UiT Norges arktiske universitet

Faculty of Engineering Science and Technology

Master Thesis: The Report

Optimal Positioning for Transceivers in Network

Using optimization techniques and viewshed analysis to optimize the location of short-range transceivers, inspired by underwater wireless communication characteristics.

Jarand Rage Aasvold

jarandaasvold@gmail.com

Department of Technology and Safety

22.05.2024

Abstract

This thesis investigates the use of optimization techniques to configure a network of short-range transceivers for best coverage of a topographic layer. Three methods are presented, tested, and analyzed.

The thesis is inspired by the limited range of underwater wireless communication transceivers. Therefore, a 100-meter radius range and omnidirectionality are selected as transceiver characteristics. The coverage is defined by free "Line of Sight" (LOS), and a point is therefore considered covered if within the 100 meter radius and in LOS. The only interference considered is topographical interference. The terrain is represented by a Digital Elevation Model, and coverage is computed using viewshed analyses. The data used are from a terrestrial area due to military restrictions on subsea data.

Method 1 solves the problem as an Integer Linear Program (ILP). Demand Points (points to be covered) and Candidate Points (points where the transceivers can be positioned) are randomly selected. Further, it is analyzed which demand points that are covered by the respective candidate points. This information is stored in a Coverage Matrix (CM) and solved as an ILP. Method 2 uses a local optimization algorithm to successively deploy the transceiver at the location where it increases the coverage of the Demand Zone (DZ) the most, given what is already covered. When the optimization algorithm finds an optimum, a transceiver is deployed, and the algorithm starts a new iteration. Method 2 is developed in this project and does not consider the task as one big combinatorial problem, but rather a sequence of sub-problems. Method 3 initially performs Method 2, but saves viewshed information computed in its search for optimum. When Method 2 is completed, this viewshed-information is structured as a CM and solved using ILP.

By configuring a network of 35 transceivers in the test area of 1 100 000m^2, it is found that Method 3 produces the best coverage, with Method 1 just behind. Method 2 produces the network configuration fastest, with $\sim$ 35% reduction in process time compared to Method 1, at the cost of only 2.3 percentage points less coverage. On larger problems it is found that Method 2 and 3 increase linearly, while Method 1 increase non-linearly following a quadratic function. Arguably Method 2 performance is appealing given its process time and coverage, however, Method 1 and 3 are expected to perform better where interwoven coverage is important.

Preface

This thesis completes my time as a student in the two-year master program "Technology and Safety in the High North" at UiT. The thesis work has been conducted between Janauary and May 2024.

I would like to thank my outstanding supervisors Anne Mai Ersdal and Jarle André Johansen for lending their expertise, support, guidance, and feedback to my project. I would also like to extend my gratitude for their patience through their many, many office consultations. I would also express my sincere appreciation to my good friend Nate for being interested in the thesis and giving me thorough feedback whenever I need it.

Finally, I want to thank my partner Vårinn, for always being by my side and supporting me. As I complete this thesis, Vårinn is 5 months pregnant, and we are expecting our first son in September. Exiting times ahead!

Jarand Rage Aasvold

Tromsø, 22.05.2024

Contents

List of Figures

List of Tables

1 Introduction

Location problems have always been important for humans, from basics such as settling near access to water and arable land, communication such as access to ports or other trade networks, or military strategic locations such as viewpoints. Location problems have not become easier, and in the modern era, we exploit the computational powers of computers to solve many of these problems. Around 1970, three fundamentally different location problems were mathematically defined. The Location Set Coverage Problem (LSCP – full coverage as cheap as possible), Maximal Coverage Location Problem (MCLP – maximal coverage within budget), and Minimum Impact Location Problem (MILP – Avoid coverage, opposite of MCLP) [1].

A generic example in location science is where should the bus stops be placed in order to service the population to be within maximum walking distance from their home to the bus stop. In such "simple" problems, terrain is often not considered, and the problem is solved on a 2D plane. Location problems where topography is a consideration advance the problem. Such problems may involve the placement and height of a new antenna. One way of integrating topographical information into the problem is to perform viewshed analyses on a Digital Elevation Model (DEM).

This project investigates various optimization methods to configure the individual locations of transceivers in a network. The project is inspired by the difficulties of Underwater Wireless Communication (UWC). Due to the high permittivity of water (and especially salt water), electromagnetic waves are not a viable option for longer distances. Traditionally, underwater acoustics has been the leading option for UWC. Although acoustic waves propagate at circa 1500 m/s, almost five times as fast as in air, this is still a fraction of the speed of electromagnetic waves in water. This property makes fast and large data transfers difficult. In recent years, non-conventional alternatives have been developed for medium-distance communications (up to a few hundred meters). Communication with free-space optical modems, utilizing the blue-green transmittance window, offers high-speed data transfers, with a satisfying bandwidth around 10 Mbps. Another alternative is UWC with magnetic induction (MI). MI is statically only affected by permeability, and water's permeability is the same as air's permeability. The principle of MI communication is modulating a varying magnetic field, inducing current in the receiver coil as the magnetic field changes [2]. In Bahr et al. [3], optical

modem networks are deployed to create high-speed areas where underwater robotics can transfer data (communicate) in a high-speed network, much like WiFi. This project will explore how optimal location of transceivers can create coverage networks, working with the severe range limitation of wireless subsea transceivers.

In the literature, problems with comparable characteristics to our problem are found. For example, in Bao et al. [4], a combination of location coverage problems and viewshed analysis is used to optimize the location of watchtowers for forest fire monitoring. By selecting candidate points (CP) and demand points (DP), they solved the problem optimally using Integer Linear Programming (ILP), both for MCLP and LSCP. In the Austrian Alps, Welscher et al. [5] strove to reduce the cost of an expensive animal monitoring radio communication system, by stationing the transceivers at the optimal location. This study also used viewshed evaluations as tool for coverage calculation and solved the problem as ILP. This thesis does not work under the characteristics of any

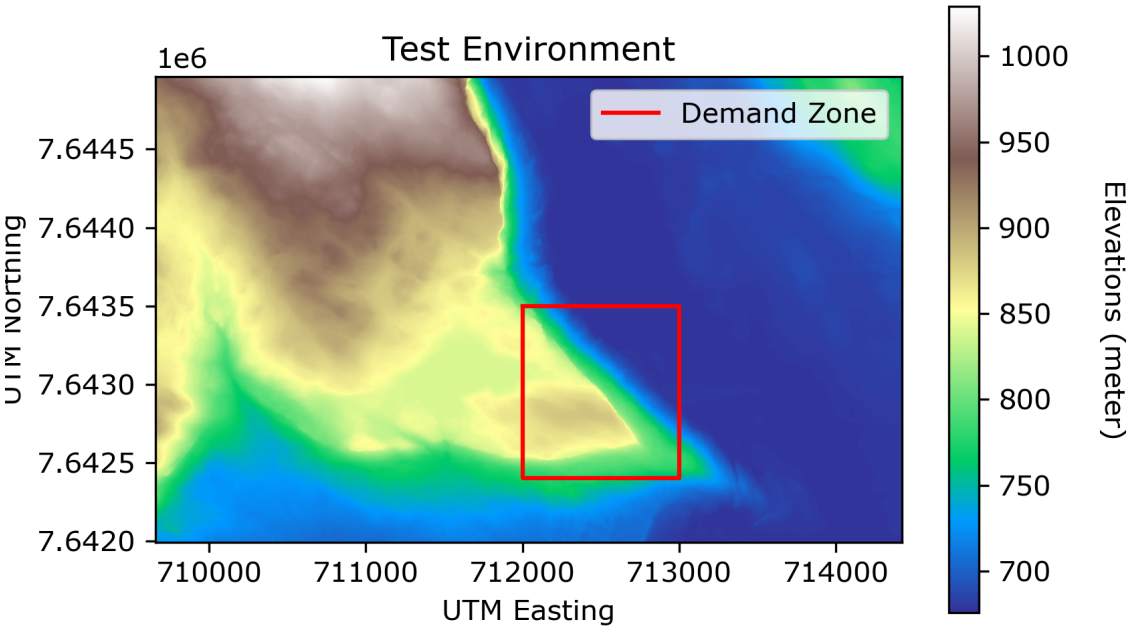

Figure 1: Plotted test area. The red box is the "demand zone" (the zone to be covered)

specific product on the market, but rather makes the assumption of omnidirectionality and a range of 100 meters, inspired by current technology. Viewshed is the parameter for line of sight, and no other sources of interference than topographical interference are

considered.

Three different methods that all produce a network configuration will be presented, analyzed, and compared. The methods configures the network to cover as much of the demand zone as possible (see Figure 1). The first method is based on a method from the literature, using ILP on a coverage matrix (CM). The coverage matrix is made by checking whether individual DPs are seen by individual CPs. The second method utilizes the assumption that neighboring points have similar viewsheds, so by treating the data as values of a function the location with the highest viewshed can be found by using optimization such as Nelder-Mead. This method will produce a network configuration by successively deploying transceivers at the location that increases the coverage in the demand zone the most, until the maximum number of transceivers has been deployed. The third method is a combination of the two first methods. The idea is to store data from the viewshed analysis used in Method 2 (successive deployment using Nelder-Mead), and use these data to create a coverage matrix that is solved as ILP. This is being done to possibly improve the successive deployment result, without having to run more viewshed analyses.

1.1 Objective

The objective of this study is the implementation of optimization algorithms to configure short-range transceiver networks on the terrain. The coverage of a given area will be optimized by positioning a given number of transceivers. The objective is not to cover the total area but as much as possible with the given number of transceivers (as MCLP). The only interference considered is topographical interference and range. Coverage on the DEM surface is the only coverage considered. However, this is arguably the most challenging cover to achieve. Coverage above the DEM will also be present in these networks, but only coverage projected on the DEM surface is considered. The main parameters in this scenario are set to simulate an underwater environment:

- The transceiver is omnidirectional with a range of 100 meters (radius)
- The transceiver is placed 2 meters above the seabed (The DEM)
- Viewshed-analysis is used to calculate coverage.

The work of this project culminates in these methods:

- Integration of ILP with random DP and CP (inspired from literature).
- Development of Successive Deployment technique.
- Combine the data from the Successive Deployment with ILP to improve the result.

1.2 Environment

The subsea environment is hostile to electromagnetic waves due to the strong attenuation caused by the permittivity of seawater. Underwater optical communication and magnetic induction communication are two options that offer a similar transmission speed, but in a very limited range [2]. However, the engineers of Hydromea explain that by using their optical modem in a concentrated network, they create "high speed communication bubbles" [3]. In underwater robotics, wireless communication is a bottleneck. Therefore, today's underwater robotics are mostly tethered. The tether connects the drone to power and communication, but restricts the maneuverability and range of the vehicle. Autonomous Underwater Vehicles (AUV) are constantly being developed, and are already performing pre-programmed tasks based on onboard intelligence. These drones

are detached and usually do not have the possibility for immediate access by an operator. [6].

This thesis' aim is to compare methods that use optimization techniques to produces locations for a set of transceivers to form a subsea wireless network. The idea is to deploy transceivers in an "area of interest", where intensified underwater drone use is present. Examples of this are areas where a survey or surveillance is being performed, the area around a subsea energy installation, etc. Deploying a wireless network opens the possibility for tetherless Remotely Operate Vehicle (ROV) operations and makes it easier for autonomous drones to deliver and receive data, making the man-machine interface easier and more accessible.

