be formulated as a maximisation or minimisation problem (recall Model 2.2.2). This can affect the quality of the resulting LP relaxations. Any LP relaxation can be further strengthened by adding valid inequalities (constraints). For example, cutting planes are used in Branch-and-Cut methods to iteratively make a non-integer optimal LP solution infeasible.

We have introduced rounding procedures for the LP relaxation for the MCLP and discussed two examples. For future work we suggest to perform an extensive literature survey and analyse the performance of various LP based solution methods. From our numerical results we have obtained strong indications that such methods would have high performance.

5.2.3.2 Primal-Dual Approximations

The Primal-Dual solution method is typically exemplified by an approximation algorithm for the Minimum Weight Vertex Cover problem (see Williamson and Shmoys (2011)). We will mimic this typical example for the MCLP by using a modified formulation, see Model 5.2.3. The decision variable $\bar{z}_j \in \mathbb{B}$ is defined as $\bar{z}_j = 1 - z_j$ for all $j \in \mathcal{J}$, i.e., $\bar{z}_j = 1$ if demand point $j \in \mathcal{J}$ is uncovered. Naturally, the restriction on the number of bases (Equation (5.2.1)) makes the model non-trivial. Suppose we relax this restriction using Lagrangian relaxation. That is, we drop constraint (5.2.1) and 'add' it to the objective function:

$$\sum_{j\in\mathcal{J}}d_j - \sum_{j\in\mathcal{J}}d_j\bar{z}_j + w\left(p - \sum_{i\in\mathcal{I}}x_i\right) = \sum_{j\in\mathcal{J}}d_j + pw - w\sum_{i\in\mathcal{I}}x_i - \sum_{j\in\mathcal{J}}d_j\bar{z}_j,$$

for some fixed weight $w \in \mathbb{R}_{\geq 0}$. The parameter w is called the Lagrange multiplier and it can be seen as the penalty for the violation of (5.2.1). The Lagrangian MCLP relaxation provides an upper bound on the optimal MCLP coverage for any $w \in \mathbb{R}_{\geq 0}$. This implies that if the Lagrangian optimum opens exactly p bases, it is also the optimal MCLP solution. Note that in the extreme case either all bases (w = 0) or no bases ($w \to \infty$) are opened by the Lagrangian optimum.

> Maximise $\sum_{j \in \mathcal{J}} d_j - \sum_{j \in \mathcal{J}} d_j \bar{z}_j$ subject to $\sum_{i \in \mathcal{I}} x_i \le p, \qquad (5.2.1)$ $\sum_{i \in \mathcal{I}} a_{ij} x_i + \bar{z}_j \ge 1 \qquad \forall j \in \mathcal{J},$ $x_i \in \mathbb{B} \qquad \forall i \in \mathcal{I},$ $\bar{z}_j \in \mathbb{B} \qquad \forall j \in \mathcal{J}.$

Model 5.2.3: Modified formulation for the Maximal Covering Location problem.

Model 5.2.4 states the (primal) LP relaxation of the Lagrangian MCLP relaxation and its dual. Note that we have rescaled the dual variables v_j , implicitly assuming that $d_j > 0$ for all $j \in \mathcal{J}$. We will use these two LP models to illustrate Primal-Dual methods.

MaximiseMinimise
$$\sum_{j \in \mathcal{J}} d_j + pw - w \sum_{i \in \mathcal{I}} x_i - \sum_{j \in \mathcal{J}} d_j \bar{z}_j$$
Minimisesubject to
$$\sum_{j \in \mathcal{J}} d_j + pw - \sum_{j \in \mathcal{J}} d_j v_j$$
subject tosubject to
$$\sum_{i \in \mathcal{I}} a_{ij} x_i + \bar{z}_j \ge 1$$
 $\forall j \in \mathcal{J},$
$$x_i \in \mathbb{R}_{\ge 0}$$
 $\forall i \in \mathcal{I},$
$$z_j \in \mathbb{R}_{\ge 0}$$
 $\forall j \in \mathcal{J}.$

Model 5.2.4: Primal and dual LP of the Lagrangian MCLP relaxation.

Primal-Dual approximation methods use the complementary slackness conditions of the LP to construct an integer MCLP solution. The complementary slackness conditions are:

$$\left(\sum_{i\in\mathcal{I}}a_{ij}x_i+\bar{z}_j-1\right)v_j=0\qquad\qquad\forall j\in\mathcal{J},\qquad(5.2.2)$$

$$\left(\sum_{j\in\mathcal{J}}a_{ij}d_jv_j - w\right)x_i = 0 \qquad \forall i\in\mathcal{I},$$
(5.2.3)

$$(v_j - 1)\bar{z}_j = 0 \qquad \forall j \in \mathcal{J}.$$
(5.2.4)

If we have a feasible pair of primal and dual LP solutions that satisfy all complementary slackness conditions, then these are optimal LP solutions. Primal-Dual methods typically ignore conditions related to the dual variables, i.e., Equation (5.2.2). For given dual variables v_j the method constructs an integer MCLP solution² that satisfies Equations (5.2.3) and (5.2.4). That is, for opened bases ($x_i = 1$) it must hold that

$$\sum_{j \in \mathcal{J}} a_{ij} d_j v_j = w$$

Likewise, for uncovered demand points $(\bar{z}_j = 1)$ we have $v_j = 1$.

Let θ_p^{L*} and θ_d^{L*} be the optimal values of primal and dual LP of the Lagrangian relaxation, respectively. By definition, we have $\theta^* \leq \theta_p^{L*} \leq \theta_d^{L*}$, where θ^* is the maximum of the MCLP. Suppose a Primal-Dual method constructs a feasible MCLP solution with coverage θ^{PD} . If $\theta^{PD} \geq \alpha \theta_d^{L*}$ for some $\alpha \in [0, 1]$, then we have an α -guarantee for the performance of the Primal-Dual method:

$$\theta^{PD} \ge \alpha \theta_d^{L*} \ge \alpha \theta_p^{L*} \ge \alpha \theta^*.$$

²Since we have relaxed Equation (5.2.1) the constructed solution can be infeasible for the MCLP. This issue will be discussed below.

Consider the following Primal-Dual algorithm. Start with $v_j = 0$ for all $j \in \mathcal{J}$, which is a feasible dual LP solution. Next, uniformly increase all v_j , whilst maintaining dual LP feasibility. If during this procedure $\sum_{j \in \mathcal{J}} a_{ij} d_j v_j = w$ holds for some $i \in \mathcal{I}$, we fix all corresponding v_j $(j \in \{j' \in \mathcal{J} : a_{ij'} = 1\})$. This ensures feasibility of v_j for the dual LP. Eventually, no v_j can be increased further (note that $v_j \leq 1$).

Construct the following integer solution: set $x_i = 1$ if $\sum_{j \in \mathcal{J}} a_{ij} d_j v_j = w$, and $x_i = 0$ otherwise. Let $\bar{z}_j \in \mathbb{B}$ correspond to the resulting coverage of demand points. Notice that by construction we have $v_j = 1$ if $\bar{z}_j = 1$ (but not necessarily vice versa). These constructed properties (i.e., some complementary slackness conditions) are often used to derive performance guarantees for the considered Primal-Dual method. For example, we have the following (not so useful) bound:

$$\sum_{i \in \mathcal{I}} wx_i + \sum_{j \in \mathcal{J}} d_j \bar{z}_j = \sum_{i \in \mathcal{I}} \left(\sum_{j \in \mathcal{J}} a_{ij} d_j v_j \right) x_i + \sum_{j \in \mathcal{J}} d_j v_j \bar{z}_j$$
$$= \sum_{j \in \mathcal{J}} \left(\sum_{i \in \mathcal{I}} a_{ij} x_i + \bar{z}_j \right) d_j v_j$$
$$\leq \left(\max\left\{ 1, \min\left\{ \sum_{i \in \mathcal{I}} x_i, \max\left\{ \sum_{i \in \mathcal{I}} a_{ij} : j \in \mathcal{J} \right\} \right\} \right\} \right) \sum_{j \in \mathcal{J}} d_j v_j.$$

We cannot directly use this bound for a performance guarantee for our Primal-Dual method: the bound excludes the constant $\sum_{j \in \mathcal{J}} d_j + pw$ of the LP objectives. Furthermore, the procedure can open more than p bases (an infeasible MCLP solution). By adjusting the Lagrange multiplier $w \in \mathbb{R}_{\geq 0}$ we could be fortunate to get a feasible MCLP solution with p opened bases. However, this cannot be guaranteed.

There are Primal-Dual solution methods for similar optimisation problems for which performance guarantees can be shown. In particular, the Metric *p*-Median problem has an approximation algorithm, see Model 5.2.5. Each demand point is assigned to at least one opened base (variables u_{ij}), which costs $c_{ij} \in \mathbb{R}_{>0}$.

Minimise	
$\sum_{i\in\mathcal{I}}\sum_j$	$\sum_{i \in \mathcal{J}} c_{ij} u_{ij}$
subject to	
$\sum_{i \in \mathcal{I}} x_i \le p,$	
$\sum_{i\in\mathcal{I}}u_{ij}\geq 1$	$\forall j \in \mathcal{J},$
$u_{ij} \le x_i$	$\forall i \in \mathcal{I}, j \in \mathcal{J},$
$x_i \in \mathbb{B}$	$\forall i \in \mathcal{I},$
$u_{ij} \in \mathbb{B}$	$\forall i \in \mathcal{I}, j \in \mathcal{J}.$

Model 5.2.5: *p*-Median model.

An important assumption is that the costs satisfy the triangle inequality:

$$c_{i_1j_1} \leq c_{i_1j_2} + c_{i_2j_2} + c_{i_2j_1} \qquad \forall i_1, i_2 \in \mathcal{I}, j_1, j_2 \in \mathcal{J}.$$

Using a similar Lagrangian relaxation approach as above, Jain and Vazirani (2001) derive a 6approximation Primal-Dual algorithm for the Metric *p*-Median problem. That is, for minimum θ^* and Primal-Dual objective value θ^{PD} we are guaranteed that $\theta^{PD} \leq 6\theta^*$.

The MCLP can be translated to the *p*-Median problem, but the costs will not satisfy the triangle inequality. As the *p*-Median problem is a minimisation problem, we have to define the costs as follows:

$$c_{ij} = \begin{cases} 0 & \text{if } a_{ij} = 1 \\ d_j & \text{otherwise} \end{cases}.$$

Thus, the objective corresponds to the minimisation of uncovered demand. Unfortunately, this definition does not satisfy the triangle inequality, see Figure 5.2.1 for an example. Therefore, we cannot use the performance guarantee of the Primal-Dual algorithm for the Metric p-Median problem.

To conclude, Primal-Dual approximation methods use the complementary slackness conditions of the LP relaxation to construct feasible (integer) solutions of the original optimisation problem. The MCLP is similar to other optimisation problems that have Primal-Dual approximation algorithms. We have only considered the work of Jain and Vazirani (2001), so a literature survey would be a good starting point for future research. The research can focus on adapting existing algorithms and using them for the MCLP.



Figure 5.2.1: Example of an MCLP instance transformation to a p-Median problem instance. Bases are depicted as squares, demand points as circles. Points are connected to adjacent bases and their objective coefficients are shown in brackets. Notice that the objective coefficients of the p-Median problem instance do not satisfy the triangle inequality.

Appendix A

Complexity Theory

Contents

A.1	Intro	oduction	178
A.2	Com	plexity Classes P and NP	179
	A.2.1	NP-Hard Problems	179

A.1 Introduction

Complexity theory provides a thorough approach to classify the computational complexity of optimisation problems, i.e., which problems are 'easy' to solve and which are 'difficult'. This classification is based on worst-case instances of the optimisation problem and it is therefore possible that typical real-life instances of 'difficult' problems can be solved efficiently. However, the classification builds on certain (widely accepted) assumptions that the complexity classes are intrinsically different. Consequently, even though we can solve some instances of 'difficult' problems to optimality efficiently, it is not possible to do so in general (regardless of the solution method).

An example of a 'difficult' problem is the Maximal Covering Location problem (MCLP), the model central to Chapters 3 and 4. We have observed that solving instances based on Euclidean or real-life distances requires little computational time. However, this is not the case for MCLP instances in general, especially when the dimension of the instance is large.

We note that solving very large instances of 'easy' problems can require an impractical amount of time (although it is mathematically efficient). This remark should be seen in the context of 'difficult' problems, where in general very large instances cannot even be solved to optimality within any reasonable computational time. Therefore, one is more inclined to use heuristic solution methods for 'difficult' problems.

The most famous assumption in complexity theory is that $P \neq NP$, which is widely accepted among mathematicians. However, there is no proof (nor disproof) of this statement. To explain the meaning of this assumption, we have to introduce some terminology.

An optimisation problem consists of a mathematical description of a feasible region and an objective function. The problem is to find a feasible solution such that the objective function is maximised or minimised. Complexity theory (for our intended use) considers so-called decision problems instead of optimisation problems. The required input for a decision problem is a feasible region, an objective function, and a scalar threshold for the objective. It asks whether there exists a feasible solution such that its objective value is greater than or equal to the scalar threshold (for maximisation problems). Therefore, the only possible answer is YES or NO. An instance where the answer is YES, is called a YES-instance (and likewise for NO).

An instance of a decision problem has a certain input size, which is the length of a 'reasonable' encoding to encode the specification of the instance data. For example, consider encodings for integers. Decimal and binary encodings are reasonable, having length $\mathcal{O}(\log(n))$ for $n \in \mathbb{N}$. However, a unary encoding is not reasonable, with a length of $\mathcal{O}(n)$.

We measure the computational time of algorithms in terms of the required number of elementary calculations and express it as a function of the instance input size. The computational time is polynomial if it can be asymptotically bounded from above by a polynomial function of the input size. In this case, the algorithm is efficient and usually applicable in practice. Inefficient algorithms typically have an exponential computational time. Additional classifications exist, such as strongly and pseudo-polynomial, see for instance Papadimitriou and Steiglitz (1998).

A.2 Complexity Classes *P* and *NP*

The complexity class NP consists of all decision problems that are efficiently YES-verifiable, that is, in polynomial time with respect to the instance input size. For the YES-verification we are given a YES-certificate (a proof) that the instance is a YES-instance. If a problem is in NP, this certificate can be verified in polynomial time. Note that the definition of NP does not involve NO-instances. The name 'NP' stands for non-deterministic polynomial time.

The decision problem variant of the Maximal Covering Location problem is in NP, as a YEScertificate is the subset of bases to open to achieve the desired coverage. Verifying this certificate is straightforward: make a list of all covered demand points for each opened base, and compare the sum of the covered demand with the given scalar threshold. This requires at most $\mathcal{O}(|\mathcal{I}||\mathcal{J}|)$ time, which is polynomial in the input size.

The complexity class P consists of a subset of decision problems in NP. Decision problems in P are verifiable and answerable in polynomial time: not only can we verify a YES-certificate efficiently, we can determine whether an instance is a YES- or NO-instance in polynomial time. Hence, we can solve problems in P efficiently. This is represented in the name: 'P' stands for polynomial time.

The widely accepted assumption $P \neq NP$ means that the complexity class P is a strict subset of NP, i.e., there are decision problems in NP that are not in P. This implies that there are at least two intrinsically different complexity classes: some problems cannot be solved efficiently. Most statements on the complexity of a problem therefore include 'unless P = NP'.