In section 1.1 it is explained that only the coverage on the DEM surface is considered. The drone itself will not "drive" at the seabed but operate in the waters just above. When moving away from the seabed, the topographical interference decreases. Therefore, to cover the seabed is believed to be the hardest to achieve. All though this project only consideres the seabed, the transceivers will naturally create a network also in the area above the seabed. The wireless network will be static, making it possible to use a coverage map. This allows the drone and the operators on the topside to plan for exits and entries, given their awareness of the coverage properties. As explained in section 1, the coverage will be based on two properties: Free line of sight, estimated with viewshed analysis, and signal range of 100 meters. The transceiver is set to be placed two meters above the seabed. The transceiver is connected by tether for power and communication. A transciever could be either on a string with a weight at the seabed and buoy, or it could be mounted on a stand or structure.

The data used in this project represent an inland area. This does not change the problem's datatype, as the topographical structures/layer is the consideration and not the altitude above/below sea level. Subsea data are not used due to military restrictions on such data (deeper than 30 meters or better resolution than 50x50 meter is classified). Both high-resolution marine and terrestrial surface data exist, however, only terrestrial data is open for public use [7].

2 Theory

In this section, the theory used in the project will be presented. First is a brief introduction to location science, followed by numerical optimization and the algorithms used in this project. In Section 2.3 there is information about the DEM format, reference system, and topography. Finally, in section 2.4 selected literature relevant for this project is reviewed.

2.1 Location Science

Location Science is a field of study that is based on a range of disciplines including mathematics, geography, logistics, economics and engineering. It emerges from the need to theoretically and technically solve practical location problems. Location problems have likely been a consideration for humans from the days of hunter-gatherers, although in simple forms like where to establish camps to assure basic needs like protection, access to water and food, and how easy it is to clear for use. Today location science is a sophisticated and mature field of study, using advanced theoretical models and tools like Geographic Information Systems (GIS) software to solve location problems such as bus scheduling, antenna network configurations, and emergency preparedness. The theory presented in this section is based on [1] and [8]

Facility Location problems are a fundamental topic in Location Science [8]. These problems optimize the location of one or more facilities/resources for the best possible coverage of a set of points. The term “coverage” can be defined by a quantification of a service that the facilities provide. The problems themselves are often trivial to understand and construct; however, the complexity of the problems make them difficult to compute. Church and Murray [1] strongly emphasize that "this fact alone hindered the development of this field until the invention of the modern computer" and further divide the time before and after the advent of the computer into two different eras in location science [1].

Church and Murray’s “Location Covering Models” [1] define three principal forms of Location Covering Models:

- The Location Set Coverage Problem (LSCP). This problem aims to cover the total defined area with the fewest number of facilities.

- The Minimum Impact Location Problem (MILP). This problem seeks to deploy a given number of facilities, while minimizing coverage (military facilities, nuclear power plants, etc.).

- The Maximal Coverage Location Problem (MCLP), which optimizes the problem on a budget. This means that the total area may not be covered, but the solution should cover as much as possible while remaining within budget.

The final model, MCLP, lays the framework for the problem in this thesis. It is originally defined as: "Maximize coverage (population covered) within a desired service distance S by locating a fixed number of facilities" [9]. In section 2.4, selected literature is reviewed, providing more information about location science, MCLP and more.

2.2 Numerical Optimization

Numerical optimization is an analytical tool to optimize variables to an objective. The theory presented in this section is based on [10]. In mathematical terms, optimization is the best value from the objective function. The objective function contains decision variables (variables that can be modified to achieve the best value). Decision variables may be subject to constraints. These are divided into two groups, equality constraints and inequality constraints. Inequality constraints are naturally denoted as inequalities and may represent that a variable needs to be larger than a certain value, for instance, greater than zero (positive). Equality constraints are more strict, for instance restricting a variable (or the sum of two variables) to a certain value. Although optimization problems can be both maximization and minimization, the standard formulation of optimization problems formulates the problem as a minimization problem. However, a simple minus before the objective function $(-f)$ will alter this.

A standard formulation of an optimization problem is as follows.

$$\min_{x \in R^n} f(x), \tag{1}$$

subject to:

$$c_i(x) = 0, i \in \mathcal{E}, \tag{2}$$

$$c_i(x) \geq 0, i \in \mathcal{I}, \tag{3}$$

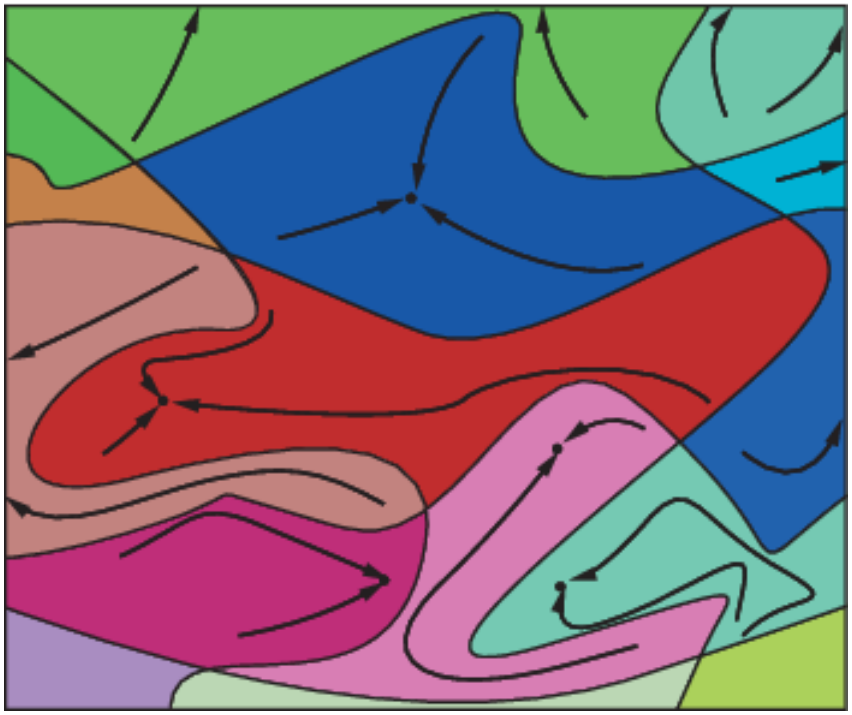

Figure 2: An illustration of how one might visualize the Basins of attraction. Figure from [11].

where $f(x)$ if the objective functions to be minimized, and x is the decision variables. $c_i(x)$ is the constraints the objective function is subject to, divided into two categories $\mathcal{I}$ (inequalities) and $\mathcal{E}$ (equalities).

There is no universal method for solving optimization problems. Instead, a collection of algorithms exists to solve them. The characteristics of the problem must be considered to find a suitable algorithm. For instance, questions to evaluate could include: Is the problem linear? Does the algorithm need derivatives, and if so is it easy to provide them? Does the problem have constraints, and does the algorithm handle constraints? Generally, linear optimization problems are easier to solve than non-linear ones. Linear optimization, often called linear programming, is a special case of optimization, where the objective function and the constraints are linear functions of x. These properties makes the problem convex, indicating that local solution are also global solution. Dantzig's Simplex Method is one of the most-used optimization algorithms, providing an efficient, systematic strategy to optimize the special case of linear programs. In the case of non-linear problems, the "local is global" feature is not present. The global solution is the optimal solution among all feasible points (points that satisfies the constraints), whereas the local solution might "just" be the best solution among the neighboring points. An

analogy is that of peaks in a mountain range. Every peak (the highest point of a mountain) is a local solution; however, only the highest point in the range will be global.

Many nonlinear optimization algorithms are local solvers, which means that they converge toward a local optimum. The resulting local optimum will often be related to the initial value given the solver. The region around the local optimum that holds the set of initial points that will lead to the local optimum is called the "basin of attraction" (see figure 2). Linear problems only have one basin of attraction, and therefore the "global is local" feature. Nonlinear problems may have several such basins, as illustrated in figure 2, and they are therefore sensitive on the initial value.

In general, optimization algorithms are iterative processes, where each step produces a suggestion for the optimization variable x. In many algorithms, this value should achieve a certain reduction in the objective function. A common strategy within optimization is to use first and second order derivatives to analyze gradient and curvature properties of the objective function. Note that the step-wise reduction in objective value is also the reason these algorithms find local optimums. The algorithm stops when a termination criterion has been met. Examples of termination criteria are: maximum number of iterations, fulfilled convergence test, and maximum process time.

The derivatives may not be directly available for the algorithm. In that case, approximation of derivatives is an option; however, this may be computationally demanding. There exist a number of derivative-free optimization (DFO) algorithms using other principles/properties than gradient and curvature. Nelder-Mead is one of them and will be explained in 2.2.3.

2.2.1 Integer Linear Programming

Integer Linear Programming is a special case in Linear Programming, where some (Mixed Integer Linear Programming) or all (Full Integer Linear Programming) decision variables of the linear problem must be integers [12]. Integer Linear Programming allows us to formulate problems where the decision variables are discrete and a standard formulation of ILP is on the following form:

$$\min_{z} \quad J(z) = c^T z \tag{4}$$

$$\text{subject to} \quad Az = b \tag{5}$$

$$l \leq z \leq u, \tag{6}$$

where some or all elements in z are restricted to be integers. J is the objective function, c holds the cost associated with the decision variables in z. Equation 5 states the equality constraints where some matrix A multiplied by z must equal vector b. The final equation states the inequality constrains, that z must be between the lower l and upper u vector. This ILP form has many real word applications. In this thesis, the integral restriction becomes very clear. Either an antenna is placed in the position ($x = 1$), or it is not ($x = 0$). Either a point is covered ($y = 1$) or it is not ($y = 0$), where x and y are elements of the vector z.

2.2.2 Branch-and-Bound

There exists a collective of algorithms that solves ILP problems. Many of them involve LP relaxation, meaning temporarily removing the integer constraint, then solving the problem as an LP, and finally compromise the solution to fit the integer constraints again. This is what the the Branch-and-Bound (B&B) algorithm does. The theory presented in this section is based on [13] and [14].

The B&B algorithm initially removes the integer constraints. This relaxation transforms the problem to a trivial LP-problem that can be solved as such. If the relaxed LP-problem solution happens to meet the relaxed integer constraints, the ILP solution is found. However, this is rarely the case, and it is likely that some of the decision variables are non-integers. The B&B algorithm then creates two sub problems to avoid the infeasible solution found by relaxation. This process is called the *branch* step. As an example, say that the solver found the optimum in $x_1 = 3.3$ and that x_1 is bounded by the integer constraint. This initial problem is named P_0. The branching divides the problem into two alternative subproblems to avoid the infeasible solution $x_1 = 3.3$. The alternative a is to pose that $x_1 \leq 3$ or alternative b is $x_1 \geq 4$. These two new sub-LP problems P_1a and P_1b now replace the original problem. Together, they form a more restricted problem and narrow down the search for the optimal solution to the original problem.

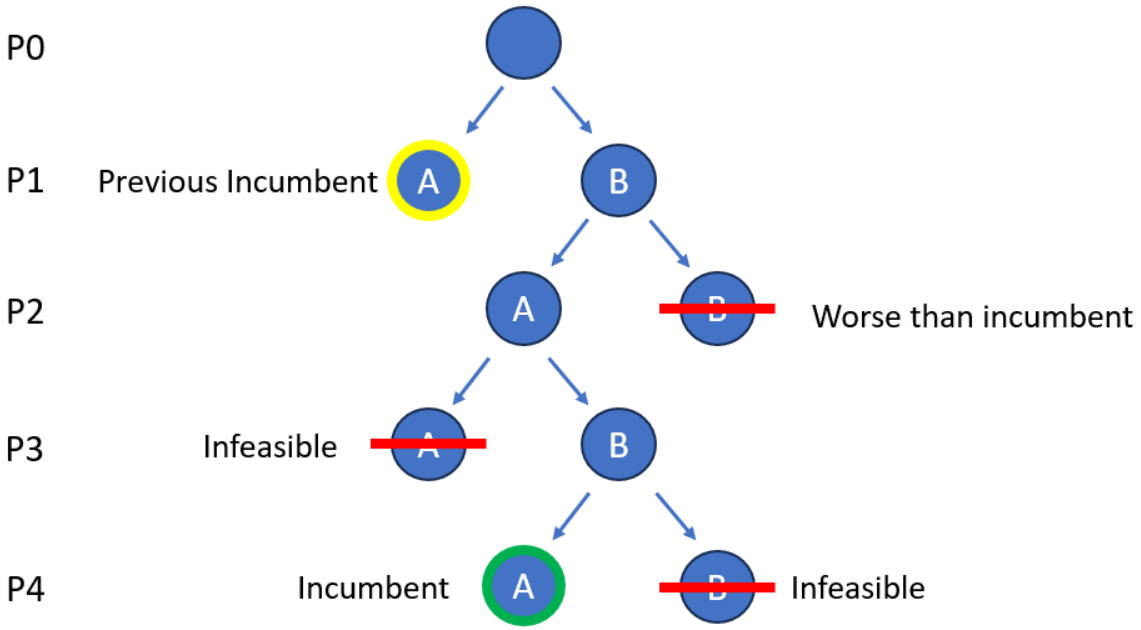

Figure 3: Branch-and-Bound tree, or search tree. The red lines symbolizes cut branches. The yellow outline symbolizes an initial solution to the problem that has now been replaced by a better solution. The green outline signals the current "incumbent". The figure is based on the example from [14].

.

The next step is to relax the new problems and solve them as LP problems. Now examine the results: If the solution meets the relaxed integral constraints and is the best "found to date" ("incumbent"), it replaces the incumbent. However, if the relaxed solution is worse than the solution stored as the incumbent (if any) or the the problem is infeasible, the branch is "cut", meaning the algorithm does not continue the search on this branch. Now if the problem does not meet the relaxed integral constraints, it branches into more subproblems. Figure 3 is an illustration of how a B&B search can evolve. This structure is called a search tree.

#One step of Branch and Bound (Adapted from B.C. Williams [14])

I: Bound Pn's solution and compare to alternatives:

1) Bound solution to Pn quickly:

- Relax integer constraints and solve as LP.

2) Use bound to "fathom" Pn if possible:

a. If relaxed solution is integer:

- Keep solution if best found to date ("incumbent").
- Cut P_n.

b. If relaxed solution is worse than incumbent:

- Cut P_n.

c. If no feasible solution:

- Cut P_n.

II: Otherwise Branch to smaller subproblems:

1) Partition P_n into subproblems $P[n+1]A, P[n+1]B, \ldots$

2) Apply B&B to all subproblems.

2.2.3 Nelder-Mead

The Nelder-Mead simplex reflection method is a gradient-free and DFO algorithm. It has been popular since being introduced in 1965. The idea of Nelder-Mead is not to follow the steepest slope to optimum, but rather to find an area of improvement and search there. The theory presented in this section is based on Nocedal and Wright [10].

The Nelder-Mead works in multidimensions. The algorithm iterations consists of manipulation of a vertex in a simplex. This simplex is a hypertetrahedron with n+1 vertices where n represents the number of dimensions in $\mathbb{R}^n$. For a two-dimensional problem, the simplex would be a triangle. For a three-dimensional problem, the simplex would be a 'pyramid' etc.

For every iteration, the Nelder-Mead method selects the vertex with the worst value (objective value) and seeks to replace this vertex with a significantly better one. This can be performed with 3 principle maneuvers: Reflection, Expansion or Contraction. If none of these maneuvers results in sufficient improvement, a fourth option is performed. From the best valued vertex, the simplex shrinks, "pulling" all the other vertices towards the best valued vertex. The three principle maneuvers are performed over the centroid of the remaining best points (all the points but the worst). The centroid is denoted by:

$$\tilde{x} = \sum_{i=1}^{n} x_i, \tag{7}$$

where x represent the function value of the respective vertices. On a triangle, the centroid is therefore located exactly between the two best points (illustrated as a dot on the line between x_1 and x_2 in figure 4). The points deducted from three first principle manoeuvres is denoted by:

$$\tilde{x}(t) = \tilde{x} + t(x_{n+1} - \tilde{x}) \tag{8}$$

The procedure for one iteration is as follows: Calculate the function value of the vertices and identify from best x_1 to worst x_{n+1}. Having identified the worst vertex, compute the reflected point $\tilde{x}(-1)$. From now on the original simplex is referred to as the "old simplex", while the simplex that contains the new point (instead of the worst point in the old simplex) is referred to as the "new simplex". At this point, the value of the reflected point fits in one of these three main scenarios, with alternate procedures

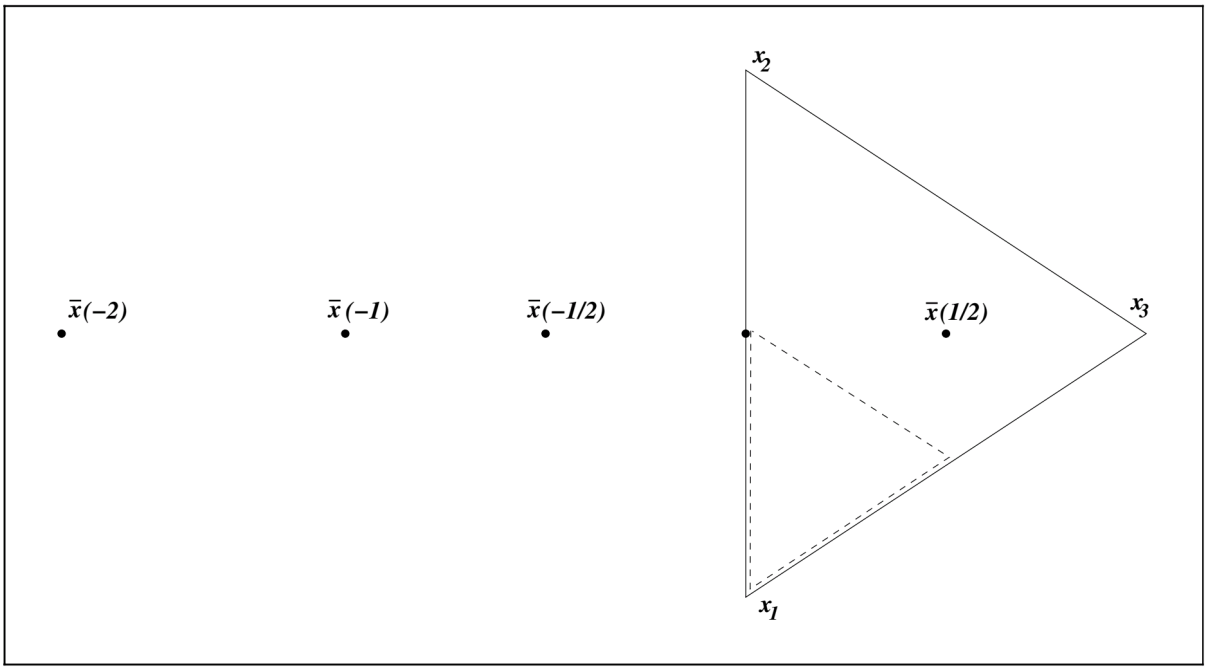

Figure 4: Nelder-Mead principle maneuver points. From left: Expansion, reflection, outside contraction and inside contraction. Note that this is in $\mathbb{R}^2$ and the number of vertices is therefore 3. They are ordered by name, where x_1 being the best, and x_3 (generally x_{n+1}) being the worst. The centroid is on the line between x_1 and x_2. Figure from [10]

.

depending on the scenario:

1. The reflected point is somewhere between the best and worst vertex in the new simplex. We replace the worst vertex in the old simplex $x_n + 1$, with the new $\tilde{x}(-1)$, and go to the next iteration.

2. The reflected point is the best point in the new simplex. We then try to go even further in this direction. We compute the expansion point $\tilde{x}(-2)$, and investigate:

   (a) The expansion point is better than the reflection point ($f_{-2} < f_{-1}$), we replace $x_n + 1$, with the new $\tilde{x}(-2)$, and go to the next iteration.

   (b) The expansion point is worse than the reflection point ($f_{-2} > f_{-1}$), we replace $x_n + 1$, with the new $\tilde{x}(-1)$, and go to the next iteration.

3. The reflected point is worse than x_n (the second worst in the old simplex, and the worst in the new simplex excluding the new point). We compute the outside

contraction point and investigate:

(a) The outside contraction point is better than the reflection point ($f_{-1/2} \geq f_{-1}$), we replace $x_n + 1$, with the new $\tilde{x}(-1/2)$, and go to the next iteration.

(b) The outside contraction point is worse than the reflection point, we compute the inside contraction point and investigate:

i. The inside contraction point is better than the worst original point ($f_{1/2} \geq f_{n+1}$), we replace $x_n + 1$, with the new $\tilde{x}(-1/2)$, and go to the next iteration.

ii. Neither the inner nor outer contraction fulfill the criteria. We perform the shrink maneuver. The shrink maneuver is performed by replacing all x_i with $(1/2)(x_1 + x_i)$ for $i = 2, 3, \ldots, n + 1$ (replacing all except the best vertex x_1). Then go to the next iteration.

The algorithm is guaranteed to decrease the average function value (average objective value among all vertex in the simplex) for every iteration unless the shrink maneuver is performed. The Nelder-Mead Simplex method is one of the best known algorithms for derivate free, unconstrained optimization [15]. However, constraints and bounds can be applied with a so-called penalty function. The function causes severe negative impact on the objective value whenever the constraint is violated. This property can cause unwanted use of the shrink maneuver and stopping, when iterative steps along the constraint are still "needed." Nevertheless, this is usually not a problem when there are few parameters involved in the optimization process [16].