A.2.1 NP-Hard Problems

There is a subset of problems that is the most difficult of all problems in NP. These decision problems are called NP-complete. A more formal definition is that a problem is NP-complete if it is in NP and for each problem in NP there exists a polynomial time reduction to it. A polynomial time reduction is a translation from one problem to another that requires at most polynomial time, where the translated instance is a YES-instance if and only if the original instance is a YES-instance. This implies that if we can solve an NP-complete problem efficiently, we can solve all NP problems efficiently by using these reductions. The Satisfiability problem is the most famous (and first) NP-complete problem, see Cook (1971).

Finally, NP-hard problems are not necessarily decision problems, but are as difficult as all NPcomplete problems. These are defined in a similar way as NP-complete problems. If an NPcomplete decision problem has an associated optimisation problem, that optimisation problem
is NP-hard.

We have already mentioned that the decision problem variant of the MCLP is in NP. In Theorem A.2.1 we prove that it is in fact NP-complete, making the MCLP NP-hard.

Theorem A.2.1 (Folklore). The decision problem variant of the MCLP is NP-complete. Hence, the MCLP is NP-hard. *Proof.* We construct a straightforward polynomial time reduction from the Set Covering decision problem to the decision problem variant of the MCLP. Karp (1972) proved that the Set Covering decision problem is NP-complete. The Set Covering decision problem is as follows. We are given a universe \mathcal{U} , a set \mathcal{S} of subsets of the universe, and an integer $k \in \mathbb{N}$. Does there exist a selection of at most k elements in \mathcal{S} such that its union is \mathcal{U} ?

Now consider the following reduction to the decision problem variant of the MCLP. First, label the elements of \mathcal{U} as $\{1, \ldots, |\mathcal{U}|\}$ and \mathcal{S} as $\{1, \ldots, |\mathcal{S}|\}$. For the MCLP, set the bases equal to $\mathcal{I} = \{1, \ldots, |\mathcal{S}|\}$ and the demand points to $\mathcal{J} = \{1, \ldots, |\mathcal{U}|\}$. Each point has unit demand, $d_j = 1$ for all $j \in \mathcal{J}$, and at most p = k bases can be opened. The adjacency parameters are defined such that $a_{ij} = 1$ if and only if universe element $j \in \mathcal{U}$ is in subset $i \in \mathcal{S}$. The objective threshold is equal to $|\mathcal{U}|$, i.e., all points must be covered. This transformation is clearly polynomial in the input size.

If the Set Covering instance is a YES-instance, we can open the corresponding bases and have all points covered (a YES-instance for the MCLP). Likewise, if the MCLP is a YES-instance, we select the corresponding subsets in S. The union is equal to U, so it is also a YES-instance. We conclude that we have constructed a polynomial time reduction from an *NP*-complete problem to the MCLP. Thus, the decision problem variant of the MCLP is *NP*-complete.
Appendix B

Implementation

Contents

B.1 Intr	oduction
B.2 A S	election of Algorithms 183
B.2.1	Coverage Calculation
B.2.2	Greedy Search
B.2.3	Swap Local Search
B.2.4	Determining the Travel Time Thresholds

B.1 Introduction

We have implemented several algorithms to obtain our numerical results. For all calculations we have used C++ (compiled with Microsoft Visual Studio 2012). The results are visualised in Matlab R2011b. The computational time required to solve the considered Maximal Covering Location problem (MCLP) instances ranges from a few seconds to a couple of minutes, depending on the used solution method. In particular, solving to optimality can be done within a few seconds. We have used a 64-bit Windows 7 operating system with a 2.5 GHz dual core processor and 4 GB of RAM.

In C++ we have built several object-oriented classes, which can be divided into four types: instance class, solution class, Gurobi class, and model class. The instance class type contains all MCLP data, such as the demand point weights, the maximum number of opened bases, and the adjacency parameters. A variation of this class contains the travel times between bases and points. The solution class stores all information of a feasible solution: the set of opened bases, the set of covered demand points, the coverage, the used solution method, the number of used iterations, etcetera. A variation is the solution class for the Linear Programming relaxation.

For a general Mixed Integer Programming solver we use Gurobi (see Gurobi Optimization (2014)) in order to solve all models and relaxations to optimality. The Gurobi class is property of Gurobi and is its C++ interface. The model class combines all three previous classes, connecting the instance data with the optimisation model. Its methods include the construction of the optimisation models and the solution algorithms (e.g., Greedy and Swap Local Search).

When appropriate, the methods and models have two variants: one for the non-robust MCLP and one for the Robust Counterpart MCLP. Furthermore, care has been taken to deal with precision errors (e.g., translating the Gurobi output to a solution class object).

The following section discusses a selection of the implemented algorithms. No actual code is provided. Instead, the algorithms are shown on an abstract level for accessibility.

B.2 A Selection of Algorithms

We go into the details of the following algorithms: the calculation of the coverage of a set of bases, the Greedy Search, the Swap Local Search, and the determination of the travel time threshold for the generation of instances. The calculation of the coverage is given for completeness. The methods for the sensitivity analysis of Chapter 3 are omitted, as they are straightforward iterations over the parameters. The Reverse, Revised, and Anti variations of the Greedy Search are almost exactly the same as the shown Greedy Search. The differences include closing bases (instead of opening) and minimising coverage (instead of maximising).

B.2.1**Coverage Calculation**

Algorithm B.2.1 shows a straightforward calculation of the coverage of a set of opened bases. It is used to determine the coverage of fixed solutions in the sensitivity analysis and of the (intermediate) solutions of the approximations.

Algorithm B.2.1 Calculate Coverage for the MCLP

Input: MCLP instance data (set of bases \mathcal{I} , set of points \mathcal{J} , number of bases to open $p \in \mathbb{N}$, adjacency parameters $a_{ij} \in \mathbb{B}$ for $i \in \mathcal{I}, j \in \mathcal{J}$, demand point weights $d_j \in \mathbb{R}_{>0}$ for $j \in \mathcal{J}$), set of opened bases $\mathcal{I}^* \subseteq \mathcal{I}$ with $|\mathcal{I}^*| \leq p$

Output: coverage of input set of opened bases

```
1: procedure CALCULATE COVERAGE(input)
```

- 2: (check input)
- initialise covered points: $\mathcal{J}^* = \{j \in \mathcal{J} : \sum_{i \in \mathcal{I}^*} a_{ij} \ge 1\}$ initialise coverage: $\theta^* = \sum_{j \in \mathcal{J}^*} d_j$ 3:
- 4:
- return θ^* 5:
- 6: end procedure

B.2.2 Greedy Search

The ρ -Greedy Search implementation is split into two algorithms. Algorithm B.2.2 is the iterative procedure of opening at most ρ additional bases until p bases are opened or until the maximum number of iterations have been reached. Usually, the maximum number of iterations is set to an unrestrictive value (e.g., equal to p). Other heuristics can easily be constructed by restricting the maximum number of iterations. For instance, a ρ -Swap Local Search can be performed after every two Greedy Search iterations. To allow additional uses of the Greedy Search, it has a set of candidate bases as input. These candidates are the only bases that can be added to the initial solution, allowing the exclusion of other bases.

Finally, notice that the initial set of opened bases and initial set of candidates are sorted (in an arbitrary but fixed way). The order of the bases affects the outcome of the algorithm, as we will show when discussing the Greedy Search iteration (Algorithm B.2.3). The sorting removes dependence on the input order, but is not required. In fact, the unsorted order can be used to indicate a preference of each base, where the more preferred bases are listed first.

The Greedy Search iteration is shown in Algorithm B.2.3. It is equivalent to a Brute Force search of all possible subsets of additional bases with cardinality ρ . It is a recursive Depth-First Search method, where each time a single additional base is opened and the search parameter is decreased by one. When no additional bases have to be opened, the input set of opened bases is returned. In the end, the subset of additional bases with the largest increase in coverage is returned.

As the demand weights are non-negative, coverage cannot decrease by opening additional bases. However, it is possible that opening additional bases does not increase the coverage. The method assumes that exactly ρ additional bases must be opened, even if this does not improve coverage. Therefore, we always accept the first intermediate solution (see the condition t == 1).

Although the search opens bases in a certain order, only the set of opened bases is relevant. To prevent symmetries, we open bases in the order of the candidates. This is achieved by iterating only up to and including the candidate on position $|\mathcal{C}_0^+| - (\rho_0 - 1)$. The remaining candidates are available in subsequent recursive calls of the method. Furthermore, all candidates preceding the selected candidate are removed, $\mathcal{C}_{new}^+ = \mathcal{C}_0^+ \setminus \{i_1^+, \ldots, i_t^+\}$, see also Example B.2.1.

Example B.2.1. Suppose we have $\mathcal{I} = \{1, \ldots, 6\}$, start without any open bases $(\mathcal{I}_0^G = \emptyset)$, and only base 6 is not a candidate $(\mathcal{C}_0^+ = \{1, \ldots, 5\})$. Each base has disjoint coverage and the coverage is equal to *i* for base $i \in \mathcal{I}$. At most p = 4 bases can be opened. We apply a single 3-Greedy Search iteration ($\rho_0 = 3$ and T = 1), see Algorithm B.2.2.

Table B.2.1 shows the sets \mathcal{I}_{new}^G and $\mathcal{I}_{current}^G$ of Algorithm B.2.3 for the different recursive call levels (where level 0 is the original). As can be seen, symmetric solutions are prevented by the search. Finally, the best set of bases to be opened is $\mathcal{I}_{best}^G = \{3, 4, 5\}$, which is returned. As we have set the maximum number of iterations T equal to one, the Greedy Search is terminated, and returns $\mathcal{I}_T^G = \{3, 4, 5\}$. One more base can still be opened, since p = 4.

Rec	Set \mathcal{I}_{new}^G onSet $\mathcal{I}_{current}^G$ onRecursive Call LevelRecursive Call Level					
0	1	2	2	1	0	$\mathcal{I}_{ ext{best}}^G$
	$\{1, 2\}$	$ \begin{array}{c} \{1,2,3\} \\ \{1,2,4\} \\ \{1,2,5\} \end{array} $		$\{1, 2, 5\}$		
{1}	$\{1, 3\}$	$\{ \begin{array}{c} \{1,3,4\} \\ \{1,3,5\} \end{array}$	$ \begin{array}{r} $	$\{1, 3, 5\}$	$\{1, 4, 5\}$	
	$\{1, 4\}$	$\{1, 4, 5\}$	$\{1, 4, 5\}$	$\{1, 4, 5\}$		$\{3, 4, 5\}$
{2}	$\{2, 3\}$	$\begin{array}{c} \{2,3,4\} \\ \{2,3,5\} \end{array}$	$\{2, 3, 4\}\ \{2, 3, 5\}$	$\{2, 3, 5\}$	$\{2, 4, 5\}$	
(-)	$\{2, 4\}$	$\{2, 4, 5\}$	$\{2, 4, 5\}$	$\{2, 4, 5\}$	(-, -, °)	
$\{3\}$	{3,4}	$\{3, 4, 5\}$	${3,4,5}$	$\{3, 4, 5\}$	$\{3, 4, 5\}$	

Table B.2.1: Example of a Greedy Search iteration.

Algorithm B.2.2 Greedy Search for the MCLP

```
Input: MCLP instance data (set of bases \mathcal{I}, set of points \mathcal{J}, number of bases to open p \in \mathbb{N},
     adjacency parameters a_{ij} \in \mathbb{B} for i \in \mathcal{I}, j \in \mathcal{J}, demand point weights d_j \in \mathbb{R}_{\geq 0} for j \in \mathcal{J}),
     initial set of opened bases \mathcal{I}_0^G \subseteq \mathcal{I} with |\mathcal{I}_0^G| \leq p,
     initial set of candidates that can be opened \mathcal{C}_0^+ \subseteq \mathcal{I} \setminus \mathcal{I}_0^G,
     search parameter \rho_0 \in \mathbb{N} with \rho_0 \leq |\mathcal{C}_0^+|, maximum number of iterations T \in \mathbb{N}_{>0}
Output: extended set of opened bases
 1: procedure \rho-GREEDY SEARCH(input)
           (check input)
 2:
           sort \mathcal{I}_0^G and \mathcal{C}_0^+ to prevent dependence on input order
 3:
           for iteration t = 1, \ldots, T do
 4:
                adjust search parameter: \rho_t = \min\{\rho_0, |\mathcal{C}_{t-1}^+|, p - |\mathcal{I}_{t-1}^G|\}
 5:
 6:
                 if \rho_t == 0 then
                      return \mathcal{I}_{t-1}^G
 7:
                else
 8:
                      get next opened bases: \mathcal{I}_t^G = \rho-GREEDY SEARCH ITERATION(\mathcal{I}_{t-1}^G, \mathcal{C}_{t-1}^+, \rho_t) set next candidates: \mathcal{C}_t^+ = \mathcal{C}_{t-1}^+ \setminus \mathcal{I}_t^G
 9:
10:
                 end if
11:
           end for
12:
           return \mathcal{I}_T^G
13:
14: end procedure
```

Algorithm B.2.3 Greedy Search Iteration for the MCLP

Input: MCLP instance data (set of bases \mathcal{I} , set of points \mathcal{J} , number of bases to open $p \in \mathbb{N}$, adjacency parameters $a_{ij} \in \mathbb{B}$ for $i \in \mathcal{I}, j \in \mathcal{J}$, demand point weights $d_j \in \mathbb{R}_{\geq 0}$ for $j \in \mathcal{J}$), initial set of opened bases $\mathcal{I}_0^G \subseteq \mathcal{I}$ with $ \mathcal{I}_0^G \leq p$, initial set of candidates that can be opened $\mathcal{C}_0^+ \subseteq \mathcal{I} \setminus \mathcal{I}_0^G$, search parameter $\rho_0 \in \mathbb{N}$ with $\rho_0 \leq \min\{p - \mathcal{I}_0^G , \mathcal{C}_0^+ \}$ Output: extended set of opened bases
1: procedure ρ-Greedy Search Iteration(input)
2: (check input)
3: sort \mathcal{I}_0^G and \mathcal{C}_0^+ to prevent dependence on input order
4: if $\rho_0 == 0$ then
5: return \mathcal{I}_0^G
6: $else$
7: initialise best opened bases: $\mathcal{I}_{\text{best}}^G = \mathcal{I}_0^G$
8: initialise best objective: $\theta_{\text{best}}^G = \text{CALCULATE COVERAGE}(\mathcal{I}_0^G)$
9: for iteration $t = 1,, (\mathcal{C}_0^+ - (\rho_0 - 1))$ do
10: select <i>t</i> -th candidate to open: $i_t^+ \in \mathcal{C}_0^+$
11: initialise new opened bases: $\mathcal{I}_{\text{new}}^G = \mathcal{I}_0^G \cup \{i_t^+\}$
12: initialise new candidates: $\mathcal{C}_{\text{new}}^+ = \mathcal{C}_0^+ \setminus \{i_1^+, \dots, i_t^+\}$
13: initialise new search parameter: $\rho_{\text{new}} = \rho_0 - 1$
14: get: $\mathcal{I}_{current}^G = \rho$ -GREEDY SEARCH ITERATION $(\mathcal{I}_{new}^G, \mathcal{C}_{new}^+, \rho_{new})$
15: get objective: $\theta_{\text{current}}^G = \text{CALCULATE COVERAGE}(\mathcal{I}_{\text{current}}^G)$
16:
17: set best opened bases: $\mathcal{I}_{\text{best}}^G = \mathcal{I}_{\text{current}}^G$
18: set best objective: $\theta_{\text{best}}^G = \theta_{\text{current}}^G$
19: end if
20: end for
21: return $\mathcal{I}^G_{ ext{best}}$
22: end if
23: end procedure

B.2.3 Swap Local Search

The Swap Local Search is similar to the Greedy Search and is also split into two algorithms. One of the main differences with the Greedy Search is that there are two sets of candidates: the bases that can be closed and those that can be opened. We assume that the union of both sets corresponds to the bases that can be modified. Hence, open bases that are closed, can be reopened and closed bases that are opened, can be closed again. It is therefore useful to combine both sets into one set of candidates, see Algorithm B.2.4. Modifications to this assumption are easy to implement.