#One step of Nelder-Mead Simplex (From Nocedal and Wright [10])

Compute reflection point $\tilde{x}(-1)$;
if $f(x_1) \leq f_{-1} \leq f(x_n)$:
    replace x_{n+1} by $\tilde{x}(-1)$ and go to next iteration;
else if $f_{-1} < f(x_1)$:
    Compute the expansion point $\tilde{x}(-2)$;
    if $f_{-2} < f_{-1}$:
        replace x_{n+1} by x_{-2} and go to next iteration;
    else:
        replace x_{n+1} by x_{-1} and fo to next iteration;
else if $f_{-1} \geq f(x_n)$:
    if $f_{x_n} \leq f(x_{-1} \leq f(x_n + 1)$:
        Compute outside contraction;
        if $f_{-1/2} \leq f_{-1}$:
            replace x_{n+1} by $x_{-1/2}$ and go to next iteration;
    else:
        Compute inside contraction;
        if $f_{1/2} < f_{n+1}$:
            replace x_{n+1} by $x_{1/2}$ and go to next iteration;
        else:
            When neither of the contractions fulfill
            the criterion, we perform the shrink
            maneuver:
            replace $x_i \to (1/2)(x_1 + x_i)$ for $i = 2, 3, ..., n + 1$
            and go to next iteration;

2.3 Topographical data and GeoTIFF

Elevation data was gathered from hoydedata.no, a web page owned by the Norwegian Mapping Authority (NMA) [17]. A study area was selected in Rostajávri and the surrounding area. The data were exported as a Digital Elevation Model (DEM) at a resolution of 1 meter. This is a "GeoTIFF" raster file, where each pixel (cell) represents an area of 1x1 meter. The elevation data are stored in a raster band (an information layer), where values correspond to each pixel. A raster file can have multiple bands, where each band can store different information. In this project, we are using GDAL as a plugin in the Python script. GDAL is (among other features) capable of reading, writing, and adding bands in the GeoTIFF file [18]. The GeoTIFF file is an extension of the well-used TIFF (Tagged Image File Format, which makes GeoTIFF Geographic Tagged Image File Format), where georeferencing information is stored in the metadata, making GeoTIFF ideal for geographic image and information storage [19]. In this project the topographical data is in the Universal Transverse Mercator (UTM) projection.

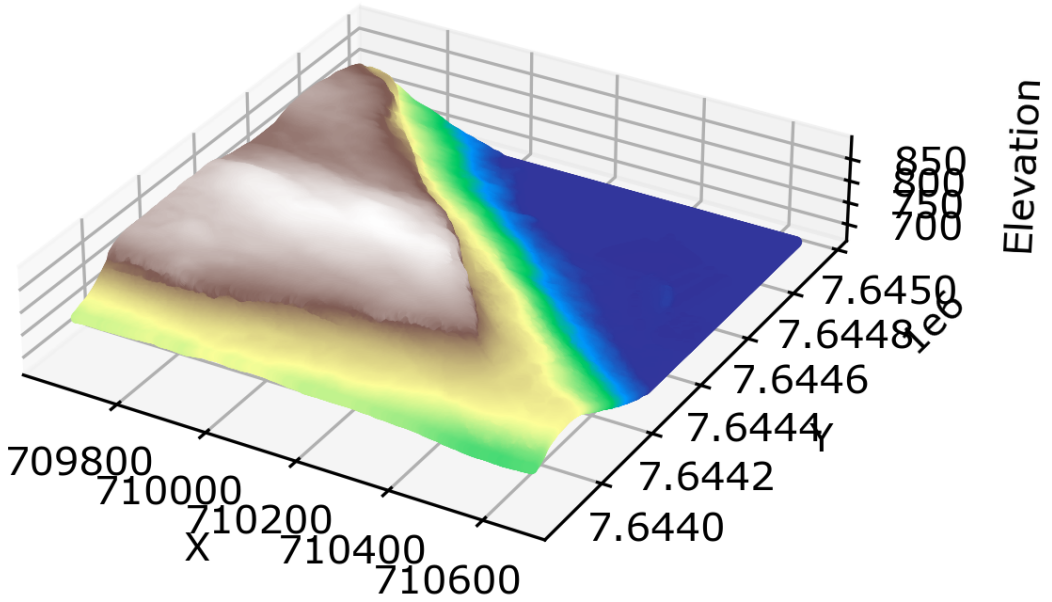

Figure 5: 3D visualization of Demand Zone.

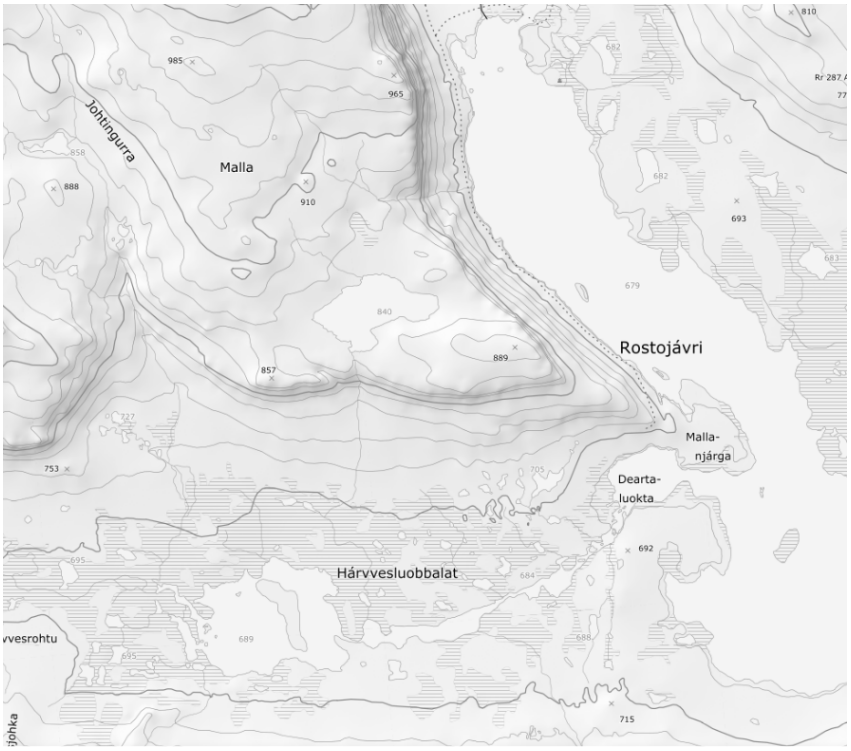

Figure 6: A topographical map of the test area (the area the dataset represents)

2.3.1 Universal Transverse Mercator

Unlike the more common geographic reference system WGS86 (latitude, longitude), the UTM system is presented as a two-dimensional plane. The northing value represents the distance from the equator to the referencing point, while the easting value represent the distance from the current zone's central meridian (with an offset that will be explained later). Both distances are given in meters. These properties allow us to work with this plane as a two-dimensional Cartesian plane. The UTM system divides the world into 60 zones. The zones use the central meridian in their respective zone as a reference for the east/west location. To ensure no negative values, the easting value for the central meridian is offset to 500 000. Consequently, the easting value 550 000 represents 50 000 km east of the central meridian, while 460 000 represents 40 000 km west of the central meridian. Within each respective zone, the projection error will not be greater than 40 cm per kilometer [20] In this project, we have data from zone 33N. Zone 33 has the meridian 15 degrees east as the center meridian.

2.4 Selected Background Literature

Optimization of facility positioning on topographical surfaces is a problem that demands significant computational power. The Maximum Covering Location Problem (MCLP) is categorized as a "non-deterministic polynomial time (NP)-hard optimization problem" [5][21]. Over the years, various strategies have been proposed to solve this problem and reduce the computational power needed. In this literature section, it is shown how the problem has been approached previously, including examples of similar problems.

Church and ReVelle formulated the MCLP in the 1974 paper "The Maximal Covering Location Problem" [9]. This problem considers that location covering problems may be constrained by some sort of budget. Realizing that a project budget may not be enough to cover the entire project area, Church and ReVelle formulated the problem as: "Maximize coverage (population covered) within a desired service distance S by locating a fixed number of facilities" [9]. The authors illustrate the necessity of this problem composition with the example of five facilities which cover 90% of the service area. An additional five facilities may be necessary to achieve total coverage (i.e. LSCP).

Goodchild and Lee [22] discussed visibility regions in topography and coverage problems in 1989. With the triangulated irregular network (TIN) as its digital elevation model (DEM), they assert that no simple theorem can derive information from one vertex's viewshed to another, and the computational challenges that result from this fact. Goodchild and Lee also introduce a corollary to location problems. Although previous discussions limited facility placement to surface level, Goodchild and Lee argued that increasing height would drastically increase the viewed area and that "since the surface z (x, y) is assumed to be single valued, there must exist some minimum height at each vertex from which it is possible to see the entire surface" [22]. This aspect gives the topographical MCLP a new parameter. How high do you mount a given facility?

Kim et al. [23] state that optimally selecting multiple sites for a network of radio masts is a combinatorial problem that cannot be solved directly, other than for the most trivial cases. Currently, viewshed calculation is a routine operation for GIS softwares. Not only can combinatorial problems not be solved directly, but the repetitive calculation of the viewshed is still computationally demanding and rapidly increases in complexity when adding examination points [23]. Kim et al. suggest reducing the candidate points (CP) to significant topographical points, such as tops. The paper points out that even though

one might think that the highest points have the best visibility, earlier research shows that the correlation coefficient between visibility and elevation is very poor (as low as 0.12 [24]). Instead, morphometric features such as ridges, pits, and peaks predict higher visibility on average. By identifying these pixels using the polynomial trend surface, the authors drastically reduced the number of candidates. They further simplify the problem by eliminating neighboring pixels (due to the high chance of similar viewsheds). They solve the problem with the remaining candidates with three different heuristics. Of these, Simulated Annealing performed best. "The results show that the use of these two strategies results in a reduction of the computing time necessary by two orders of magnitude, but at the cost of a loss of 10% in the area viewed" [23].

More recently, Bao et al. [4] use a combination of location-allocation models and visibility analysis (viewshed analysis) to optimize the locations of forest fire monitors. The watchtowers are equipped with laser night vision cameras or HD video cameras, so their positioning is critical to maximize the covered area while minimizing cost. In this study, the terrain data is represented by a raster Digital Elevation Model (DEM), and the viewshed analysis is performed with the ArcGIS GIS software view-shaded tool. The candidate points were selected mainly along ridges and hilltops in the forest. In their work, both minimum cost with full coverage (LSCP), and maximal coverage with a budget (MCLP) were calculated. By selecting CP and demand points (DP) both problems can be solved as integer programs. The study used the open source Mixed Integer Linear Program (MILP) solver LP_Solve [4][25].

Welscher et al. [5] attempt to maximize the coverage of Long-Range radio transceivers to monitor cattle in the Austrian mountains. The system used, Viehfinder, is an integrated solution for tracking animals. The animal node in this system consists of a battery/solar driven module with GNSS and other sensors attached to the animal's collar. The information collected with the node can be retrieved with a "Long Range Wide Area Network antenna," and from there be routed via cellular. The antennas are expensive, so coverage must be optimized to make them a viable solution for farmers. Like Bao et al. [4] this paper uses the DEM format and viewshed analysis as the foundation of their optimization problem. An equivalent to MCLP is introduced for antennas specifically with the name Antenna Coverage Location Problem (ACLP). This subproblem is also classified as NP-hard. The ACLP defines the zones that must be covered and the

zones where an antenna can be placed. The authors discussed three strategies to define candidate points.

1. Generate a number of random points
2. Handpick/select significant morphometric features (as in [4])
3. Implement a GIS-based approach for CP starting with the whole relevant plane as potential sites, and then begin filtering. For instance, filter out terrain steeper than 15 degrees, proximity to infrastructure such as roads, etc. When this filtering is complete, to reduce the number of CP even more, the remaining potential locations are reduced to morphometric structures such as ridges. The result is a limited set of well-qualified points ready for optimization.

Demand points, on the other hand, might be less intuitive. Generation of random points is also an alternative when it comes to DP. Various other strategies exist, such as resampling by skipping neighboring points [5]. In Welscher et al. [5] they use the GIS-based approach to select CP. They constrain their selection by proximity to roads, avoiding demanding terrain, cellular reception for remote antenna operation, and establishing outer boundaries (ensuring geographical relevancy). This paper uses ILP to solve their problem. They create a coverage matrix consisting of all the demand points (DP) (the points to be covered) represented in rows, and all the candidate points (CP) represented in columns. In addition, a demand column is added. This column allows the authors to adjust the priority by weighting demand points. By performing viewshed analysis for each candidate point, the matrix is updated with "True" if the corresponding DP is seen, and "False" if not. Ultimately, the coverage matrix and constraints are solved by the MCLP optimization program Allagash.

As a final remark in this review section, a report the author submitted as a part of the master program is included. Aasvold [26] investigated the use of conventional numerical optimization techniques such as BFGS on topographical data. The data set was a set of elevation measurements (not in any defined order), with a corresponding position coordinate (UTM). The idea was to treat the topographical surface as a mathematical plane. The function for this plane was not accessible, so the derivatives had to be approximated. The goal was to optimize towards the highest points. The objective function included an interpolator to smooth the plane, making the data continuous. As the BFGS

is a local optimization technique, to achieve the optimal elevation (the highest point in the dataset), the algorithm is sensitive to the initial guess. The solution provided stable results optimizing to local peaks, and the conclusion was that it optimized to the highest point in its "basin of attraction". This shows that conventional optimization techniques can indeed be used on topographical data [26].

3 Methods

In this section, three different methods will be presented. The first method is based on the literature, and can be classified as the conventional method in this field. It computes a CM by performing multiple viewshed analyses, and solves as an ILP problem. The second method successively deploys each transceiver to the location where it will contribute the most (considering what is already covered by deployed transceivers). It uses the Nelder-Mead algorithm to find these locations, and performs viewshed analyses as part of the search. This method is not formulated as a distribution problem as in the first method. The third method is essentially a combination of the first two. By storing data from the viewshed analyses performed in the second method, a CM can easily be produced without having to perform extra viewshed analyses. By applying the optimization algorithm from the first method, the hope is to improve the coverage generated by the second method. Results will be presented, compared and discussed in section 4.

3.1 Method 1: Integer Linear Programming on the Coverage Matrix

Among others, Bao et. al [4] and Welscher et. al [5] selected CPs and DPs and solved it as an ILP problems. The CPs are coordinates where a transceiver may be placed, while DPs are coordinates to be covered. Together these two sets of points form the axes of the CM. In this project, the Geographical Information System (GIS) software GDAL's viewshed generator is used to calculate the viewshed of a point. To create the CM, the viewshed of every CP is calculated, and which DPs that it seen from the CP is investigated. If a CP covers a DP, the corresponding cell is assigned with the value 1, and if it is not covered, it is assigned the value 0. When the CM is complete, the ILP optimization process can begin.

Note that the optimization will find the optimal solution given the CM. This is both a strength and a weakness of this method. It is a strength because solving like this will provide the optimal solution of the data provided in the CM. It is the optimal solution in that it identifies the CP's that together cover the most DP's. It is a weakness because it is a significant simplification of the real world problem, where the aim is to cover a defined area, not just certain points by placing transceivers in certain CP's. However a

good sampling of both CP's and DP's may be a good representation of the "real world problem". This will be discussed in section 4.

The problem formulation

In the CM, the DPs are represented by the rows, and the CPs are represented by columns. Let us investigate the example CM:

$$CM = \begin{array}{c} \\ \text{DP 1} \\ \text{DP 2} \\ \text{DP 3} \\ \text{DP 4} \end{array} \begin{array}{c} \begin{array}{ccc} \text{CP 1} & \text{CP 2} & \text{CP 3} \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \tag{9}$$

In matrix (9) CP 1 covers DP 2, DP 3 and DP 4, but not DP 1. CP 2 is the only CP that covers DP 1 etc. The problem is to maximize the total number of DP's seen, with the number of CP's available in the budget. In other words the problem is to maximize the number of DP's with the number 1. Let us call x CPs, and y DPs. Both of these values are binary, meaning either a CP is utilized, or not, but more importantly, either a DP is covered or it is not. The last statement implies that if a DP is covered by several CP's it is only considered "once", meaning that it will contribute to the objective value as 1 (if it is seen).

As described in section 2 the standard form for Integer Linear Programming is:

$$\min_{z} \quad J(z) = c^T z \tag{10}$$

$$\text{subject to:} \quad Az \geq b \tag{11}$$

$$z \geq 0 \tag{12}$$

The vector z stores both the decision variable x and y. As the current problem is to maximize rather than minimize, this is corrected in this formulation. It can be easily corrected by assigning a negative sign in front of the object function. To formulate our problem, constraints are added, which includes the CM in the problem formulation:

$$\min_{z} \quad J(z) = c^T z \tag{13}$$

$$\text{subject to:} \quad Az \geq b \tag{14}$$

$$z = \begin{bmatrix} x \\ y \end{bmatrix} \tag{15}$$

$$z \in \{0, 1\} \tag{16}$$

$$A = [[CM][-I]], \tag{17}$$

where $x = [x_1, ..., x_m]$ where m is the number of CPs, and $y = [y_1, ..., y_n]^T$ where n represents the number of DPs . The vector b is a zero vector with dimension $[n, 1]$. Matrix A consists of the CM with dimensions $[n, m]$ and the negative identity matrix with dimension $[n, n]$. The coverage information is as described stored in the CM, one of the two matrices in A. In the formulation, there is also the restriction on the z elements x and y to be binary, either 0 or 1, as described earlier. The vector $c = [[c_x][c_y]]$, where c_x defines the cost of placing a transceiver at each CP and c_y defines how important it is to cover each DP (weight). If all DPs are equally "valued", meaning it is equally important to cover each, then the corresponding values of c_y will be the same. If there is no cost associated with the deployment of a transceiver, the corresponding values c_x will be 0 (negative cost if it had a cost). Let us investigate the objective function closer:

$$J(z) = c_x^T x + c_y^T y \tag{18}$$

In this problem, there is no cost associated with the deployment of a transceiver, nor is the coverage of DP's weighed differently, hence cx contains only zeros and cy contains only ones. This will reduce the objective function J to be the sum of y (the sum of DP's seen):

$$J(z) = c_x^T x + c_y^T y = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \sum_{i=1}^{4} y_i \tag{19}$$

Let us further investigate the behavior of the inequality constraint. By using the example coverage (9) is further elaboration the matrix A will be on this form:

$$A = \begin{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \end{bmatrix} \tag{20}$$

We start by evaluating the inequality constraint for row 1 in matrix A. This row corresponds to DP1:

$$A(1,:)z = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \geq 0 \tag{21}$$

$$A(1,:)z = 0 \cdot x_1 + 1 \cdot x_2 + 0 \cdot x_3 - 1 \cdot y_1 + 0 \cdot y_2 + 0 \cdot y_3 + 0 \cdot y_4 \geq 0 \tag{22}$$

$$A(1,:)z = 0 \cdot x_1 + 1 \cdot x_2 + 0 \cdot x_3 \geq y_1 \tag{23}$$

$$A(1,:)z = x_2 \geq y_1 \tag{24}$$

What this sequence of equations say, is that if x_2 is selected to be one of candidate positions to actually get a transceiver (then the $x_2 = 1$), this choice will allow y_1 to be the value 1. At this point, it is important to remember that positive y's are the only parameters that contribute to the objective value, and as this is a maximization problem, the algorithm will try to get as may positive y's as possible within the constraints.

Next, let us evaluate the inequality constraint for row 4 in matrix A. This row corresponds to DP4:

$$A(4,:)z = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \geq 0 \tag{25}$$

$$A(4,:)z = 1 \cdot x_1 + 0 \cdot x_2 + 1 \cdot x_3 + 0 \cdot y_1 + 0 \cdot y_2 + 0 \cdot y_3 - 1 \cdot y_4 \geq 0 \tag{26}$$

$$A(4,:)z = 1 \cdot x_1 + 0 \cdot x_2 + 1 \cdot x_3 \geq y_4 \tag{27}$$

$$A(4,:)z = x_1 + x_3 \geq y_4 \tag{28}$$

From the final inequality (28), note that the DP4 can be seen from both CP1 and CP3. This scenario illustrates the importance of the binary constraint, because without, y_4 could potentially contribute twice to the objective value. Without the binary constraint, the problem is now stated as "how many DPs do you cover, and how many times do you cover each". The problem should be "how many unique DPs do you cover, while allowing a DP to be seen multiple times only counts as 1". To address this issue, both x and y are defined as binary variables in the program, meaning they can either be 1 or 0. To exemplify the difference, a non documented experiment was conducted: The program ran first with y as binary, and second with y as "integer". For binary, 283 out of 300 DPs were covered, giving the optimal objective value at 283. Next, the binary restriction was loosened to integer, giving 900 in objective value, while only 102 covered DPs. The mismatch in objective and the objective value is obvious, and illustrates the importance of the binary criterion.

The program

Method 1 is sometimes in later sections referred to as the ILP method. The method is inspired by various papers; of them, Welscher et al. [5] is the most similar. Welscher et al., however, used "Allagash", a software library which solves the problem using PuLP given the CM, and what type of problem (MCLP). This approach gives the project limited control over the process, and therefore in this project PuLP is implemented directly. PuLP calls on the Gurobi solver that solves the LP problem with the Branch-and-Bound algorithm. For more information about the software library, see Section 3.4. The problem formulation was constructed based on our perception of the optimization processes in [5] and [4] with the help of chatGPT.

In Figure 7 a flow chart of the ILP method is presented. The number of DP's and CP's is determined in Section 4.