In each iteration of Algorithm B.2.4 all allowed swaps are performed, starting with 1-Swap and continuing to ρ -Swap. Therefore, swaps with the least changes are preferred. This can have an impact on the convergence of the method to local maxima. The search is terminated if no improvements have been made at the end of a complete iteration.

Algorithm B.2.5 is equivalent to a Brute Force enumeration of all possible ρ -Swaps. A Depth-First Search is implemented by using recursive calls, similar to Algorithm B.2.3. However, we need to select two candidates at each step: one to close and one to open. Symmetric solutions are prevented using the same approach as the Greedy Search, see also Example B.2.2.

Example B.2.2. Suppose we have $\mathcal{I} = \{1, 2, 3, 4, 5, 6\}$ and p = 3 bases can be opened. The initial feasible solution is $\mathcal{I}_0^L = \{1, 2, 3\}$ and all bases are candidates $(\mathcal{C}_0^- = \{1, 2, 3\}$ and $\mathcal{C}_0^+ = \{4, 5, 6\}$). Each base has disjoint coverage and the coverage is equal to *i* for base $i \in \mathcal{I}$. We apply 2-Swap Local Search to the initial solution, but allow only one iteration (T = 1), see Algorithm B.2.4.

Table B.2.2 shows the sets \mathcal{I}_{new}^L and $\mathcal{I}_{current}^L$ of Algorithm B.2.5 for the different recursive call
levels (where level 0 is the original). Furthermore, the sets \mathcal{I}_{inner}^L and \mathcal{I}_{best}^L of Algorithm B.2.4
are given. To conclude, the 2-Swap Local Search returns the set $\{3, 5, 6\}$ as final solution.

	\mathbf{Set}	$\mathcal{I}_{\mathbf{new}}^L$ on	$\mathbf{Set} \mathcal{I}$	$\mathbf{c}_{\mathbf{urrent}}^{L}$ on	Sets in	
	Recursiv	ve Call Level	Recursive Call Level		Algorith	nm B.2.4
ρ -Swap	0	1	1	0	$\mathcal{I}^L_{ ext{inner}}$	$\mathcal{I}^L_{ ext{best}}$
1	$\begin{array}{l} \{2,3,4\}\\ \{2,3,5\}\\ \{2,3,6\}\\ \{1,3,4\}\\ \{1,3,5\}\\ \{1,3,6\}\\ \{1,2,4\}\\ \{1,2,5\}\\ \{1,2,6\}\end{array}$			$ \begin{array}{l} \{2,3,4\} \\ \{2,3,5\} \\ \{2,3,6\} \\ \{1,3,4\} \\ \{1,3,5\} \\ \{1,3,6\} \\ \{1,2,4\} \\ \{1,2,5\} \\ \{1,2,6\} \end{array} $	$\{2, 3, 6\}$	
	{2,3,4}	$\{ 3, 4, 5 \} \\ \{ 3, 4, 6 \} \\ \{ 2, 4, 5 \} \\ \{ 2, 4, 6 \} $		$\{3, 4, 6\}$	-	$\{3, 5, 6\}$
2	$\{2, 3, 5\}$	$\{3, 5, 6\}\ \{2, 5, 6\}$	$ \{ 3, 5, 6 \} \\ \{ 2, 5, 6 \} $	$\{3, 5, 6\}$	$\{3, 5, 6\}$	
	$\{1, 3, 4\}$	$\{1,4,5\} \\ \{1,4,6\}$	$\{1,4,5\} \\ \{1,4,6\}$	$\{1, 4, 6\}$		
	$\{1, 3, 5\}$	$\{1, 5, 6\}$	$\{1, 5, 6\}$	$\{1, 5, 6\}$		

Table B.2.2: Example of a Swap Local Search iteration.

Algorithm B.2.4 Swap Local Search for the MCLP

```
Input: MCLP instance data (set of bases \mathcal{I}, set of points \mathcal{J}, number of bases to open p \in \mathbb{N},
       adjacency parameters a_{ij} \in \mathbb{B} for i \in \mathcal{I}, j \in \mathcal{J}, demand point weights d_j \in \mathbb{R}_{\geq 0} for j \in \mathcal{J}),
       initial set of opened bases \mathcal{I}_0^L \subseteq \mathcal{I} with |\mathcal{I}_0^L| = p,
       initial set of candidates that can be closed and reopened \mathcal{C}_0^- \subseteq \mathcal{I}_0^L,
initial set of candidates that can be opened and reclosed \mathcal{C}_0^+ \subseteq \mathcal{I} \setminus \mathcal{I}_0^L,
        search parameter \rho_0 \in \mathbb{N} with \rho_0 \leq \min\{|\mathcal{C}_0^-|, |\mathcal{C}_0^+|\}, maximum number of iterations T \in \mathbb{N}_{>0}
Output: swapped set of opened bases
  1: procedure \rho-SWAP LOCAL SEARCH(input)
               (check input)
  2:
              set all candidates: \mathcal{C}_0^{\pm} = \mathcal{C}_0^- \cup \mathcal{C}_0^+
sort \mathcal{I}_0^L and \mathcal{C}_0^{\pm} to prevent dependence on input order
  3:
  4:
              initialise best opened bases: \mathcal{I}_{\text{best}}^L = \mathcal{I}_0^L
initialise best objective: \theta_{\text{best}}^L = \text{CALCULATE COVERAGE}(\mathcal{I}_0^L)
for iteration t_{\text{outer}} = 1, \dots, T do
 5:
 6:
  7:
                     initialise outer candidates to close: \mathcal{C}_{outer}^{L} = \mathcal{C}_{0}^{L} \setminus \mathcal{C}_{outer}^{L}
initialise outer candidates to close: \mathcal{C}_{outer}^{+} = \mathcal{C}_{0}^{\pm} \setminus \mathcal{C}_{outer}^{L}
 8:
 9:
10:
                      initialise improvement flag: c_{\text{improvement}} = \text{false}
11:
                      for iteration t_{\text{inner}} = 1, \ldots, \rho_0 do
12:
                             get: \mathcal{I}_{inner}^{L} = \rho-Swap LOCAL SEARCH ITERATION(\mathcal{I}_{outer}^{L}, \mathcal{C}_{outer}^{-}, \mathcal{C}_{outer}^{+}, t_{inner})
get inner objective: \theta_{inner}^{L} = CALCULATE COVERAGE(\mathcal{I}_{inner}^{L})
13:
14:
                             if \theta_{\text{inner}}^L > \theta_{\text{best}}^L then
15:
16:
                                    set improvement flag: c_{\text{improvement}} = \mathbf{true}
                                    set best opened bases: \mathcal{I}_{\text{best}}^{L} = \mathcal{I}_{\text{inner}}^{L}
set best objective: \theta_{\text{best}}^{L} = \theta_{\text{inner}}^{L}
17:
18:
                              end if
19:
                      end for
20:
21:
                      if c_{\text{improvement}} == false then
                             return \mathcal{I}_{\text{best}}^L
22:
                      end if
23:
               end for
24:
               return \mathcal{I}_{	ext{best}}^L
25:
26: end procedure
```

Algorithm B.2.5 Swap Local Search Iteration for the MCLP

```
Input: MCLP instance data (set of bases \mathcal{I}, set of points \mathcal{J}, number of bases to open p \in \mathbb{N},
        adjacency parameters a_{ij} \in \mathbb{B} for i \in \mathcal{I}, j \in \mathcal{J}, demand point weights d_j \in \mathbb{R}_{\geq 0} for j \in \mathcal{J}),
        initial set of opened bases \mathcal{I}_0^L \subseteq \mathcal{I} with |\mathcal{I}_0^L| = p,
        initial set of candidates that can be closed and reopened \mathcal{C}_0^- \subseteq \mathcal{I}_0^L, initial set of candidates that can be opened and reclosed \mathcal{C}_0^+ \subseteq \mathcal{I} \setminus \mathcal{I}_0^L,
        search parameter \rho_0 \in \mathbb{N} with \rho_0 \leq \min\{|\mathcal{C}_0|, |\mathcal{C}_0|\}
Output: swapped set of opened bases
  1: procedure \rho-SWAP LOCAL SEARCH ITERATION(input)
  2:
                (check input)
                sort \mathcal{I}_0^L, \hat{\mathcal{C}}_0^- and \mathcal{C}_0^+ to prevent dependence on input order
  3:
  4:
                if \rho_0 == 0 then
                       return \mathcal{I}_0^L
  5:
                else
  6:
                       initialise best opened bases: \mathcal{I}_{\text{best}}^L = \mathcal{I}_0^L
initialise best objective: \theta_{\text{best}}^L = \text{CALCULATE COVERAGE}(\mathcal{I}_0^L)
for iteration t^- = 1, \dots, (|\mathcal{C}_0^-| - (\rho_0 - 1)) do
  7:
  8:
  9.
                               for iteration t^+ = 1, ..., (|\mathcal{C}_0^+| - (\rho_0 - 1)) do
10:
                                       select t^--th candidate to close: i_{t^-}^- \in C_0^-
11:
                                       select t^+-th candidate to open: i_{t^+}^+ \in \mathcal{C}_0^+
12:
                                      initialise new opened bases: \mathcal{I}_{\text{new}}^L = (\mathcal{I}_0^L \setminus \{i_{t^-}^-\}) \cup \{i_{t^+}^+\}
initialise new candidates to close: \mathcal{C}_{\text{new}}^- = \mathcal{C}_0^- \setminus \{i_1^-, \dots, i_{t^-}^-\}
initialise new candidates to open: \mathcal{C}_{\text{new}}^+ = \mathcal{C}_0^+ \setminus \{i_1^+, \dots, i_{t^+}^+\}
13:
14:
15:
                                       initialise new search parameter: \rho_{\text{new}} = \rho_0 - 1
16:
                                      get: \mathcal{I}_{current}^{L} = \rho-Swap Local Search Iteration(\mathcal{I}_{new}^{L}, \mathcal{C}_{new}^{-}, \mathcal{C}_{new}^{+}, \rho_{new})
get objective: \theta_{current}^{L} = CALCULATE COVERAGE(\mathcal{I}_{current}^{L})
if \theta_{current}^{L} > \theta_{best}^{L} then
set best opened bases: \mathcal{I}_{best}^{L} = \mathcal{I}_{current}^{L}
set best objective: \theta_{best}^{L} = \theta_{current}^{L}
17:
18:
19:
20:
21:
                                       end if
22:
                               end for
23:
                       end for
24:
                       return \mathcal{I}_{\text{best}}^L
25:
                end if
26:
27: end procedure
```

B.2.4 Determining the Travel Time Thresholds

As described in Section 4.5.3, we have generated instances where the maximum coverage of each instance has been specified. In order to generate such instances, we have to determine a suitable travel time threshold. That is, given a desired maximum fraction of covered demand $\alpha_{\max} \in [0, 1]$, find the minimum travel time threshold such that the maximum coverage of the resulting instance is as close as possible to α_{\max} .

One approach would be to iteratively increase the travel time threshold until the desired coverage has been attained. Our implementation uses a bisection method to determine the threshold, see Algorithm B.2.6. The problem does not make sense for p = 0, so let $p \in \mathbb{N}_{\geq 1}$. Notice that we can limit the possible thresholds to the value zero and the unique travel time values between bases and points. Hence, the initial lower bound of the bisection approach is set to a travel time threshold of zero and the corresponding maximum coverage α_{LB} is determined. Note that α_{LB} is not equal to zero if the travel time τ_{ij} is zero for some $i \in \mathcal{I}$ and $j \in \mathcal{J}$.

As we do not allow negative thresholds, the maximum coverage fraction cannot be lower than α_{LB} . Therefore, if the desired maximum coverage is lower, the best possible threshold (zero) has been found. Otherwise, the initial upper bound of the bisection approach is set: full coverage is attained if the maximum travel time is used as threshold.

Finally, a standard bisection method determines the desired threshold. Note that we bisect on the indexes of the sorted thresholds, not on the actual values. Although not explicitly shown, the number of iterations is bounded by $\mathcal{O}(\log (|\mathcal{I}||\mathcal{J}|))$. The bisection method terminates if the lower and upper bound are adjacent, and the threshold resulting in a maximum coverage nearest to α_{\max} is chosen. By design, α_{LB} is always strictly less than α_{\max} . As a result, we obtain the minimum travel time threshold to attain the desired bound. See also Example B.2.3.

Example B.2.3. Consider the instance where $\mathcal{I} = \mathcal{J} = \{1, \ldots, 5\}$, at most p = 1 base can be opened, and the travel times are

$$\tau_{ij} = \begin{cases} 5(j-1) & \text{if } i = 1 \text{ and } j \in \mathcal{J} \\ 100 & \text{otherwise} \end{cases}$$

By construction, it is always optimal to open the first base. Suppose we set $\alpha_{\text{max}} = \frac{2}{3}$ and apply Algorithm B.2.6. The possible thresholds are $\mathcal{T} = \{0, 5, 10, 15, 20, 100\}$. For $t_{\text{LB}} = 1$ (a threshold of zero) we have $\alpha_{\text{LB}} = \frac{1}{5} < \alpha_{\text{max}}$ and continue with the bisection method.