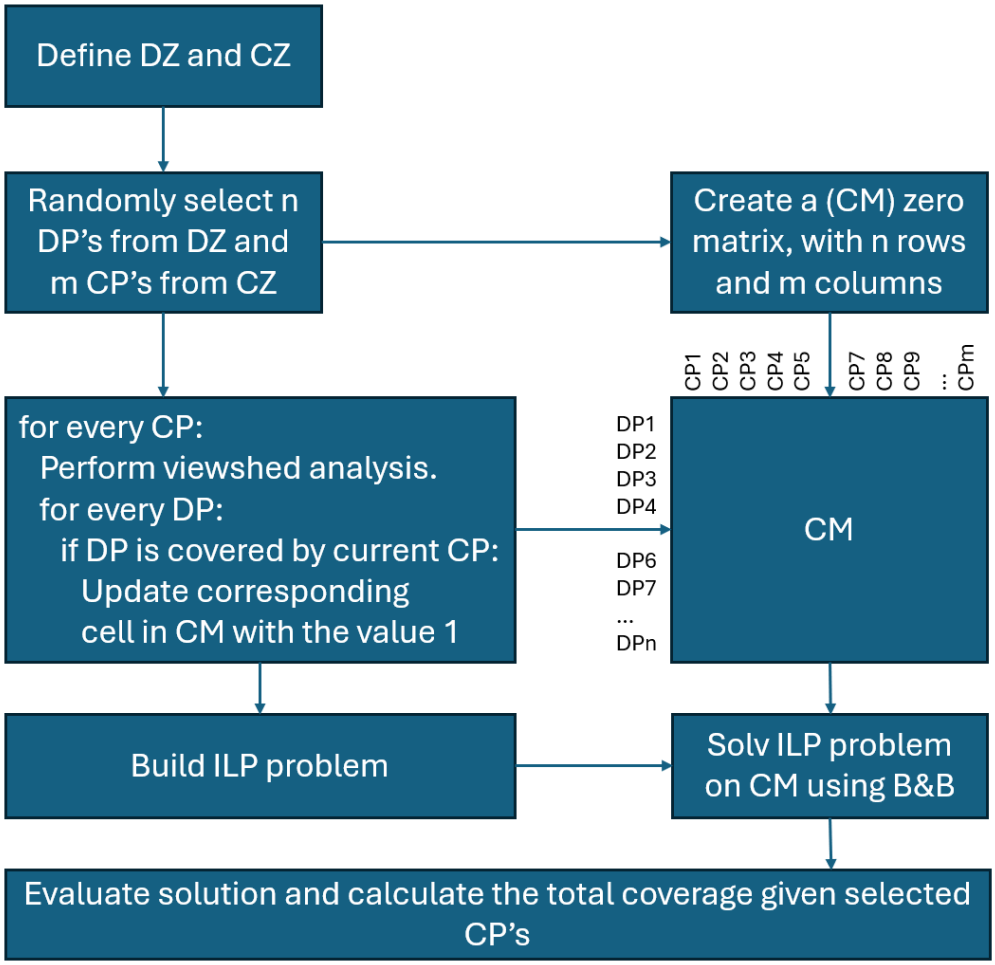

Figure 7: Flow chart of method 1's program.

Number of Transceivers

The number of transceivers to deploy is set to 35. This number is not of great importance for this project, as it is a MCLP. The explanation of why 35, is presented in the following: The area of a circle with radius = 100 meters (the range of the project transceivers) is:

$$\pi r^2 = \pi 100^2 = 31400 \tag{29}$$

The total area of our demand zone is $1100000m^2$, and further:

$$\frac{1100000}{31400} = 35.032 \tag{30}$$

Obviously there is no way to distribute 35 circles with a total area of 11000000 to perfectly cover a rectangle with the same area. This is also not the aim with this number of transceivers. The reason for computing this number is to make the total number of transceivers non-arbitrary.

3.2 Method 2: Successive Deployment of Transceivers using Nelder-Mead

As an alternative to the conventional ILP CM, this thesis explores the idea of successively finding the location of the transceiver that will improve the current coverage the most. Now, obviously this is no longer a distribution problem (one big optimization problem), but rather a sequence of subproblems to be optimized. In every subproblem's optimum, a transceiver is deployed. The idea is to create an objective function containing the viewshed analyses of the current iterate position and the pre-covered area. The pre-covered area is the area that is already covered. In our problem initially the pre-covered area is empty (no transceivers = no coverage). When a transceiver is deployed, it's coverage will be added to the pre-covered area. With this defined, we can now present the objective function. The objective function is the ratio of the covered area to the total area to the cover:

$$\frac{\text{pre covered area} + \text{iterate's viewshed}}{\text{total area to cover}}$$

When the optimization algorithm converges to a location, a transceiver is deployed, and its viewshed will be added to the covered area (pre covered). Every iteration is fed with an initial location. This location can affect where the transceiver is placed. In this project, we selected the location where the "center of gravity" of uncovered points are. There are ups and downs to this strategy discussed later on, but in short the hypothesis is that the algorithm will also handle "bad" initial positions. This process will run until its maximum number of transceivers is deployed. The ambition with this strategy is to find an alternative, less computationally demanding method to find a pleasing network structure that will provide coverage in the designated area. The product delivered by this non-combinatorial method is theoretically sub-optimal, however the hope is that it will be sufficient.

For this to work, there has to be some relation between viewshed and location. From the literature, Goodchild and Lee [22] claims that no theorem lets us derive viewshed information from one point to another. However, Kim et. al [23] want to reduce CPs by removing neighboring CPs due to the chance that those CP have similar viewsheds. This implies that even though no therorem lets us derive information from one point to another, neighboring viewshed have a high chance of similar viewsheds:

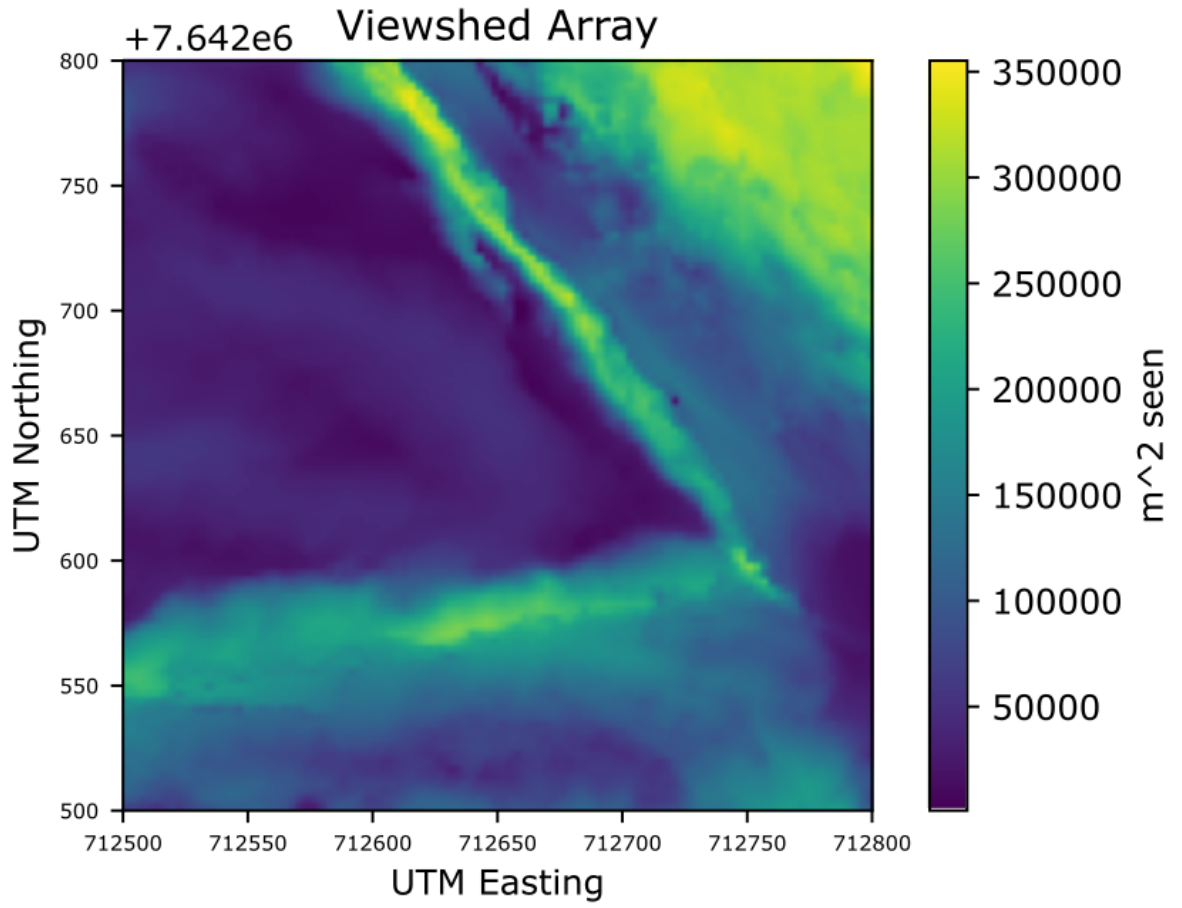

Figure 8: This figure shows the how much area that is visible from each point. It is proof of concept that neighboring pixels have similar viewsheds.

As a proof of concept to the assumption that neighbouring points have similar viewshed, computations of viewsheds for every point a mesh-grid were conducted. The result are presented in Figure 8. The plot is constructed by the area seen (m^2) as the h-coordinate, and the point's position as e and n value. The plot clearly shows structures that could be mistaken for being a topographical plot. This behavior tells us that even though Goodchild and Lee [22] claims that no theorem lets us derive viewshed information from one point to another, the relationship between neighbouring points is far from random. In the authors previous work [26], it is shown that the use of conventional optimization algorithms such as BFGS on topographical data performs well. The topographical data is structured as a 2 dimensional function $f(e, n) = h$, where e is the easting coordinate, n is the northing coordinate and h is the elevation value. In method 2 the setup is very similar, where e and n are the coordinates, and h is a value that represents coverage. In the report "Topography in Optimization" [26] an interpolator is

used to provide continuity as the elevation data is based on discrete measurements. This project on the other hand, as a raster file, is already continuous within its database.

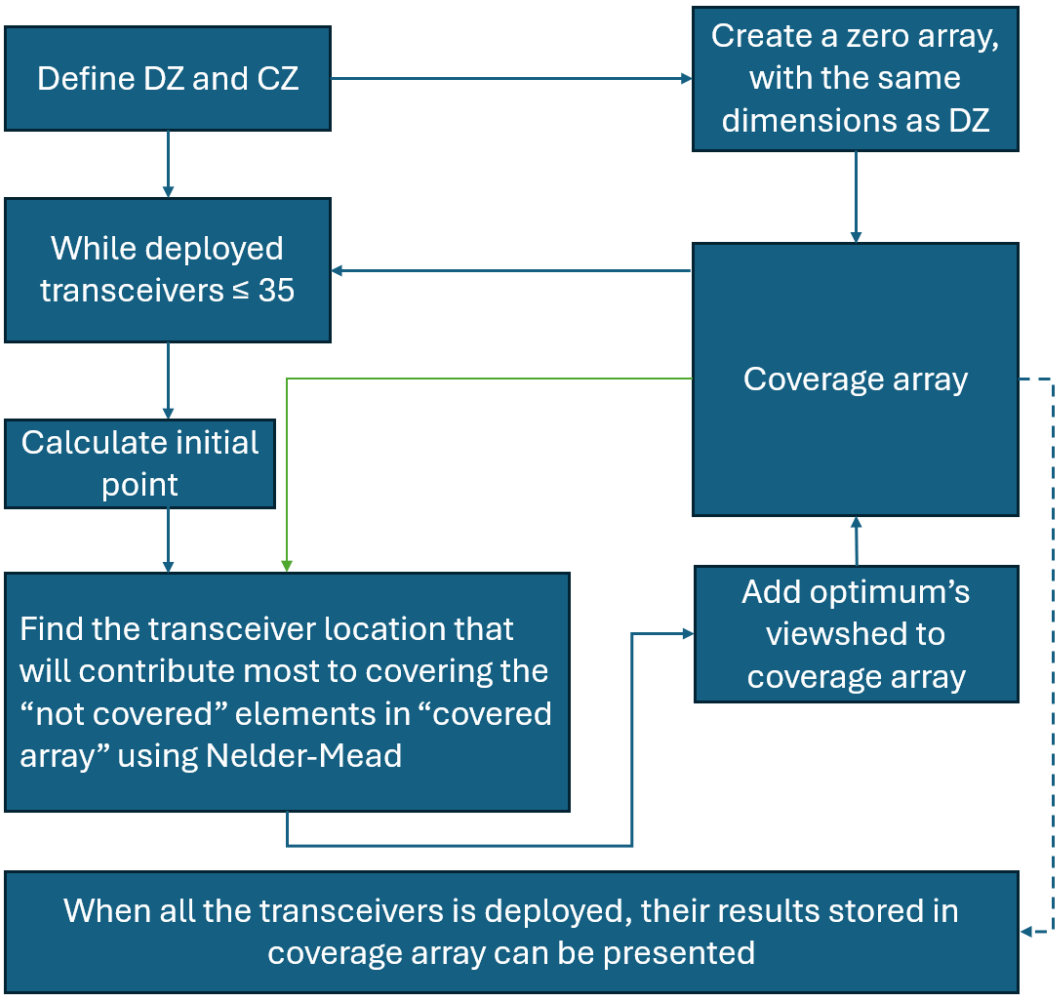

Figure 9: Flow chart of method 2's program. Dotted line is followed when "while" condition is not present, and the 35 transceivers are deployed.