The first selected index is $t_{\text{current}} = 1 + \left\lceil \frac{6-1}{2} \right\rceil = 4$ (a threshold of 15), resulting in a maximum covered fraction of $\frac{4}{5} \ge \alpha_{\text{max}}$. Therefore, the upper bound is lowered to $t_{\text{UB}} = 4$ and the process is repeated. The next iteration selects $t_{\text{current}} = 3$ (a threshold of 10) with coverage fraction $\frac{3}{5} < \alpha_{\text{max}}$. The lower bound is updated to $t_{\text{LB}} = 3$ and the bisection is terminated, since $t_{\text{UB}} - t_{\text{LB}} = 1$. We conclude that the lower bound leads to a coverage closest to the desired coverage and a travel time threshold of 10 is returned.

 $$\overline{\mbox{Algorithm B.2.6}}$$ Determine Travel Time Thresholds for the MCLP

Input: MCLP instance data (set of bases \mathcal{I} , set of points \mathcal{J} , number of bases to open $p \in \mathbb{N}_{\geq 1}$, travel times $\tau_{ij} \in \mathbb{R}_{\geq 0}$ for $i \in \mathcal{I}, j \in \mathcal{J}$, demand point weights $d_j \in \mathbb{R}_{\geq 0}$ for $j \in \mathcal{J}$),
desired maximum fraction of covered demand $\alpha_{\max} \in [0, 1]$
Output: minimum travel time threshold to attain the maximum fraction of covered demand
nearest to α_{\max}
1: procedure DETERMINE TRAVEL TIME THRESHOLDS(input)
2. (check input) 2. sot all unique travel time thresholds: $\mathcal{T} = \{0\} \cup \{\sigma_i, i \in \mathcal{T} \mid i \in \mathcal{T}\}$
5: Set all unique travel time timesholds. $T = \{0\} \cup \{T_{ij} : i \in \mathcal{I}, j \in \mathcal{J}\}$
4: Soft / III ascending order
5. initialize lower bound index: $t_{-} = 1$
5. Initialise lower bound index. $t_{\rm LB} = 1$ 6. soloct $t_{\rm TD}$ th travel time as threshold: $\sigma_{\rm travel} = \sigma_{\rm travel} \in \mathcal{T}$
$ \text{Select } t_{\text{LB}} \text{-th travel time as timeshold. } t_{\text{theshold}} = t_{t_{\text{LB}}} \in \mathcal{T} $
7: set adjacency parameters: $a_{ii} = \begin{cases} 1 & \tau_{ij} \leq \tau_{\text{threshold}} \\ \forall i \in \mathcal{I}, i \in \mathcal{J} \end{cases}$
0 otherwise
8: solve corresponding MCLP giving maximum coverage fraction $\alpha_{\text{LB}} \in [0, 1]$
9: if $\alpha_{\rm LB} \ge \alpha_{\rm max}$ then
10: return $ au_{\text{threshold}}$
11: end if
12: initialise upper bound index and coverage fraction: $t_{\rm UB} = \mathcal{T} , \alpha_{\rm UB} = 1$
13: loop
14: set current index: $t_{\text{current}} = t_{\text{LB}} + \left[\frac{1}{2}(t_{\text{UB}} - t_{\text{LB}})\right]$
15: select t_{current} -th travel time as threshold: $\tau_{\text{theshold}} = \tau_{t_{\text{current}}} \in \mathcal{T}$
16: set adjacency parameters: $a_{ij} = \begin{cases} 1 & \gamma_{ij} \geq \gamma_{\text{threshold}} \\ 0 & \text{otherwise} \end{cases} \forall i \in \mathcal{I}, j \in \mathcal{J} \end{cases}$
17: solve corresponding MCLP giving maximum coverage fraction $\alpha_{\text{current}} \in [0, 1]$
18: if $\alpha_{\text{current}} < \alpha_{\text{max}}$ then
19: set lower bound: $t_{\rm LB} = t_{\rm current}$, $\alpha_{\rm LB} = \alpha_{\rm current}$
20: else
21: set upper bound: $t_{\rm UB} = t_{\rm current}$, $\alpha_{\rm UB} = \alpha_{\rm current}$
22: end if
23: if $(t_{\rm UB} - t_{\rm LB}) == 1$ then
24: if $(\alpha_{\max} - \alpha_{LB}) < (\alpha_{UB} - \alpha_{\max})$ then
25: return $ au_{t_{LB}}$
26: else
27: return $ au_{t_{\mathrm{UB}}}$
28: end if
29: end if
30: end loop
31: end procedure

Appendix C

Additional Tables

Contents

C.1	Travel Time Thresholds	194
C.2	Greedy Search Results	196
C.3	Swap Local Search Results	201

C.1 Travel Time Thresholds

The generation of the Euclidean MCLP instances requires the specification of the travel time threshold. The used thresholds depend on the desired maximum attainable coverage, as explained in Section 4.5.3. The resulting average thresholds are shown in Table C.1.1.

		A	verage (Standard	Deviation) of Tra	vel Time Thresho	lds
	Opened		Maximum	Coverage		
Size	Bases	70%	80%	90%	100%	Combined
125	5	0.1886(0.0079)	$0.2141 \ (0.0079)$	0.2444(0.0083)	0.3339(0.0081)	$0.2452 \ (0.0556)$
125	10	0.1204(0.0055)	0.1397(0.0062)	0.1609(0.0073)	0.2152(0.0061)	0.1591(0.0361)
125	15	0.0910(0.0056)	0.1074(0.0053)	0.1259(0.0047)	0.1670(0.0059)	0.1228 (0.0289)
125	Combined	$0.1333\ (0.0415)$	$0.1538\ (0.0453)$	$0.1771 \ (0.0503)$	$0.2387 \ (0.0707)$	$0.1757 \ (0.0661)$
250	5	$0.1954 \ (0.0052)$	0.2196(0.0051)	0.2482(0.0061)	0.3337 (0.0096)	0.2492 (0.0528)
250	10	0.1293(0.0030)	0.1465(0.0035)	0.1659(0.0037)	0.2159(0.0044)	0.1644 (0.0327)
250	15	0.1005(0.0025)	0.1144(0.0026)	0.1313(0.0027)	0.1706(0.0031)	0.1292(0.0265)
250	Combined	$0.1417 \ (0.0400)$	$0.1602 \ (0.0443)$	0.1818 (0.0494)	$0.2401 \ (0.0693)$	0.1809 (0.0637)

Table C.1.1: Average travel time thresholds for the generated Euclidean instance classes.

C.2 Greedy Search Results

The following tables contain the details of the Greedy Search performance (the relative optimality gaps) for the generated Euclidean instances. We refer to Section 4.5 for the description of the instances and the Greedy Search variants.

		Average (Sta	reedy Search			
	Opened		Maximum	Coverage		
Size	Bases	70%	80%	90%	100%	Combined
125	5	$0.0153 \ (0.0178)$	$0.0286 \ (0.0215)$	0.0310(0.0227)	$0.0134\ (0.0108)$	$0.0221 \ (0.0202)$
125	10	0.0109(0.0114)	0.0208(0.0149)	$0.0312 \ (0.0197)$	$0.0145 \ (0.0125)$	$0.0194 \ (0.0168)$
125	15	$0.0110 \ (0.0109)$	$0.0172 \ (0.0112)$	$0.0223 \ (0.0120)$	$0.0106\ (0.0083)$	$0.0153 \ (0.0117)$
125	Combined	$0.0124\ (0.0138)$	$0.0222 \ (0.0170)$	$0.0282\ (0.0190)$	$0.0128 \ (0.0107)$	$0.0189 \ (0.0168)$
250	5	0.0218(0.0147)	$0.0295 \ (0.0205)$	0.0332(0.0246)	0.0137 (0.0129)	$0.0245 \ (0.0201)$
250	10	0.0169(0.0118)	$0.0281 \ (0.0142)$	$0.0352 \ (0.0195)$	0.0230(0.0102)	$0.0258 \ (0.0158)$
250	15	$0.0139\ (0.0101)$	$0.0256\ (0.0158)$	$0.0351 \ (0.0123)$	$0.0135\ (0.0062)$	$0.0220 \ (0.0146)$
250	Combined	$0.0175 \ (0.0127)$	$0.0277 \ (0.0170)$	$0.0345\ (0.0193)$	$0.0167 \ (0.0110)$	$0.0241 \ (0.0170)$

Table C.2.1: Average relative optimality gaps of 1-Greedy Search for the generated Euclidean instances.

		Average (Sta	reedy Search			
	Opened		Maximum	Coverage		
Size	Bases	70%	80%	90%	100%	Combined
125	5	$0.0125 \ (0.0156)$	0.0269(0.0213)	0.0279(0.0217)	0.0102(0.0103)	$0.0194 \ (0.0195)$
125	10	0.0103(0.0112)	$0.0203 \ (0.0153)$	0.0307 (0.0200)	0.0148(0.0124)	$0.0190 \ (0.0168)$
125	15	$0.0111 \ (0.0109)$	$0.0168\ (0.0110)$	0.0222 (0.0113)	$0.0103 \ (0.0086)$	$0.0151 \ (0.0115)$
125	Combined	$0.0113 \ (0.0127)$	$0.0213 \ (0.0168)$	$0.0269 \ (0.0185)$	$0.0118 \ (0.0107)$	$0.0178\ (0.0163)$
250	5	0.0179(0.0138)	$0.0296 \ (0.0178)$	0.0300(0.0191)	0.0073 (0.0102)	$0.0212 \ (0.0181)$
250	10	0.0170(0.0118)	0.0254(0.0141)	$0.0327 \ (0.0179)$	$0.0226 \ (0.0102)$	$0.0244 \ (0.0148)$
250	15	$0.0138\ (0.0097)$	$0.0248\ (0.0161)$	$0.0348\ (0.0123)$	$0.0133\ (0.0058)$	$0.0217 \ (0.0146)$
250	Combined	$0.0162 \ (0.0119)$	$0.0266 \ (0.0161)$	$0.0325 \ (0.0167)$	$0.0144 \ (0.0109)$	$0.0224 \ (0.0160)$

Table C.2.2: Average relative optimality gaps of 2-Greedy Search for the generated Euclidean instances.

		Average (Standa	e 1-Greedy Search			
	Opened		Maximum	Coverage		
Size	Bases	70%	80%	90%	100%	Combined
125	5	0.1395(0.0340)	$0.1477 \ (0.0353)$	0.1320(0.0360)	$0.0321 \ (0.0187)$	$0.1128 \ (0.0567)$
125	10	0.1210(0.0386)	$0.1181 \ (0.0353)$	0.1133(0.0334)	0.0378(0.0204)	$0.0976 \ (0.0475)$
125	15	0.0888(0.0290)	$0.0983 \ (0.0313)$	0.0948(0.0284)	0.0342 (0.0193)	0.0790(0.0377)
125	Combined	$0.1164 \ (0.0399)$	$0.1214 \ (0.0395)$	$0.1134 \ (0.0359)$	$0.0347 \ (0.0195)$	$0.0965 \ (0.0498)$
250	5	0.1475(0.0410)	0.1426(0.0381)	0.1478(0.0374)	$0.0356 \ (0.0259)$	$0.1184 \ (0.0598)$
250	10	$0.1382 \ (0.0368)$	$0.1398\ (0.0307)$	$0.1313 \ (0.0315)$	$0.0497 \ (0.0197)$	$0.1148 \ (0.0483)$
250	15	$0.1257 \ (0.0231)$	$0.1272 \ (0.0271)$	$0.1195\ (0.0225)$	$0.0466\ (0.0175)$	$0.1047 \ (0.0407)$
250	Combined	$0.1371 \ (0.0354)$	$0.1365 \ (0.0328)$	$0.1329 \ (0.0330)$	$0.0440 \ (0.0220)$	$0.1126 \ (0.0505)$

Table C.2.3: Average relative optimality gaps of Reverse 1-Greedy Search for the generated Euclidean instances.

198

		Average (Standa	e 2-Greedy Search			
	Opened		Maximum	a Coverage		
Size	Bases	70%	80%	90%	100%	Combined
125	5	$0.1384\ (0.0360)$	0.1478(0.0353)	$0.1316\ (0.0359)$	0.0298(0.0194)	$0.1119 \ (0.0577)$
125	10	0.1210(0.0386)	0.1179(0.0352)	0.1132(0.0334)	0.0378(0.0203)	0.0974 (0.0474)
125	15	$0.0888 \ (0.0290)$	$0.0983\ (0.0313)$	$0.0948\ (0.0284)$	$0.0342 \ (0.0193)$	$0.0790 \ (0.0377)$
125	Combined	$0.1161 \ (0.0402)$	$0.1213 \ (0.0394)$	$0.1132 \ (0.0358)$	$0.0339 \ (0.0198)$	$0.0961 \ (0.0501)$
250	5	0.1479(0.0410)	0.1411(0.0402)	$0.1443 \ (0.0353)$	$0.0355 \ (0.0251)$	$0.1172 \ (0.0593)$
250	10	$0.1380\ (0.0372)$	$0.1398\ (0.0306)$	$0.1306\ (0.0316)$	$0.0493 \ (0.0192)$	$0.1144 \ (0.0484)$
250	15	$0.1251 \ (0.0229)$	$0.1273 \ (0.0272)$	$0.1202 \ (0.0232)$	$0.0462 \ (0.0175)$	$0.1047 \ (0.0409)$
250	Combined	$0.1370 \ (0.0356)$	$0.1361 \ (0.0335)$	$0.1317 \ (0.0318)$	$0.0437 \ (0.0215)$	$0.1121 \ (0.0503)$

Table C.2.4: Average relative optimality gaps of Reverse 2-Greedy Search for the generated Euclidean instances.

		Average (Stand	Average (Standard Deviation) of Relative Optimality Gaps of Anti					
	Opened		Maximum	Coverage				
Size	Bases	70%	80%	90%	100%	Combined		
125	5	$0.8745\ (0.0363)$	$0.8687 \ (0.0352)$	0.8660(0.0341)	$0.8085 \ (0.0416)$	$0.8544 \ (0.0454)$		
125	10	$0.8710 \ (0.0295)$	$0.8666 \ (0.0280)$	$0.8614 \ (0.0305)$	$0.8211 \ (0.0332)$	$0.8550 \ (0.0361)$		
125	15	$0.8716\ (0.0344)$	$0.8558\ (0.0298)$	$0.8406\ (0.0279)$	$0.8084\ (0.0381)$	$0.8441 \ (0.0401)$		
125	Combined	$0.8723\ (0.0333)$	$0.8637 \ (0.0315)$	$0.8560\ (0.0327)$	$0.8127 \ (0.0380)$	$0.8512 \ (0.0409)$		
250	5	$0.8960 \ (0.0293)$	0.8909(0.0244)	$0.8909 \ (0.0196)$	0.8502(0.0259)	0.8820 (0.0310)		
250	10	0.8942(0.0227)	$0.8945 \ (0.0214)$	0.8879(0.0222)	0.8682(0.0268)	$0.8862 \ (0.0256)$		
250	15	$0.8919\ (0.0200)$	$0.8877 \ (0.0199)$	$0.8820 \ (0.0237)$	$0.8646\ (0.0201)$	$0.8816 \ (0.0233)$		
250	Combined	$0.8940 \ (0.0242)$	$0.8910 \ (0.0220)$	$0.8869\ (0.0221)$	$0.8610 \ (0.0255)$	$0.8832 \ (0.0268)$		

Table C.2.5: Average relative optimality gaps of Anti 1-Greedy Search for the generated Euclidean instances.

Average (Standard Deviation) of Relative	Optimality	Gaps of Ant	i 2-Greedy Search
-----------------------------	---------------	------------	-------------	-------------------

			,			•		
	Opened		Maximum Coverage					
Size	Bases	70%	80%	90%	100%	Combined		
125	5	0.8840(0.0302)	0.8829(0.0306)	0.8787(0.0273)	0.8236(0.0309)	$0.8673 \ (0.0390)$		
125	10	0.8911(0.0244)	0.8763(0.0234)	0.8729(0.0254)	0.8402(0.0268)	$0.8701 \ (0.0311)$		
125	15	0.8863(0.0281)	0.8731(0.0243)	0.8557(0.0246)	0.8267(0.0285)	0.8605(0.0345)		
125	Combined	$0.8872 \ (0.0276)$	$0.8774\ (0.0264)$	$0.8691 \ (0.0274)$	$0.8302\ (0.0295)$	$0.8660 \ (0.0352)$		
250	5	0.9015(0.0231)	0.8994 (0.0194)	0.8933(0.0234)	0.8567(0.0208)	$0.8877 \ (0.0282)$		
250	10	0.9048(0.0194)	0.9019(0.0177)	0.8995 (0.0188)	0.8804(0.0246)	$0.8967 \ (0.0223)$		
250	15	0.8990 (0.0180)	$0.8941 \ (0.0164)$	$0.8887 \ (0.0163)$	$0.8739\ (0.0169)$	$0.8889 \ (0.0193)$		
250	Combined	$0.9017 \ (0.0203)$	$0.8985\ (0.0181)$	$0.8938\ (0.0201)$	$0.8703\ (0.0232)$	$0.8911 \ (0.0239)$		

Table C.2.6: Average relative optimality gaps of Anti 2-Greedy Search for the generated Euclidean instances.

		Average (Stand	Average (Standard Deviation) of Relative Optimality Gaps of Revised Anti 1-Greedy Search					
	Opened		Maximur	n Coverage				
Size	Bases	70%	80%	90%	100%	Combined		
125	5	0.7722(0.0505)	0.7343(0.0449)	$0.6931 \ (0.0500)$	0.4916(0.0403)	$0.6728\ (0.1180)$		
125	10	0.8007(0.0474)	0.7583(0.0462)	0.7038(0.0528)	0.5096(0.0437)	$0.6931 \ (0.1212)$		
125	15	0.8169(0.0526)	$0.7606\ (0.0505)$	0.6972(0.0450)	0.5120(0.0389)	$0.6967 \ (0.1241)$		
125	Combined	$0.7966 \ (0.0532)$	$0.7511 \ (0.0484)$	$0.6980 \ (0.0492)$	$0.5044 \ (0.0417)$	$0.6875 \ (0.1214)$		
250	5	0.7528(0.0439)	0.7203(0.0376)	$0.6885 \ (0.0316)$	0.4994(0.0333)	$0.6652 \ (0.1053)$		
250	10	0.7718(0.0314)	0.7359(0.0318)	0.6934(0.0310)	0.5323(0.0343)	$0.6834\ (0.0971)$		
250	15	$0.7877 \ (0.0289)$	$0.7504\ (0.0298)$	$0.6974\ (0.0290)$	$0.5263\ (0.0293)$	$0.6904 \ (0.1044)$		
250	Combined	$0.7708\ (0.0379)$	$0.7355\ (0.0352)$	$0.6931 \ (0.0306)$	$0.5193 \ (0.0352)$	$0.6797 \ (0.1027)$		

Table C.2.7: Average relative optimality gaps of Revised Anti 1-Greedy Search for the generated Euclidean instances.

\sim	
0	
0	

		Average (Stand	Anti 2-Greedy Search			
	Opened		Maximur	n Coverage		
Size	Bases	70%	80%	90%	100%	Combined
125	5	0.7753(0.0481)	0.7405(0.0429)	$0.7021 \ (0.0461)$	0.4978(0.0386)	$0.6789 \ (0.1165)$
125	10	0.8038(0.0461)	0.7629(0.0439)	0.7108(0.0522)	0.5249(0.0410)	0.7006(0.1163)
125	15	0.8203(0.0513)	0.7677 (0.0465)	0.7030(0.0421)	$0.5277 \ (0.0356)$	$0.7047 \ (0.1190)$
125	Combined	$0.7998 \ (0.0517)$	$0.7571 \ (0.0457)$	$0.7053\ (0.0468)$	$0.5168 \ (0.0405)$	$0.6947 \ (0.1176)$
250	5	0.7586(0.0390)	0.7299(0.0360)	$0.6921 \ (0.0328)$	0.5159(0.0390)	$0.6741 \ (0.1014)$
250	10	0.7774(0.0297)	0.7428(0.0299)	0.7009(0.0300)	$0.5466 \ (0.0295)$	0.6919(0.0932)
250	15	$0.7939\ (0.0261)$	0.7567 (0.0295)	0.7043(0.0271)	0.5387 (0.0288)	$0.6984 \ (0.1016)$
250	Combined	$0.7766\ (0.0350)$	$0.7431 \ (0.0336)$	$0.6991 \ (0.0303)$	$0.5337 \ (0.0351)$	$0.6881 \ (0.0992)$

Table C.2.8: Average relative optimality gaps of Revised Anti 2-Greedy Search for the generated Euclidean instances.

C.3 Swap Local Search Results

In the following tables the details of the Swap Local Search performance for the generated Euclidean instances are given. Shown are the relative optimality gaps and the used number of iterations. We refer to Section 4.5 for the description of the instances, the Greedy Search variants (the initial solutions), and the Swap Local Search method.

	1-Swap Local Search after 1-Greedy Search								
		Av	erage (Standard I	Deviation) of Rela	tive Optimality G	aps			
	Opened		Maximum	Coverage					
Size	Bases	70%	80%	90%	100%	Combined			
125	5	$0.0084 \ (0.0137)$	$0.0118 \ (0.0155)$	0.0107 (0.0148)	0.0017(0.0041)	$0.0081 \ (0.0134)$			
125	10	0.0075(0.0092)	0.0122(0.0133)	0.0154(0.0141)	0.0046(0.0061)	0.0099(0.0118)			
125	15	$0.0070 \ (0.0087)$	$0.0089 \ (0.0094)$	$0.0122 \ (0.0085)$	$0.0051 \ (0.0066)$	$0.0083 \ (0.0087)$			
125	Combined	$0.0076 \ (0.0107)$	$0.0110 \ (0.0130)$	$0.0127 \ (0.0129)$	$0.0038 \ (0.0058)$	$0.0088 \ (0.0115)$			
250	5	0.0120(0.0132)	0.0162(0.0149)	0.0136(0.0169)	0.0008(0.0032)	$0.0107 \ (0.0143)$			
250	10	0.0118(0.0107)	0.0136(0.0119)	0.0155(0.0144)	0.0062(0.0069)	0.0118 (0.0117)			
250	15	$0.0093 \ (0.0084)$	$0.0169\ (0.0132)$	$0.0180\ (0.0104)$	$0.0064 \ (0.0044)$	$0.0127 \ (0.0108)$			
250	Combined	$0.0111 \ (0.0109)$	$0.0155 \ (0.0134)$	0.0157 (0.0142)	0.0045 (0.0057)	0.0117 (0.0124)			

	Opened		Maximum	Coverage		
Size	Bases	70%	80%	90%	100%	Combined
125	5	1.4600(0.7616)	2.0800(1.2262)	2.5400(1.3881)	1.9800(0.7420)	$2.0150 \ (1.1274)$
125	10	1.4200(0.8352)	2.0400(1.2447)	2.7800(1.6325)	2.7600(1.6107)	$2.2500 \ (1.4725)$
125	15	1.4600(0.8855)	2.0200(0.9998)	2.4800(1.4462)	2.6200(1.5504)	$2.1450\ (1.3239)$
125	Combined	$1.4467 \ (0.8236)$	$2.0467 \ (1.1547)$	$2.6000 \ (1.4882)$	$2.4533 \ (1.3931)$	$2.1367 \ (1.3169)$
250	5	1.9000(0.9530)	2.2800(1.2296)	2.6800(1.4061)	2.4000(0.9258)	$2.3150 \ (1.1716)$
250	10	1.9000(1.1824)	2.7600(1.5851)	3.6800(1.8674)	4.4200(1.8305)	$3.1900 \ (1.8846)$
250	15	1.9400(0.9564)	2.6000(1.3093)	4.5000 (1.9404)	3.3400 (1.8026)	$3.0950 \ (1.8117)$
250	Combined	$1.9133\ (1.0293)$	$2.5467 \ (1.3883)$	$3.6200 \ (1.8955)$	$3.3867 \ (1.7715)$	$2.8667 \ (1.6972)$

Table C.3.1: Average relative optimality gaps and iterations of 1-Swap Local Search after 1-Greedy Search for the generated Euclidean instances.

	1-Swap Local Search after 2-Greedy Search							
		$\mathbf{A}\mathbf{v}$	erage (Standard I	Deviation) of Rela	tive Optimality G	aps		
	Opened		Maximum	ı Coverage				
Size	Bases	70%	80%	90%	100%	Combined		
125	5	0.0073(0.0136)	0.0097(0.0131)	$0.0104 \ (0.0152)$	0.0024(0.0048)	$0.0075 \ (0.0126)$		
125	10	0.0069(0.0087)	0.0124(0.0135)	0.0148(0.0137)	0.0041(0.0058)	0.0096(0.0117)		
125	15	0.0070(0.0087)	0.0085(0.0089)	0.0120(0.0087)	0.0049(0.0060)	$0.0081 \ (0.0085)$		
125	Combined	$0.0071 \ (0.0105)$	$0.0102 \ (0.0120)$	$0.0124 \ (0.0129)$	$0.0038 \ (0.0056)$	$0.0084\ (0.0111)$		
250	5	0.0097 (0.0117)	$0.0184 \ (0.0150)$	$0.0145 \ (0.0173)$	0.0003 (0.0011)	$0.0107 \ (0.0144)$		
250	10	0.0117(0.0106)	0.0134(0.0113)	0.0158(0.0136)	0.0060(0.0063)	0.0118(0.0113)		
250	15	0.0094 (0.0081)	0.0162(0.0134)	$0.0183 \ (0.0105)$	0.0070(0.0042)	$0.0127 \ (0.0107)$		
250	Combined	$0.0102 \ (0.0102)$	$0.0160\ (0.0134)$	$0.0162\ (0.0141)$	$0.0044 \ (0.0053)$	$0.0117 \ (0.0123)$		

	Opened		Maximum	Coverage		
Size	Bases	70%	80%	90%	100%	Combined
125	5	1.4200(0.7025)	2.1400(1.3704)	2.5000(1.2817)	1.6400(0.7494)	$1.9250 \ (1.1428)$
125	10	1.4200(0.8352)	2.0000(1.2289)	2.8000(1.5779)	2.9200(1.7243)	$2.2850 \ (1.5050)$
125	15	1.4800(0.8862)	$1.9800\ (0.9998)$	2.5000(1.5152)	2.5400(1.6189)	$2.1250 \ (1.3559)$
125	Combined	$1.4400 \ (0.8067)$	$2.0400 \ (1.2034)$	$2.6000 \ (1.4609)$	$2.3667 \ (1.5213)$	$2.1117 \ (1.3487)$
250	5	1.7600(0.8704)	2.1200(1.1183)	2.7400(1.3372)	1.8200(0.7475)	2.1100(1.1064)
250	10	1.9600(1.1773)	2.5600(1.5800)	3.4200(1.8745)	4.2800(1.7266)	$3.0550 \ (1.8245)$
250	15	1.8800(0.9613)	2.6000(1.3851)	4.3800 (1.9155)	3.3200(1.9319)	$3.0450 \ (1.8384)$
250	Combined	$1.8667 \ (1.0078)$	$2.4267 \ (1.3823)$	$3.5133 \ (1.8455)$	$3.1400 \ (1.8502)$	$2.7367 \ (1.6829)$

Table C.3.2: Average relative optimality gaps and iterations of 1-Swap Local Search after 2-Greedy Search for the generated Euclidean instances.

	2-Swap Local Search after 1-Greedy Search							
		Av	erage (Standard I	Deviation) of Rela	tive Optimality G	aps		
	Opened		Maximum	Coverage				
Size	Bases	70%	80%	90%	100%	Combined		
125	5	$0.0016 \ (0.0059)$	$0.0021 \ (0.0053)$	$0.0030 \ (0.0067)$	0 (0)	$0.0017 \ (0.0053)$		
125	10	$0.0005 \ (0.0014)$	0.0019(0.0049)	$0.0024 \ (0.0050)$	$0.0005 \ (0.0015)$	$0.0013 \ (0.0037)$		
125	15	$0.0002 \ (0.0009)$	$0.0009 \ (0.0027)$	$0.0019 \ (0.0036)$	$0.0010 \ (0.0023)$	$0.0010 \ (0.0026)$		
125	Combined	$0.0008 \ (0.0036)$	$0.0016 \ (0.0044)$	$0.0024 \ (0.0053)$	$0.0005 \ (0.0016)$	$0.0013 \ (0.0040)$		
250	5	0.0018(0.0043)	0.0033 (0.0065)	0.0038(0.0048)	0 (0)	$0.0022 \ (0.0048)$		
250	10	0.0012(0.0031)	0.0033(0.0054)	$0.0042 \ (0.0063)$	0.0004(0.0012)	$0.0023 \ (0.0047)$		
250	15	0.0010(0.0024)	0.0027 (0.0043)	0.0049(0.0046)	0.0015(0.0020)	0.0025 (0.0038)		
250	Combined	0.0013 (0.0034)	$0.0031 \ (0.0055)$	0.0043 (0.0053)	$0.0006 \ (0.0015)$	0.0023 (0.0044)		

	Opened		Maximum	Coverage		
Size	Bases	70%	80%	90%	100%	Combined
125	5	1.7600(0.7160)	2.2600(0.8992)	2.5000(0.9313)	1.9600(0.5330)	$2.1200 \ (0.8301)$
125	10	2.0600(0.8430)	2.8000(1.2454)	3.4200(1.4581)	2.8000(1.2122)	2.7700(1.2944)
125	15	2.2400(1.0012)	2.9800(1.0784)	3.6000(1.2289)	3.0800(1.3223)	$2.9750 \ (1.2539)$
125	Combined	$2.0200 \ (0.8783)$	$2.6800 \ (1.1192)$	$3.1733 \ (1.3095)$	$2.6133 \ (1.1746)$	$2.6217 \ (1.2004)$
250	5	2.3200(0.7407)	2.8000(1.2778)	2.8200(1.2070)	2.0200(0.5147)	$2.4900 \ (1.0370)$
250	10	2.9600(1.1241)	3.4600(1.1643)	4.2400(1.7677)	3.8600(1.0882)	$3.6300 \ (1.3902)$
250	15	3.3200 (1.3468)	4.2200 (1.8103)	5.1400(1.8737)	4.2000 (1.4846)	$4.2200 \ (1.7541)$
250	Combined	$2.8667 \ (1.1682)$	$3.4933 \ (1.5490)$	$4.0667 \ (1.8919)$	$3.3600 \ (1.4576)$	$3.4467 \ (1.5931)$

Table C.3.3: Average relative optimality gaps and iterations of 2-Swap Local Search after 1-Greedy Search for the generated Euclidean instances.

	1-Swap Local Search after Reverse 1-Greedy Search							
		A	verage (Standard	Deviation) of Relat	ive Optimality Ga	ps		
	Opened		Maximum	Coverage				
Size	Bases	70%	80%	90%	100%	Combined		
125 125 125	5 10 15	$\begin{array}{c} 0.0100 \ (0.0191) \\ 0.0136 \ (0.0162) \\ 0.0122 \ (0.0146) \end{array}$	$\begin{array}{c} 0.0156 \ (0.0209) \\ 0.0181 \ (0.0184) \\ 0.0160 \ (0.0148) \end{array}$	$\begin{array}{c} 0.0115 \ (0.0160) \\ 0.0230 \ (0.0184) \\ 0.0241 \ (0.0188) \end{array}$	$\begin{array}{c} 0.0020 \ (0.0068) \\ 0.0060 \ (0.0077) \\ 0.0089 \ (0.0073) \end{array}$	$\begin{array}{c} 0.0098 \; (0.0172) \\ 0.0152 \; (0.0169) \\ 0.0153 \; (0.0154) \end{array}$		
125	Combined	$0.0120 \ (0.0167)$	$0.0166\ (0.0181)$	$0.0195 \ (0.0185)$	$0.0056\ (0.0078)$	$0.0134\ (0.0167)$		
$250 \\ 250 \\ 250 \\ 250$	$5\\10\\15$	$\begin{array}{c} 0.0134 \ (0.0178) \\ 0.0141 \ (0.0132) \\ 0.0139 \ (0.0137) \end{array}$	$\begin{array}{c} 0.0160 \ (0.0154) \\ 0.0200 \ (0.0149) \\ 0.0146 \ (0.0102) \end{array}$	$\begin{array}{c} 0.0166 \ (0.0172) \\ 0.0215 \ (0.0122) \\ 0.0208 \ (0.0106) \end{array}$	$\begin{array}{c} 0.0012 \ (0.0028) \\ 0.0064 \ (0.0075) \\ 0.0082 \ (0.0059) \end{array}$	$\begin{array}{c} 0.0118 \ (0.0158) \\ 0.0155 \ (0.0135) \\ 0.0144 \ (0.0113) \end{array}$		
250	Combined	$0.0138 \ (0.0149)$	$0.0168\ (0.0138)$	$0.0196 \ (0.0137)$	$0.0053 \ (0.0064)$	$0.0139 \ (0.0137)$		

205

	Opened		Maximum	Coverage		
Size	Bases	70%	80%	90%	100%	Combined
125	5	5.5200(1.3130)	5.6200(1.1045)	5.7000(1.3740)	2.8800(0.9613)	$4.9300 \ (1.6820)$
125	10	7.6800(2.0348)	7.4000(2.0203)	7.1600(2.1224)	4.1200 (1.3037)	6.5900(2.3728)
125	15	7.8200(2.2560)	8.6400(2.4641)	8.0200(2.3516)	4.3000 (1.7291)	$7.1950\ (2.7831)$
125	Combined	$7.0067 \ (2.1721)$	$7.2200\ (2.2993)$	$6.9600 \ (2.2009)$	$3.7667 \ (1.4991)$	$6.2383 \ (2.5104)$
250	5	6.0800(1.2095)	6.1800(1.4525)	6.5200(1.7640)	3.3000(1.1650)	5.5200(1.9126)
250	10	9.5000(1.5551)	9.4600(1.9400)	9.5800(2.2505)	6.1800(2.0373)	8.6800(2.4263)
250	15	12.1600 (2.1699)	12.6200 (2.7913)	12.1800(2.3963)	7.1400 (2.0305)	11.0250 (3.2554)
250	Combined	9.2467 (3.0099)	9.4200(3.3843)	9.4267 (3.1566)	5.5400 (2.4181)	8.4083 (3.4334)

Table C.3.4: Average relative optimality gaps and iterations of 1-Swap Local Search after Reverse 1-Greedy Search for the generated Euclidean instances.