To avoid having to calculate approximate derivatives, the Derivative Free Optimization (DFO) algorithm Nelder-Mead is applied. Other optimization algorithms will work with the same frame work. The Nelder- Mead is utilized this from the "scipy.optimize" library. In Figure 9, a flow chart of the program is presented. The subproblems can be formulated as follows:

$$\max_{k} \quad P(k) \tag{31}$$

subject to:

$$k_{\min} \leq k \leq k_{\max} \tag{32}$$

$$P(k) = P_{VS}(k) \cup \bar{P}, \tag{33}$$

Where $k = \begin{bmatrix} e & n \end{bmatrix}^T$ is a 2x1-vector defining the position of the current transceiver, and $k_{max} = \begin{bmatrix} e_{max} & n_{max} \end{bmatrix}^T$ and $k_{min} = \begin{bmatrix} e_{min} & n_{min} \end{bmatrix}^T$ defines the boundaries for placing a transceiver. As shown, $P(k) = P_{VS}(k) \cup \bar{P}$, where $P_{VS}(k)$ is the viewshed in position k, and $\bar{P}$ is the pre-covered area. This subproblem is decides the location for every transceiver successively. When a transceiver is deployed, $\bar{P}$ is updated by including the new transceivers viewshed.

A convergence tolerance is set on 0.1, meaning the Nelder-Mead algorithm will terminate if it can not improve more that 0.1 with the next iteration step.

In this project, the outer bounds of the algorithm are set to be the Candidate Zone. For this method, the candidate zone is set to be similar in shape to DZ, but enlarged by 100 meters (the radius of the transceivers). This is visualized in Figure 10. This is to allow the algorithm to search outside the borders of the DZ, but only in close proximity to ensure that it can contribute.

3.3 Method 3: ILP on Stored Iteration Data from Successive Deployment

The third method is essentially a combination of the two first. With the Successive Deployment algorithm, a large quantity of iteration data is computed while performing the viewshed analysis (iteration steps is shown in Figure 10). If this data is stored into an array and the correct indexing is performed, the result is a large CM. The idea in this third method is to implement this CM in an ILP, and use the same tools as in the first method to solve it. Then at last compare the result with the result from the successive deployment method.

If all the data generated from option two is used, the overwhelming size of the CM is too much to handle for the author's computer. If all values were to be kept, one

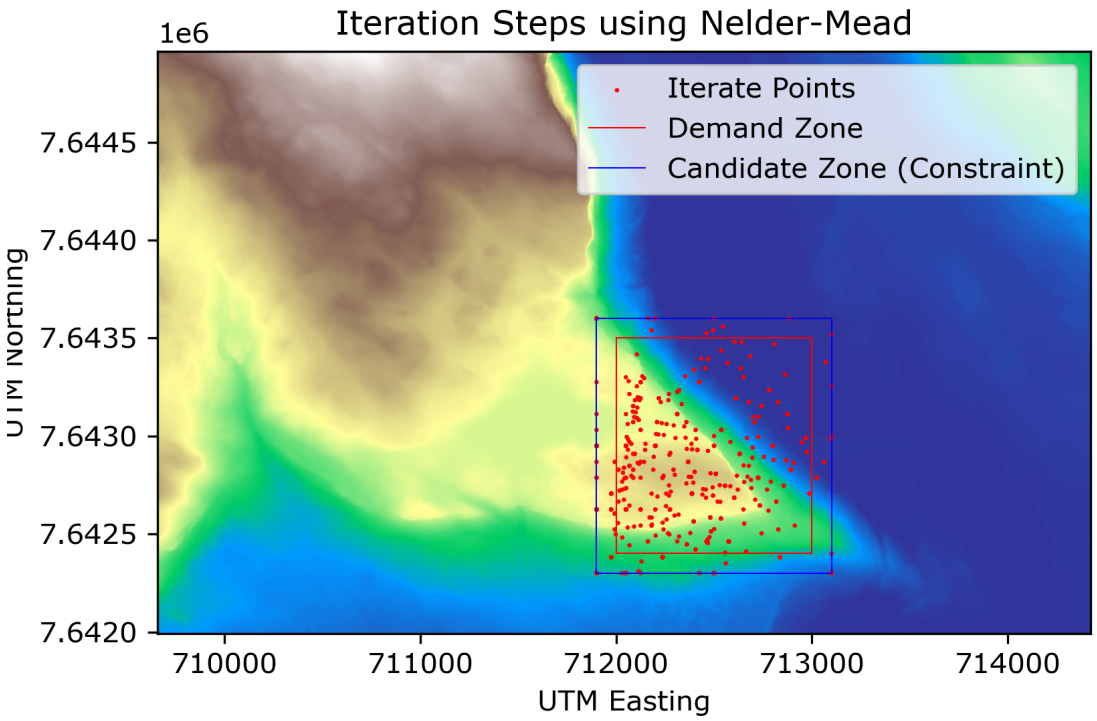

Figure 10: 744 iteration steps placing less than 35 transceivers.

viewshed would be stored in a column 1 100 000 elements in length. The deployment of 35 transceivers will need more than 2000 viewshed analyses (from section 4.3).

Therefore, in order to make the CM manageable, only a selection of rows is kept to relieve the computational burden. 5000 random rows were kept (implicit selection of 5000 DP's). Thus for each viewshed analysis, only the properties of the same 5000 DP's is kept (similar to the making of the CM from method 1). When the Nelder-Mead process is finished (all the transceivers have been deployed), another reduction is made to the dataset. All the elements that are not seen by any candidate are removed. This will not affect the result, but will reduce the computational burden.

The simplifications result in a CM without having to run any extra viewshed analyses. By applying the program developed for method 1, the problem can be solved on the new CM. In Figure 11 a flowchart of the process is presented.

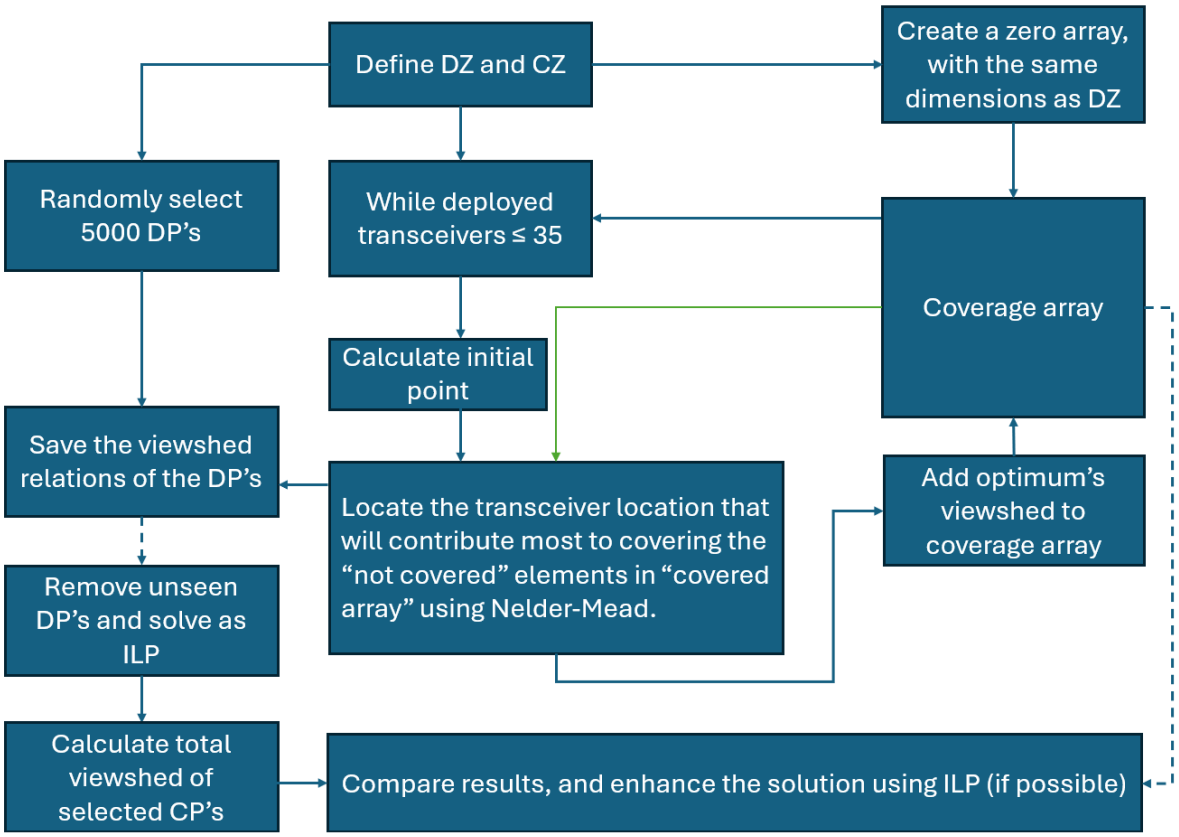

Figure 11: Flow chart of method 3's program. Dotted lines are followed when the "while" condition is not present, and the 35 transceivers is deployed.

3.4 Implementation

Project Computer

This project was conducted on a Huawei Matebook WRT-WX9. The results are subject to the computer's properties. Here is an excerpt of the technical specifications [27][28]:

- Processor: Intel Core i7-8565U:
  - Processor base frequency: 1.80GHz
  - 4 cores
  - 8 threads
- Memory: 8GB RAM

Software

The software development in this thesis has been written in Python in the Spyder and Visual Studio Code environments. The most common libraries used are briefly presented in this section. More specialized libraries will be presented in individual subsections of this.

- Numpy is a software library for scientific computing in Python. It provides various operations such as mathematics, array manipulation, and basic statistics [29]. Some of the features this project uses Numpy to is: array operations, random selection, statistical calculation, matrix transformation.
- Matplotlib is a library for the visualization of data [30]. It has been used to present the results.
- Shapely is a library for spatial object analysis and manipulation [31]. In this thesis, it has been used for creating zones ("Polygon") and points ("Point").

GDAL

Geospatial Data Abstraction Library (GDAL) is an open-source software library for raster and geospatial data. This thesis utilizes GDAL's viewshed analysis tool in its Python API. The tool is called on as GDALViewshedGenerate(), and is given the DEM as a raster file, as well as center coordinate, height above raster, and range of viewshed. The tool returns a binary masked array, where the pixels with value 1 are seen, and 0 if otherwise [32].