	1-Swap Local Search after Reverse 2-Greedy Search							
		A	verage (Standard	Deviation) of Relat	ive Optimality Gap	ps		
	Opened		Maximum	Coverage				
Size	Bases	70%	80%	90%	100%	Combined		
125	5	0.0094(0.0188)	0.0168(0.0220)	$0.0111 \ (0.0155)$	$0.0020 \ (0.0068)$	$0.0099 \ (0.0175)$		
125	10	0.0136(0.0162)	0.0173(0.0178)	0.0230(0.0184)	0.0061 (0.0077)	$0.0150 \ (0.0167)$		
125	15	$0.0122 \ (0.0146)$	$0.0160\ (0.0148)$	$0.0241 \ (0.0188)$	$0.0089 \ (0.0073)$	$0.0153 \ (0.0154)$		
125	Combined	$0.0118 \ (0.0166)$	$0.0167 \ (0.0183)$	$0.0194\ (0.0185)$	$0.0057 \ (0.0077)$	$0.0134 \ (0.0167)$		
250	5	0.0135(0.0179)	$0.0160 \ (0.0154)$	0.0169(0.0173)	0.0012 (0.0028)	$0.0119 \ (0.0159)$		
250	10	0.0143(0.0131)	0.0195(0.0141)	$0.0221 \ (0.0120)$	$0.0061 \ (0.0071)$	$0.0155\ (0.0133)$		
250	15	0.0139 (0.0137)	0.0149(0.0102)	0.0213 (0.0103)	0.0084(0.0060)	$0.0146\ (0.0113)$		
250	Combined	0.0139 (0.0149)	$0.0168\ (0.0135)$	0.0201 (0.0136)	$0.0052 \ (0.0063)$	0.0140 (0.0137)		

Ν	٥
\subset	2
C	5

	Opened		Maximum	Coverage		
Size	Bases	70%	80%	90%	100%	Combined
125	5	5.5200(1.3130)	5.6000(1.1249)	5.7000(1.5152)	2.9000(1.0926)	$4.9300 \ (1.7263)$
125	10	7.6800(2.0348)	7.4400(1.9914)	7.1600(2.1224)	4.1000(1.2976)	$6.5950 \ (2.3747)$
125	15	7.8200(2.2560)	8.6400(2.4641)	8.0200(2.3516)	4.3000(1.7291)	$7.1950\ (2.7831)$
125	Combined	$7.0067\ (2.1721)$	$7.2267\ (2.3001)$	$6.9600 \ (2.2312)$	$3.7667 \ (1.5213)$	$6.2400 \ (2.5212)$
250	5	6.1000(1.2330)	6.1800(1.4525)	6.4000(1.6660)	3.2800(1.1436)	$5.4900 \ (1.8835)$
250	10	9.4800(1.5418)	9.5200(1.9403)	9.5000(2.1500)	6.0600(2.0042)	$8.6400 \ (2.4228)$
250	15	12.1400(2.1760)	12.4800(2.6668)	12.1400(2.3300)	7.0400(2.0894)	$10.9500 \ (3.2357)$
250	Combined	$9.2400 \ (2.9982)$	$9.3933 \ (3.3073)$	$9.3467 \ (3.1238)$	$5.4600 \ (2.3958)$	$8.3600 \ (3.4093)$

Table C.3.5: Average relative optimality gaps and iterations of 1-Swap Local Search after Reverse 2-Greedy Search for the generated Euclidean instances.

	2-Swap Local Search after Reverse 1-Greedy Search							
		\mathbf{Av}	erage (Standard I	Deviation) of Rela	tive Optimality G	aps		
	Opened		Maximum	Coverage				
Size	Bases	70%	80%	90%	100%	Combined		
125	5	0.0013(0.0048)	$0.0044 \ (0.0091)$	0.0017(0.0041)	0 (0)	$0.0019 \ (0.0057)$		
125	10	0.0002(0.0009)	0.0024(0.0061)	0.0033(0.0068)	0.0008(0.0026)	0.0017 (0.0049)		
125	15	$0.0001 \ (0.0005)$	$0.0012 \ (0.0028)$	$0.0035 \ (0.0054)$	$0.0013 \ (0.0025)$	$0.0015 \ (0.0035)$		
125	Combined	$0.0005 \ (0.0029)$	$0.0027 \ (0.0066)$	$0.0028 \ (0.0056)$	$0.0007 \ (0.0021)$	$0.0017 \ (0.0048)$		
250	5	$0.0021 \ (0.0045)$	0.0039(0.0065)	$0.0034 \ (0.0054)$	$0.0001 \ (0.0006)$	$0.0024 \ (0.0050)$		
250	10	0.0021(0.0036)	0.0032(0.0048)	0.0069(0.0089)	0.0009(0.0021)	0.0033 (0.0059)		
250	15	0.0013 (0.0034)	0.0018(0.0027)	0.0056(0.0058)	0.0014 (0.0022)	$0.0025 \ (0.0042)$		
250	Combined	$0.0018 \ (0.0039)$	$0.0029 \ (0.0049)$	$0.0053 \ (0.0070)$	$0.0008 \ (0.0019)$	$0.0027 \ (0.0051)$		

2
\subset
\neg

	Opened		Maximum	Coverage		
Size	Bases	70%	80%	90%	100%	Combined
125	5	3.7800(0.8401)	3.9600(0.8797)	3.9000(0.8864)	2.2600(0.4870)	$3.4750 \ (1.0559)$
125	10	5.4000(1.0498)	5.5800(1.2631)	5.7600(1.4923)	3.5000(1.0152)	$5.0600 \ (1.5159)$
125	15	5.7600(1.3180)	6.4000(1.4142)	6.6600(1.7798)	4.1200(1.2229)	$5.7350\ (1.7464)$
125	Combined	$4.9800 \ (1.3829)$	$5.3133\ (1.5721)$	$5.4400\ (1.8336)$	$3.2933 \ (1.2291)$	$4.7567 \ (1.7452)$
250	5	4.2800(1.0309)	4.3800(0.9452)	4.7000(0.9530)	2.3600(0.5980)	$3.9300 \ (1.2819)$
250	10	6.4000(1.1606)	6.7000(1.1995)	6.6600(1.4086)	4.6200(1.2599)	$6.0950 \ (1.5190)$
250	15	8.0800(1.5097)	8.6200 (1.9886)	8.8400 (1.7066)	5.7600(1.6850)	7.8250 (2.1111)
250	Combined	$6.2533 \ (1.9939)$	$6.5667 \ (2.2567)$	6.7333 (2.1880)	$4.2467 \ (1.8929)$	5.9500(2.3101)

Table C.3.6: Average relative optimality gaps and iterations of 2-Swap Local Search after Reverse 1-Greedy Search for the generated Euclidean instances.

	1-Swap Local Search after Anti 1-Greedy Search							
		A	verage (Standard	Deviation) of Relat	ive Optimality Gap	ps		
	Opened		Maximum	Coverage				
Size	Bases	70%	80%	90%	100%	Combined		
125	5	$0.0103 \ (0.0174)$	0.0149(0.0181)	0.0139(0.0174)	$0.0008 \ (0.0029)$	$0.0100 \ (0.0162)$		
125	10	0.0073 (0.0092)	0.0121 (0.0130)	0.0194(0.0184)	$0.0061 \ (0.0071)$	$0.0112 \ (0.0137)$		
125	15	$0.0073 \ (0.0093)$	$0.0086 \ (0.0085)$	$0.0149 \ (0.0114)$	$0.0073 \ (0.0062)$	$0.0095 \ (0.0095)$		
125	Combined	$0.0083 \ (0.0126)$	$0.0119 \ (0.0139)$	$0.0161 \ (0.0161)$	$0.0047 \ (0.0063)$	$\boldsymbol{0.0102}~(\boldsymbol{0.0134})$		
250	5	0.0123(0.0131)	$0.0171 \ (0.0205)$	$0.0155 \ (0.0167)$	$0.0007 \ (0.0016)$	$0.0114 \ (0.0160)$		
250	10	0.0129(0.0124)	0.0145(0.0120)	0.0182(0.0142)	0.0058(0.0052)	$0.0128 \ (0.0123)$		
250	15	0.0090(0.0078)	0.0163(0.0131)	0.0190(0.0111)	0.0075(0.0054)	$0.0130 \ (0.0109)$		
250	Combined	$0.0114 \ (0.0114)$	$0.0159 \ (0.0156)$	$0.0176 \ (0.0142)$	$0.0047 \ (0.0053)$	$0.0124 \ (0.0132)$		

2)
\subset)
α)

		Average (Standard Deviation) of Used Iterations					
	Opened		Maximum	Coverage			
Size	Bases	70%	80%	90%	100%	Combined	
125	5	7.0000 (1.1429)	7.3400(1.3032)	7.6800(1.4630)	5.9200(0.8999)	$6.9850\ (1.3800)$	
125	10	11.8000(1.0880)	12.2600(1.6264)	12.6200(1.4412)	10.8200(1.5211)	$11.8750\ (1.5754)$	
125	15	$16.5200\ (1.0150)$	17.0200(1.2856)	17.1800(1.6499)	14.1600(1.9311)	$16.2200\ (1.9314)$	
125	Combined	$11.7733 \ (4.0453)$	$12.2067 \ (4.2066)$	$12.4933 \ (4.1753)$	$10.3000 \ (3.7124)$	$11.6933 \ (4.1171)$	
250	5	7.1800(1.4665)	7.9000(1.7291)	8.0800(1.5628)	6.4600(0.9733)	$7.4050 \ (1.5854)$	
250	10	12.4400(1.5274)	13.3000(1.8654)	13.7800(2.1313)	13.0200(1.6841)	$13.1350\ (1.8666)$	
250	15	17.8400(1.6081)	18.2000(1.7261)	$19.4000 \ (2.3123)$	16.2800(1.6418)	$17.9300\ (2.1443)$	
250	Combined	$12.4867 \ (4.6252)$	$13.1333\ (4.5740)$	$\boldsymbol{13.7533}~(5.0553)$	$11.9200 \ (4.3497)$	$12.8233 \ (4.6970)$	

Table C.3.7: Average relative optimality gaps and iterations of 1-Swap Local Search after Anti 1-Greedy Search for the generated Euclidean instances.

	1-Swap Local Search after Anti 2-Greedy Search							
		A	verage (Standard	Deviation) of Relat	ive Optimality Gap	ps		
	Opened		Maximum	Coverage				
Size	Bases	70%	80%	90%	100%	Combined		
$125 \\ 125 \\ 125 \\ 125$	5 10 15	$\begin{array}{c} 0.0093 \ (0.0152) \\ 0.0070 \ (0.0089) \\ 0.0069 \ (0.0087) \end{array}$	$\begin{array}{c} 0.0139 \ (0.0161) \\ 0.0110 \ (0.0123) \\ 0.0089 \ (0.0093) \end{array}$	$\begin{array}{c} 0.0133 \ (0.0174) \\ 0.0172 \ (0.0168) \\ 0.0130 \ (0.0109) \end{array}$	$\begin{array}{c} 0.0011 \ (0.0030) \\ 0.0062 \ (0.0066) \\ 0.0062 \ (0.0066) \end{array}$	$\begin{array}{c} 0.0094 \; (0.0149) \\ 0.0104 \; (0.0125) \\ 0.0087 \; (0.0093) \end{array}$		
125	Combined	$0.0077 \ (0.0113)$	$0.0113 \ (0.0130)$	$0.0145\ (0.0153)$	$0.0045 \ (0.0061)$	$0.0095 \ (0.0125)$		
$250 \\ 250 \\ 250 \\ 250$	$5\\10\\15$	$\begin{array}{c} 0.0124 \ (0.0130) \\ 0.0119 \ (0.0112) \\ 0.0089 \ (0.0075) \end{array}$	$\begin{array}{c} 0.0163 \ (0.0199) \\ 0.0151 \ (0.0118) \\ 0.0163 \ (0.0127) \end{array}$	$\begin{array}{c} 0.0171 \ (0.0180) \\ 0.0167 \ (0.0148) \\ 0.0186 \ (0.0119) \end{array}$	$\begin{array}{c} 0.0006 \ (0.0016) \\ 0.0054 \ (0.0057) \\ 0.0069 \ (0.0046) \end{array}$	$\begin{array}{c} 0.0116 \ (0.0162) \\ 0.0123 \ (0.0121) \\ 0.0127 \ (0.0109) \end{array}$		
250	Combined	$0.0110 \ (0.0109)$	$0.0159\ (0.0151)$	$0.0175 \ (0.0150)$	$0.0043 \ (0.0051)$	$0.0122\ (0.0132)$		

209

		Average (Standard Deviation) of Used Iterations					
	Opened		Maximum	a Coverage			
Size	Bases	70%	80%	90%	100%	Combined	
125	5	6.6800(0.9134)	7.3600(1.3962)	7.5600(1.4591)	6.0200(1.0200)	$6.9050\ (1.3547)$	
125	10	11.6600(1.0422)	12.3200(1.4348)	13.0400(2.0199)	10.9400(1.5039)	$11.9900 \ (1.7160)$	
125	15	$16.6400 \ (1.1021)$	17.0200(1.1156)	$16.9200 \ (1.6015)$	14.3800(1.9784)	$16.2400\ (1.8383)$	
125	Combined	$11.6600 \ (4.2042)$	$12.2333 \ (4.1699)$	$12.5067 \ (4.2105)$	$10.4467 \ (3.7712)$	$11.7117 \ (4.1590)$	
250	5	7.4000(1.4708)	7.5000(1.4743)	8.1000(1.6444)	6.5400(0.9304)	$7.3850\ (1.5027)$	
250	10	12.4200(1.5265)	13.1400(1.6289)	13.6800(2.3250)	13.1600(1.6458)	$13.1000 \ (1.8513)$	
250	15	$17.7000 \ (1.5152)$	18.2000(1.7143)	19.4200(2.3043)	16.5000(1.7291)	$17.9550\ (2.1060)$	
250	Combined	$12.5067 \ (4.4763)$	$12.9467\ (4.6672)$	$13.7333\ (5.0907)$	$12.0667 \ (4.4051)$	$12.8133 \ (4.6963)$	

Table C.3.8: Average relative optimality gaps and iterations of 1-Swap Local Search after Anti 2-Greedy Search for the generated Euclidean instances.

	2-Swap Local Search after Anti 1-Greedy Search							
		A	verage (Standard	Deviation) of Relat	ive Optimality Gap	ps		
	Opened		Maximum	Coverage				
Size	Bases	70%	80%	90%	100%	Combined		
125	5	$0.0011 \ (0.0050)$	0.0052 (0.0128)	0.0023(0.0069)	0 (0)	$0.0022 \ (0.0079)$		
125	10	0.0005(0.0014)	0.0016(0.0044)	0.0025(0.0053)	0.0006(0.0020)	$0.0013 \ (0.0037)$		
125	15	$0.0002 \ (0.0009)$	$0.0007 \ (0.0015)$	$0.0032 \ (0.0054)$	$0.0008 \ (0.0017)$	$0.0012 \ (0.0032)$		
125	Combined	$0.0006 \ (0.0031)$	$0.0025 \ (0.0080)$	$0.0027 \ (0.0059)$	$0.0005 \ (0.0015)$	$0.0016 \ (0.0054)$		
250	5	0.0015(0.0040)	$0.0021 \ (0.0049)$	$0.0043 \ (0.0065)$	0 (0)	$0.0020 \ (0.0048)$		
250	10	0.0016(0.0040)	0.0034(0.0057)	0.0044(0.0067)	0.0010(0.0025)	$0.0026 \ (0.0052)$		
250	15	0.0010(0.0023)	0.0037 (0.0062)	0.0049 (0.0053)	0.0016 (0.0022)	$0.0028 \ (0.0046)$		
250	Combined	$0.0014 \ (0.0035)$	$0.0031 \ (0.0057)$	$0.0045 \ (0.0062)$	$0.0008 \ (0.0020)$	$0.0025 \ (0.0049)$		

t	0
ŀ	-
¢	

	Opened Maximum Coverage						
Size	Bases	70%	80%	90%	100%	Combined	
125	5	4.4200(0.7309)	4.7400 (0.8033)	4.8600 (0.9260)	3.7400(0.5997)	$4.4400 \ (0.8833)$	
125	10	7.2600(0.8992)	7.7200 (1.1787)	8.5200(1.6811)	6.9000(0.9530)	$7.6000 \ (1.3525)$	
125	15	$9.8000\ (0.9035)$	10.2400(1.0606)	10.8400(1.2993)	8.8800(1.3649)	$9.9400 \ (1.3659)$	
125	Combined	$7.1600 \ (2.3604)$	$7.5667 \ (2.4752)$	$8.0733 \ (2.8048)$	$6.5067 \ (2.3538)$	$7.3267 \ (2.5640)$	
250	5	4.7800 (0.7637)	5.3000(1.3286)	5.2800(0.9697)	4.0800(0.4882)	$4.8600 \ (1.0564)$	
250	10	8.2600(1.1920)	8.4400(1.1634)	8.9200(1.6517)	8.2600(1.2423)	$8.4700\ (1.3447)$	
250	15	$10.9800 \ (1.2204)$	11.7600(1.8905)	12.6400(1.8709)	10.6400(1.3815)	$11.5050\ (1.7821)$	
250	Combined	$8.0067 \ (2.7624)$	$8.5000 \ (3.0339)$	$8.9467 \ (3.3833)$	$7.6600 \ (2.9352)$	$8.2783 \ (3.0686)$	

Table C.3.9: Average relative optimality gaps and iterations of 2-Swap Local Search after Anti 1-Greedy Search for the generated Euclidean instances.

1-Swap Local Search after Revised Anti 1-Greedy Search							
		A	verage (Standard	Deviation) of Relat	ive Optimality Ga	ps	
	Opened		Maximum	Coverage			
Size	Bases	70%	80%	90%	100%	Combined	
125 125 125	$5\\10\\15$	$\begin{array}{c} 0.0097 \ (0.0180) \\ 0.0074 \ (0.0091) \\ 0.0064 \ (0.0081) \end{array}$	$\begin{array}{c} 0.0144 \; (0.0178) \\ 0.0128 \; (0.0125) \\ 0.0095 \; (0.0095) \end{array}$	$\begin{array}{c} 0.0191 \ (0.0210) \\ 0.0192 \ (0.0147) \\ 0.0123 \ (0.0095) \end{array}$	$\begin{array}{c} 0.0027 \ (0.0046) \\ 0.0082 \ (0.0071) \\ 0.0086 \ (0.0084) \end{array}$	$\begin{array}{c} 0.0115 \; (0.0176) \\ 0.0119 \; (0.0121) \\ 0.0092 \; (0.0091) \end{array}$	
125	Combined	$0.0078 \ (0.0126)$	$0.0123\ (0.0138)$	$0.0168 \ (0.0160)$	$0.0065 \ (0.0074)$	$0.0109 \ (0.0134)$	
$250 \\ 250 \\ 250 \\ 250$	$5\\10\\15$	$\begin{array}{c} 0.0112 \ (0.0130) \\ 0.0129 \ (0.0118) \\ 0.0094 \ (0.0083) \end{array}$	$\begin{array}{c} 0.0138 \ (0.0133) \\ 0.0150 \ (0.0128) \\ 0.0163 \ (0.0130) \end{array}$	$\begin{array}{c} 0.0143 \ (0.0151) \\ 0.0202 \ (0.0129) \\ 0.0228 \ (0.0113) \end{array}$	$\begin{array}{c} 0.0021 \ (0.0046) \\ 0.0063 \ (0.0063) \\ 0.0080 \ (0.0059) \end{array}$	$\begin{array}{c} 0.0104 \; (0.0131) \\ 0.0136 \; (0.0123) \\ 0.0141 \; (0.0115) \end{array}$	
250	Combined	$0.0112 \ (0.0112)$	$0.0150\ (0.0130)$	$0.0191 \ (0.0136)$	$0.0055 \ (0.0061)$	$0.0127 \ (0.0124)$	

211

	Opened		Maximum	Coverage		
Size	Bases	70%	80%	90%	100%	Combined
125	5	7.1200(1.2061)	7.1400(1.1430)	7.5600(1.7398)	5.8600(1.0882)	$6.9200 \ (1.4575)$
125	10	11.7600(0.9806)	12.3600(1.4535)	12.6400(1.5877)	9.5600(1.6056)	$11.5800 \ (1.8657)$
125	15	$16.5600 \ (0.9071)$	$17.0000 \ (0.9897)$	$16.8000 \ (1.6288)$	12.4600(1.9609)	$15.7050\ (2.3656)$
125	Combined	$11.8133 \ (4.0023)$	$12.1667 \ (4.2163)$	$12.3333 \ (4.1315)$	$9.2933 \ (3.1379)$	$11.4017\ (4.0769)$
250	5	7.4200 (1.4010)	7.5600(1.4450)	8.4400(1.5407)	6.4600(0.9941)	$7.4700 \ (1.5234)$
250	10	12.3800(1.1933)	13.0800(1.5497)	14.0200(1.9430)	11.7000(1.5419)	$12.7950 \ (1.7886)$
250	15	17.9200(1.6142)	18.4600(1.7752)	19.4200(2.8578)	14.9800(1.9534)	$17.6950 \ (2.6698)$
250	Combined	$12.5733 \ (4.5264)$	$13.0333 \ (4.7380)$	$13.9600 \ (4.9938)$	11.0467 (3.8414)	$12.6533 \ (4.6547)$

Table C.3.10: Average relative optimality gaps and iterations of 1-Swap Local Search after Revised Anti 1-Greedy Search for the generated Euclidean instances.

	1-Swap Local Search after Revised Anti 2-Greedy Search							
		A	verage (Standard	Deviation) of Relat	ive Optimality Gaj	ps		
	Opened		Maximum	Coverage				
Size	Bases	70%	80%	90%	100%	Combined		
125	5	$0.0088 \ (0.0178)$	0.0139(0.0181)	0.0173(0.0192)	0.0026(0.0049)	$0.0107 \ (0.0169)$		
125	10	0.0074(0.0092)	$0.0131 \ (0.0130)$	$0.0194\ (0.0144)$	0.0077(0.0083)	$0.0119 \ (0.0124)$		
125	15	$0.0065\ (0.0081)$	$0.0083 \ (0.0084)$	$0.0139 \ (0.0094)$	$0.0093 \ (0.0074)$	$0.0095 \ (0.0087)$		
125	Combined	$0.0076 \ (0.0124)$	$0.0118 \ (0.0139)$	$0.0169 \ (0.0150)$	$0.0065 \ (0.0075)$	$0.0107 \ (0.0131)$		
250	5	$0.0108 \ (0.0126)$	$0.0146\ (0.0151)$	0.0163(0.0181)	$0.0028 \ (0.0056)$	$0.0111 \ (0.0145)$		
250	10	0.0119(0.0104)	$0.0161 \ (0.0121)$	0.0172(0.0124)	0.0072(0.0063)	$0.0131 \ (0.0112)$		
250	15	$0.0096\ (0.0081)$	$0.0180\ (0.0135)$	$0.0232 \ (0.0135)$	$0.0091 \ (0.0062)$	$0.0150 \ (0.0123)$		
250	Combined	$0.0108 \ (0.0105)$	0.0162 (0.0136)	$0.0189 \ (0.0151)$	0.0064 (0.0066)	$0.0131 \ (0.0128)$		

Þ	٢
H	-
Þ	٢

	Opened		Maximum	Coverage		
Size	Bases	70%	80%	90%	100%	Combined
125	5	7.1000(1.1650)	7.2400(1.2382)	7.6800(1.6714)	5.8600(1.1430)	$6.9700 \ (1.4765)$
125	10	$11.6800 \ (0.9570)$	12.4000(1.4142)	12.4400(1.5274)	9.8400(1.3149)	$11.5900 \ (1.6841)$
125	15	$16.6000 \ (0.9258)$	$17.1200\ (0.9823)$	17.2200(1.7177)	12.5200(1.6811)	$15.8650\ (2.3823)$
125	Combined	$11.7933 \ (4.0222)$	$12.2533 \ (4.2271)$	$12.4467 \ (4.2340)$	$9.4067 \ (3.0764)$	$11.4750 \ (4.0948)$
250	5	7.3400(1.4086)	7.6000(1.7023)	8.4200(1.7154)	6.6400(1.2081)	$7.5000 \ (1.6412)$
250	10	12.5000(1.3439)	13.1400(1.5254)	14.1600(2.4105)	11.5800(1.5791)	$12.8450 \ (1.9876)$
250	15	$17.8800 \ (1.2558)$	18.4800(1.8210)	$19.2000 \ (2.9898)$	$15.0800 \ (1.8052)$	$17.6600 \ (2.5802)$
250	Combined	$12.5733 \ (4.5174)$	$13.0733\ (4.7616)$	$13.9267 \ (5.0342)$	$11.1000 \ (3.7999)$	$12.6683 \ (4.6549)$

Table C.3.11: Average relative optimality gaps and iterations of 1-Swap Local Search after Revised Anti 2-Greedy Search for the generated Euclidean instances.

	2-Swap Local Search after Revised Anti 1-Greedy Search							
		A	verage (Standard I	Deviation) of Relat	ive Optimality Ga	ps		
	Opened		Maximum	Coverage				
Size	Bases	70%	80%	90%	100%	Combined		
125	5	$0.0011 \ (0.0043)$	$0.0043 \ (0.0077)$	$0.0042 \ (0.0078)$	0 (0)	$0.0024 \ (0.0062)$		
125	10	$0.0004 \ (0.0014)$	$0.0017 \ (0.0039)$	$0.0032 \ (0.0062)$	$0.0007 \ (0.0018)$	$0.0015 \ (0.0040)$		
125	15	$0.0002 \ (0.0009)$	$0.0007 \ (0.0019)$	$0.0015 \ (0.0032)$	0.0009(0.0022)	$0.0008 \ (0.0022)$		
125	Combined	$0.0006 \ (0.0027)$	$0.0022 \ (0.0053)$	$0.0030 \ (0.0061)$	$0.0005 \ (0.0017)$	$0.0016 \ (0.0045)$		
250	5	$0.0020 \ (0.0045)$	$0.0034 \ (0.0066)$	$0.0039 \ (0.0065)$	0 (0)	$0.0023 \ (0.0053)$		
250	10	$0.0013 \ (0.0036)$	$0.0034 \ (0.0065)$	$0.0059 \ (0.0072)$	0.0013 (0.0024)	$0.0030 \ (0.0056)$		
250	15	$0.0010 \ (0.0022)$	$0.0031 \ (0.0054)$	$0.0047 \ (0.0055)$	$0.0017 \ (0.0016)$	$0.0026 \ (0.0043)$		
250	Combined	$0.0014 \ (0.0036)$	$0.0033 \ (0.0061)$	$0.0048 \ (0.0065)$	0.0010 (0.0018)	$0.0026 \ (0.0051)$		

	Opened		Maximum	Coverage		
Size	Bases	70%	80%	90%	100%	Combined
125	5	4.3400(0.5928)	4.6200(0.7530)	5.0200(1.0200)	3.6200(0.5675)	$4.4000 \ (0.9078)$
125	10	7.2200(0.9100)	7.8400(1.4337)	7.9000 (1.3740)	6.4000(0.9897)	7.3400(1.3354)
125	15	9.7400(0.7775)	10.3600(1.1563)	10.3600(1.2415)	8.1600(1.3456)	$9.6550 \ (1.4548)$
125	Combined	$7.1000\ (2.3423)$	$7.6067 \ (2.6187)$	$7.7600 \ (2.5028)$	$6.0600 \ (2.1307)$	$7.1317\ (2.4902)$
250	5	4.8600(0.8084)	5.0800(1.3377)	5.6200(1.1586)	4.0200(0.5529)	$4.8950 \ (1.1579)$
250	10	8.2800(1.2129)	8.2800(1.1613)	9.0200(1.5318)	7.2200(1.0746)	8.2000(1.4035)
250	15	10.9800(1.3013)	11.6200(1.6149)	12.7600(1.6358)	9.6200(1.3076)	$11.2450 \ (1.8526)$
250	Combined	$8.0400 \ (2.7512)$	$8.3267 \ (3.0112)$	$9.1333 \ (3.2639)$	6.9533 (2.5180)	$8.1133 \ (2.9959)$

Table C.3.12: Average relative optimality gaps and iterations of 2-Swap Local Search after Revised Anti 1-Greedy Search for the generated Euclidean instances.

Bibliography

- Ageev, A. and M. Sviridenko (1999). "Approximation algorithms for maximum coverage and max cut with given sizes of parts". *Integer Programming and Combinatorial Optimization*. Ed. by G. Cornuéjols, R. Burkard, and G. Woeginger. Vol. 1610. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 17–30.
- Alsalloum, O. and G. Rand (2006). "Extensions to emergency vehicle location models". Computers & Operations Research 33 (9), pp. 2725–2743.
- Amiri, A. (2001). "The multi-hour service system design problem". European Journal of Operational Research 128 (3), pp. 625–638.
- Ashwinkumar, B. and J. Vondrák (2014). "Fast algorithms for maximizing submodular functions". Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms.
- Ball, M. and F. Lin (1993). "A reliability model applied to emergency service vehicle location". Operations Research 41 (1), pp. 18–36.
- Baron, O., J. Milner, and H. Naseraldin (2011). "Facility location: a robust optimization approach". Production and Operations Management 20 (5), pp. 772–785.
- Başar, A., B. Çatay, and T. Ünlüyurt (2009). "A new model and tabu search approach for planning the emergency service stations". *Operations Research Proceedings 2008.* Ed. by B. Fleischmann, K.-H. Borgwardt, R. Klein, and A. Tuma. Vol. 2008. Springer Berlin Heidelberg, pp. 41–46.
- Batta, R., J. Dolan, and N. Krishnamurthy (1989). "The maximal expected covering location problem: revisited". *Transportation Science* 23 (4), pp. 277–287.
- Batta, R. and N. Mannur (1990). "Covering-location models for emergency situations that require multiple response units". *Management Science* 36 (1), pp. 16–23.
- Ben-Tal, A., B. Chung, S. Mandala, and T. Yao (2011). "Robust optimization for emergency logistics planning: risk mitigation in humanitarian relief supply chains". *Transportation Research Part B: Methodological* 45 (8), pp. 1177–1189.
- Ben-Tal, A., L. El Ghaoui, and A. Nemirovski (2009). Robust optimization. Princeton Series in Applied Mathematics. Princeton University Press.
- Beraldi, P. and M. Bruni (2009). "A probabilistic model applied to emergency service vehicle location". European Journal of Operational Research 196 (1), pp. 323–331.
- Beraldi, P., M. Bruni, and D. Conforti (2004). "Designing robust emergency medical service via stochastic programming". European Journal of Operational Research 158 (1), pp. 183–193.
- Berman, O., Z. Drezner, and D. Krass (2010). "Generalized coverage: new developments in covering location models". Computational Operations Research 37 (10), pp. 1675–1687.
- Berman, O., Z. Drezner, and D. Krass (2011). "Discrete cooperative covering problems". Journal of the Operational Research Society 62 (11), pp. 2002–2012.
- Bertsimas, D., D. Brown, and C. Caramanis (2011). "Theory and applications of robust optimization". SIAM Review 53 (3), pp. 464–501.

- Boers, I., P. Duijf, E. Grummels, G. Leerkes, and H. Van der Werff (2010). Ambulances in-zicht 2009. Ambulancezorg Nederland.
- Borrás, F. and J. Pastor (2002). "The ex-post evaluation of the minimum local reliability level: an enhanced probabilistic location set covering model". Annals of Operations Research 111 (1-4), pp. 51–74.
- Brotcorne, L., G. Laporte, and F. Semet (2003). "Ambulance location and relocation models". European Journal of Operational Research 147 (3), pp. 451–463.
- Burwell, T., J. Jarvis, and M. McKnew (1993). "Modeling co-located servers and dispatch ties in the hypercube model". *Computers & Operations Research* 20 (2), pp. 113–119.
- Church, R. and R. Gerrard (2003). "The multi-level location set covering model". *Geographical Analysis* 35 (4), pp. 277–289.
- Church, R. and C. ReVelle (1974). "The maximal covering locational problem". Papers In Regional Science 32 (1), pp. 101–108.
- Conforti, M. and G. Cornuéjols (1984). "Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the Rado-Edmonds theorem". *Discrete Applied Mathematics* 7 (3), pp. 251–274.
- Cook, S. (1971). "The complexity of theorem proving procedures". Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, pp. 151–158.
- Cornuéjols, G., M. Fisher, and G. Nemhauser (1977). "Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms". *Management Science* 23 (8), pp. 789–810.
- Dantzig, G. and P. Wolfe (1960). "Decomposition principle for linear programs". Operations Research 8 (1), pp. 101–111.
- Daskin, M. (1983). "A maximal expected covering location model: formulation, properties, and heuristic solution". Transportation Science 17 (1), pp. 48–69.
- Daskin, M. and E. Stern (1981). "A hierarchical objective set covering model for emergency medical service vehicle deployment". *Transportation Science* 15 (2), pp. 137–152.
- Doerner, K., W. Gutjahr, R. Hartl, M. Karall, and M. Reimann (2005). "Heuristic solution of an extended double coverage ambulance location problem for Austria". *Central European Journal* of Operations Research 13 (4), pp. 325–340.
- Erkut, E., A. Ingolfsson, and G. Erdoğan (2008). "Ambulance location for maximum survival". Naval Research Logistics 55 (1), pp. 42–58.
- Farahani, R., N. Asgari, N. Heidari, M. Hosseininia, and M. Goh (2012). "Covering problems in facility location: a review". Computers & Industrial Engineering 62 (1), pp. 368–407.
- Feige, U. (1998). "A threshold of ln n for approximating set cover". *Journal of the ACM* 45 (4), pp. 634–652.
- Feige, U., V. Mirrokni, and J. Vondrák (2007). "Maximizing non-monotone submodular functions". Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science.
- Gabrel, V., C. Murat, and A. Thiele (2013). "Recent advances in robust optimization: an overview". *European Journal of Operational Research*. Available online.
- Galvão, R., F. Chiyoshi, and R. Morabito (2005). "Towards unified formulations and extensions of two classical probabilistic location models". Computers & Operations Research 32 (1), pp. 15– 33.
- Gendreau, M., G. Laporte, and F. Semet (1997). "Solving an ambulance location model by tabu search". Location Science 5 (2), pp. 75–88.
- Gendreau, M., G. Laporte, and F. Semet (2001). "A dynamic model and parallel tabu search heuristic for real-time ambulance relocation". *Parallel Computing* 27 (12), pp. 1641–1653.
- Gendreau, M., G. Laporte, and F. Semet (2006). "The maximal expected coverage relocation problem for emergency vehicles". Journal of the Operational Research Society 57 (1), pp. 22– 28.
- Geroliminis, N., M. Karlaftis, and A. Skabardonis (2006). "A generalized hypercube queueing model for locating emergency response vehicles in urban transportation networks". Transportation Research Board 2006 Annual Meeting.
- Geroliminis, N., M. Karlaftis, A. Stathopoulos, and K. Kepaptsoglou (2004). "A districting and location model using spatial queues". *Transportation Research Board 2004 Annual Meeting*.

Glover, F. and M. Laguna (1997). Tabu search. Kluwer Academic Publishers.

- Goemans, M., N. Harvey, S. Iwata, and V. Mirrokni (2009). "Approximating submodular functions everywhere". Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms.
- Goldberg, J. (2004). "Operations research models for the deployment of emergency services vehicles". EMS Management Journal 1 (1), pp. 20–39.
- Goldberg, J., R. Dietrich, J. Chen, M. Mitwasi, T. Valenzuela, and E. Criss (1990). "Validating and applying a model for locating emergency medical vehicles in Tuczon, AZ". European Journal of Operational Research 49 (3), pp. 308–324.
- Goldberg, J. and L. Paz (1991). "Locating emergency vehicle bases when service time depends on call location". *Transportation Science* 25 (4), pp. 264–280.
- Goundan, P. and A. Schulz (2007). "Revisiting the greedy approach to submodular set function maximization". Working paper.
- Grötschel, M., L. Lovász, and A. Schrijver (1981). "The ellipsoid method and its consequences in combinatorial optimization". *Combinatorica* 1 (2), pp. 169–197.
- Gülpınar, N., D. Pachamanova, and E. Çanakoğlu (2013). "Robust strategies for facility location under uncertainty". European Journal of Operational Research 225 (1), pp. 21–35.
- Gurobi Optimization (2014). Gurobi optimizer reference manual. URL: http://www.gurobi.com.
- Hochbaum, D. and A. Pathria (1998). "Analysis of the greedy approach in problems of maximum k-coverage". Naval Research Logistics 45 (6), pp. 615–627.
- Hogan, K. and C. ReVelle (1986). "Concepts and applications of backup coverage". Management Science 32 (11), pp. 1434–1444.
- Iannoni, A. and R. Morabito (2007). "A multiple dispatch and partial backup hypercube queuing model to analyze emergency medical systems on highways". Transportation Research Part E: Logistics and Transportation Review 43 (6), pp. 755–771.
- Ingolfsson, A., S. Budge, and E. Erkut (2008). "Optimal ambulance location with random delays and travel times". *Health Care Management Science* 11 (3), pp. 262–274.
- Jain, K. and V. Vazirani (2001). "Approximation algorithms for metric facility location and kmedian problems using the primal-dual schema and Lagrangian relaxation". Journal of the ACM 48 (2), pp. 274–296.
- Jarvis, J. (1985). "Approximating the equilibrium behavior of multi-server loss systems". Management Science 31 (2), pp. 235–239.
- Karasakal, O. and E. Karasakal (2004). "A maximal covering location model in the presence of partial coverage". Computers & Operations Research 31 (9), pp. 1515–1526.
- Karp, R. (1972). "Reducibility among combinatorial problems". Complexity of Computer Computations. Ed. by R. Miller, J. Thatcher, and J. Bohlinger. The IBM Research Symposia Series. Springer US, pp. 85–103.
- Kommer, G., B. Zuzáková, P. Van den Berg, and S. Bhulai (2012). "Optimal ambulance capacities under uncertain demand and response times". Working paper.

- Kommer, G. and S. Zwakhals (2011). Modellen referentiekader ambulancezorg 2008. Documentatie rijtijden- en capaciteitsmodel. Rapport 270412001/2011. Rijksinstituut voor Volksgezondheid en Milieu (RIVM).
- Larson, R. (1974). "A hypercube queuing model for facility location and re-sub-areaing in urban emergency services". *Computers & Operations Research* 1 (1), pp. 67–95.
- Larson, R. (1975). "Approximating the performance of urban emergency service systems". Operations Research 23 (5), pp. 845–868.
- Lee, J., V. Mirrokni, V. Nagarajan, and M. Sviridenko (2009). "Non-monotone submodular maximization under matroid and knapsack constraints". Proceedings of the 41st Annual ACM Symposium on Theory of Computing, pp. 323–332.
- Lee, L., T. Lau, and Y. Ho (1999). "Explanation of goal softening in ordinal optimization". IEEE Transactions on Automatic Control 44 (1), pp. 94–99.
- Li, X., Z. Zhao, X. Hu, and T. Wyatt (2011). "Covering models and optimization techniques for emergency response facility location and planning: a review". *Mathematical Methods of Operations Research* 74 (3), pp. 281–310.
- Liebchen, C., M. Lübbecke, R. Möhring, and S. Stiller (2009). "The concept of recoverable robustness, linear programming recovery, and railway applications". *Robust and Online Large-Scale Optimization*. Ed. by R. Ahuja, R. Möhring, and C. Zaroliagis. Vol. 5868. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 1–27.
- Looije, D. (2013). "Optimal ambulance locations in the Netherlands". Master's thesis. Delft University of Technology.
- Mandell, M. (1998). "Covering models for two-tiered emergency medical services systems". Location Science 6 (1), pp. 355–368.
- Marianov, V. and C. ReVelle (1994). "The queuing probabilistic location set covering problem and some extensions". *Socio-Economic Planning Sciences* 28 (3), pp. 167–178.
- Marianov, V. and C. ReVelle (1996). "The queueing maximal availability location problem: a model for the siting of emergency vehicles". European Journal of Operational Research 93 (1), pp. 110–120.
- Marianov, V. and D. Serra (2001). "Hierarchical location-allocation models for congested systems". European Journal of Operational Research 135 (1), pp. 195–208.
- McLay, L. (2009). "A maximum expected covering location model with two types of servers". *IIE Transactions* 41 (8), pp. 730–741.
- Nemhauser, G. and L. Wolsey (1978). "Best algorithms for approximating the maximum of a submodular set function". Mathematics of Operations Research 3 (3), pp. 177–188.
- Nemhauser, G., L. Wolsey, and M. Fisher (1978). "An analysis of approximations for maximizing submodular set functions-I". *Mathematical Programming* 14 (1), pp. 265–294.
- Noyan, N. (2010). "Alternate risk measures for emergency medical service system design". Annals of Operations Research 181 (1), pp. 559–589.
- Papadimitriou, C. and K. Steiglitz (1998). Combinatorial optimization : algorithms and complexity. Dover Publications, Inc.
- Rajagopalan, H., C. Saydam, and J. Xiao (2008). "A multiperiod set covering location model for dynamic redeployment of ambulances". Computers & Operations Research 35 (3), pp. 814–826.
- Repede, J. and J. Bernardo (1994). "Developing and validating a decision support system for locating emergency medical vehicles in Louisville, Kentucky". European Journal of Operational Research 75 (3), pp. 567–581.
- Restrepo, M. (2008). "Computational methods for static allocation and real-time redeployment of ambulances". PhD thesis. Cornell University.
- ReVelle, C. (1989). "Review, extension and prediction in emergency service siting models". European Journal of Operational Research 40 (1), pp. 58–69.

- ReVelle, C. and K. Hogan (1989a). "The maximum availability location problem". Transportation Science 23 (3), pp. 192–200.
- ReVelle, C. and K. Hogan (1989b). "The maximum reliability location problem and alpha-reliable p-center problems: derivatives of the probabilistic location set covering problem". Annals of Operations Research 18 (1), pp. 155–174.
- Schilling, D., D. Elzinga, J. Cohen, R. Church, and C. ReVelle (1979). "The Team/Fleet models for simultaneous facility and equipment siting". *Transportation Science* 13 (2), pp. 163–175.
- Schmid, V. and K. Doerner (2010). "Ambulance location and relocation problems with timedependent travel times". European Journal of Operational Research 207 (3), pp. 1293–1303.
- Schrijver, A. (2000). "A combinatorial algorithm minimizing submodular functions in strongly polynomial time". Journal of Combinatorial Theory 80 (2), pp. 346–355.
- Shen, Z., Q.-C. Zhao, and Q.-S. Jia (2010). "Quantifying heuristics in the ordinal optimization framework". Discrete Event Dynamic Systems 20 (4), pp. 441–471.
- Silva, F. and D. Serra (2008). "Locating emergency services with different priorities: the priority queuing covering location problem". Journal of the Operational Research Society 59 (9), pp. 1229–1238.
- Snyder, L. (2006). "Facility location under uncertainty: a review". IIE Transactions 38 (7), pp. 537–554.
- Sorensen, P. and R. Church (2010). "Integrating expected coverage and local reliability for emergency medical services location problems". Socio-Economic Planning Sciences 44 (1), pp. 8– 18.
- Storbeck, J. (1982). "Slack, natural slack, and location covering". Socio-Economic Planning Sciences 16 (3), pp. 99–105.
- Toregas, C., R. Swain, C. ReVelle, and L. Bergman (1971). "The location of emergency service facilities". Operations Research 19 (6), pp. 1363–1373.
- Van Buuren, M. (2010). "TIFAR modeling package for the evaluation of emergency medical services. With EMS modeling results for the Amsterdam area". Master's thesis. Delft University of Technology.
- Van den Berg, P. and K. Aardal (2013). "Time-dependent MEXCLP with start-up and relocation cost". Working paper.
- Vohra, R. and N. Hall (1993). "A probabilistic analysis of the maximal covering location problem". Discrete Applied Mathematics 43 (2), pp. 175–183.
- Vondrák, J. (2007). "Submodularity in combinatorial optimization". PhD thesis. Charles University.
- Vondrák, J. (2009). "Symmetry and approximability of submodular maximization problems". Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science.
- Voß, S. (2001). "Meta-heuristics: the state of the art". Local Search for Planning and Scheduling. Ed. by A. Nareyek. Vol. 2148. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 1–23.
- Voß, S., I. Osman, and C. Roucairol, eds. (1999). Meta-heuristics: advances and trends in local search paradigms for optimization. Kluwer Academic Publishers.
- Williamson, D. and D. Shmoys (2011). The design of approximation algorithms. Cambridge University Press.
- Zhang, Z. and H. Jiang (2013). "A robust counterpart approach to the bi-objective emergency medical service design problem". *Applied Mathematical Modelling*. Accepted for publication.