Rasterio

Rasterio is a GDAL based software that has many of the same functions as GDAL, but is configured towards Python (not C, as GDAL is written for)[33]. In this project, rasterio was mainly used for masking and layering on the data sets.

PuLP

PuLP is a Python LP optimization framework. PuLP can call on multiple solvers; of them, this project used Gurobi [34]. Both Method 1 and Method 3 use this library to build the ILP problem.

Gurobi

Gurobi is a professional optimization solver, free for academic purposes. In this project Gurobi's Branch and Bound algorithm is used [13].

Scipy.optimize

Scipy is a software library that provides scientific algorithms such as optimization and regression [35]. In this project, the optimizer feature is its most central contribution. The Nelder-Mead algorithm is provided through "scipy.optimize.minimize". Nelder-Mead is used both in method 2 and 3.

ChatGPT

Open AI' textrobot ChatGPT has been used as inspiration in this project [36]. It has mainly been used as inspiration in various programming challenges. ChatGPT's most impactful contribution to this project is to help formulating and stating the ILP problem for the PuLP software.

4 Results and Discussion

In this section, the results will be presented and discussed. The first sub-sections present the methods individually, before the final section compares all three methods. The main parameters are coverage performance and process time. How each method's performance changes with enlarged problems will also be discussed. Two types of enlargements are in focus. Increasing the number of transceivers and expanding the DZ. These experiments are important for understanding how well the methods will perform on larger problems. Finally, all methods will be compared and discussed.

Throughout the project, it has been observed that the computer spends consecutively more time on the same scripts, resulting in inconsistent processing time in the results. Therefore, in this section, focus on the coverage result (as they have a more constant behavior) rather than the process time accuracy. All though inconsistant, it will be demonstrated that some methods are faster than others, for instance, Method 2 is faster than Method 3.

4.1 Method 1: ILP

For Method 1, DP and CP are selected randomly from the DZ and CZ. It is necessary to decide on a reasonable number of DP and CP that represent the area in a good manner. When this is decided, the results of Method 1's performance and finally performance on enlarged problems will be presented. It will be shown that an increase in points will increase the processing time dramatically.

4.1.1 Coverage Matrix Dimension and Proxy Evaluation

In the process of determining the quantity of DP's and CP's, the number of DP's and CP's were set to be the same (to simplify the process). The assumption is that more points are better than fewer when it comes to representing the DZ; however, it will be at the cost of computational power and process time. Therefore, a trade-off will have to be made.

In figure 12 the process time on various CM sizes is illustrated in a scatter plot. The initial size of the CM is $[250, 250]$, hence $n = 250$ and $m = 250$. The script ran the same CM size 3 times before increasing by 250. This ended when a CM size of

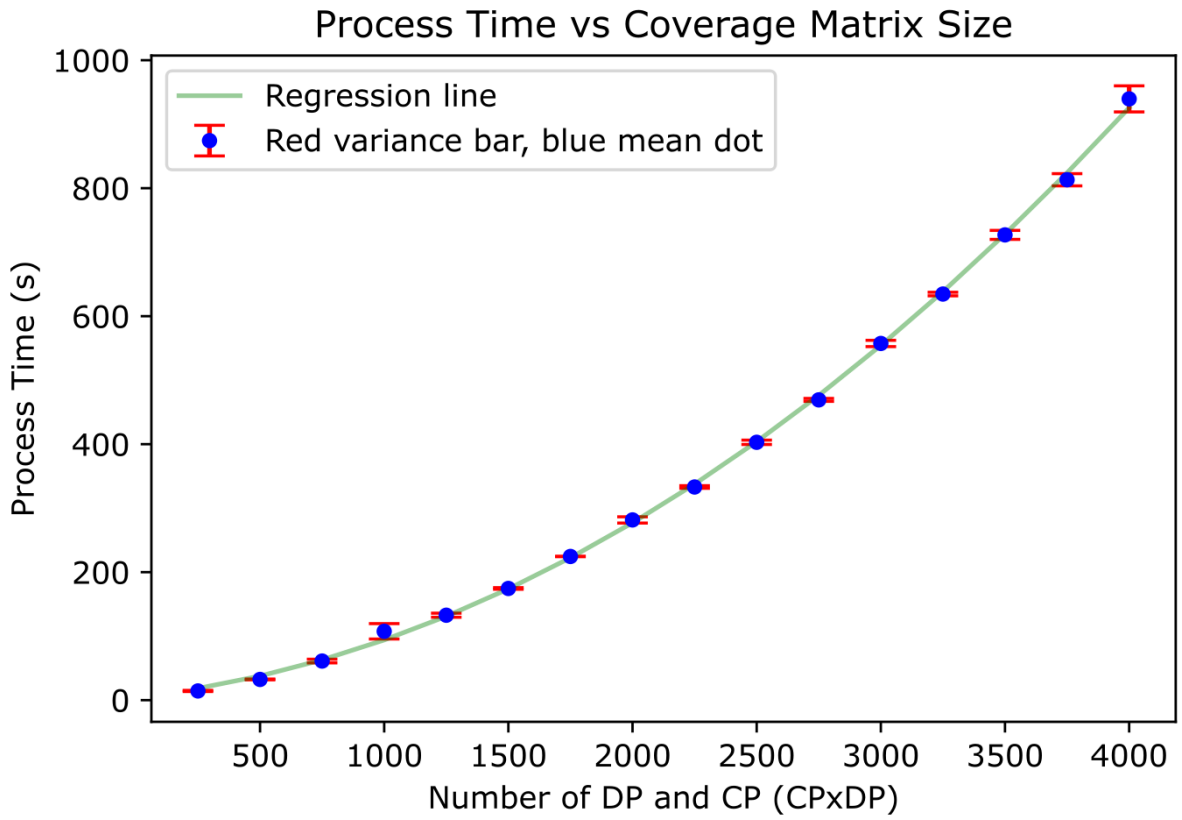

Figure 12: For every 250-unit increase in dimension, three runs were conducted. The values presented here represent the average of the three. The regression line follows the function: $y = 4.71 \times 10^{-5}x^2 + 7.70 \times 10^{-2}x + 4.96$, where x is the number of points: $x = m = n$, and y is the process time.

$[4000, 4000]$ had been running 3 times. The plot shows that there is a clear relation between process time and CM size. The regression line emphasizes the non-linear behavior in process time with the increasing CM size. The regression line is denoted as $y = 4.71 \times 10^{-5}x^2 + 7.70 \times 10^{-2}x + 4.96$, where x is the number of points: $x = m = n$, and y is the process time.

In figure 13 the same data shows the relation between covered DP vs total coverage or covered DZ. Note that the y axis is offset, and only shows the area between 50% and 70% covered. It is important to remember that the size of the CM increases, but that the demand zone does not increase, nor does the number of deployed transceivers (35). This impacts the point density of DPs and CPs. The percentage difference of covered DPs and covered DZ is highest with smaller CM dimension.

This is likely due to a few DP's not being well-enough spread to "represent" the DZ.

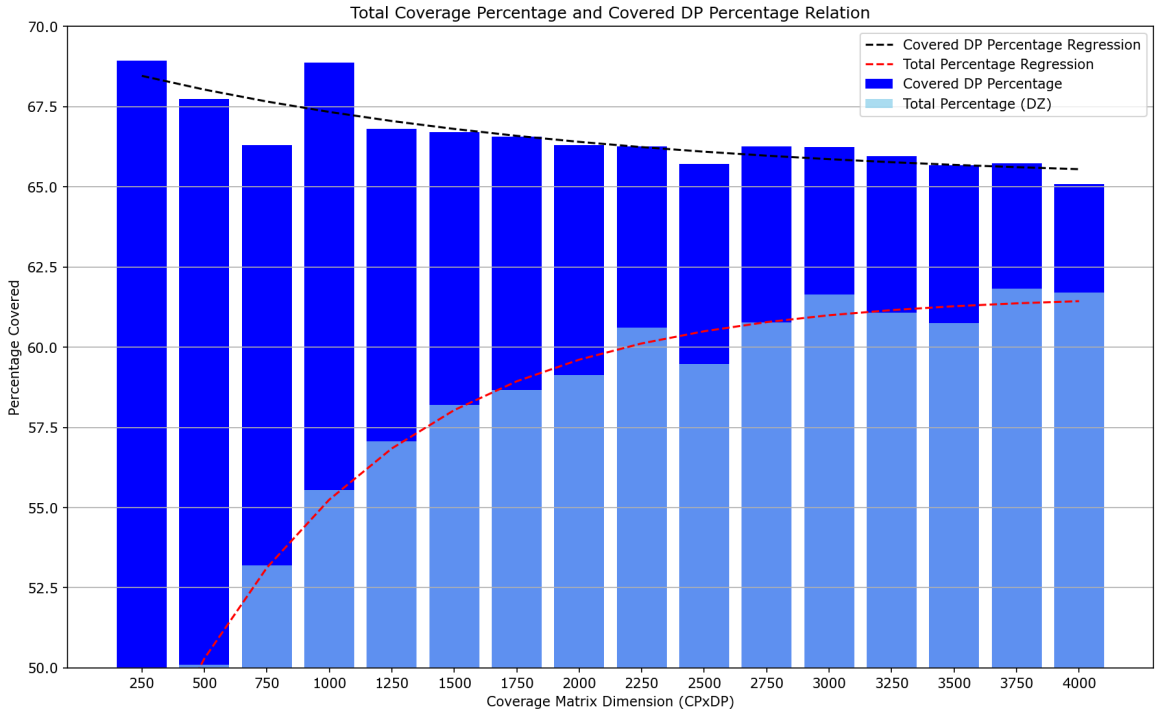

Figure 13: For every 250-unit increase in dimension, three runs were conducted. The values presented here represent the average of the corresponding three. Note that the y axis is offsetto only show the area between 50% and 70%. The regression lines are denoted: black regression line: $y = 3.34e^{-0.14x} + 65.12$, red regression line: $y = -15.12e^{-0.29x} + 61.64$, where x is the number of points: $x = m = n$, and y is percentage covered.

In this scenario, where $m = n = 250$ the transceivers are placed nearby the DP's to cover these (i.e., the optimization objective). When the number of DPs (and CM size) increase, it is observed that the percentage of covered DPs marginally decrease, and the total coverage increases. One way to explain this is to say that there are more DPs to cover but the same number of transceivers (although there are more CPs to put them on). The idea of presenting the data like this is to illustrate the hypothesis that a greater number of DPs can better represent the DZ and thereby can work as a proxy on DZ coverage. From the limited data set presented in figure 13 the beginning of convergence can be observed, as both DP and DZ covered close in and stabilize at $\sim 65\%$ and $\sim 62\%$ with 35 transceivers.

In the next tests, the CM size is 2000x2000. Also, CZ and DZ are set to be the same zone. This gives a point density of

$$\frac{2000}{1100000} = \frac{1}{550} = 0.00181818 \tag{34}$$

Another way to say this is that for each $550m^2$, there will be 1 DP and 1 CP. This is a trade-off of the DP/DZ relation and process time. Another argument for being "restrictive" on number of points is that this project has a fairly small demand zone. If Method 1 were to be applied on a larger area and the same point density were required, then the computational burden could potentially be too demanding to be a viable method for an average computer. It is also possible that the current point density overreaches if the interest area increases significantly.

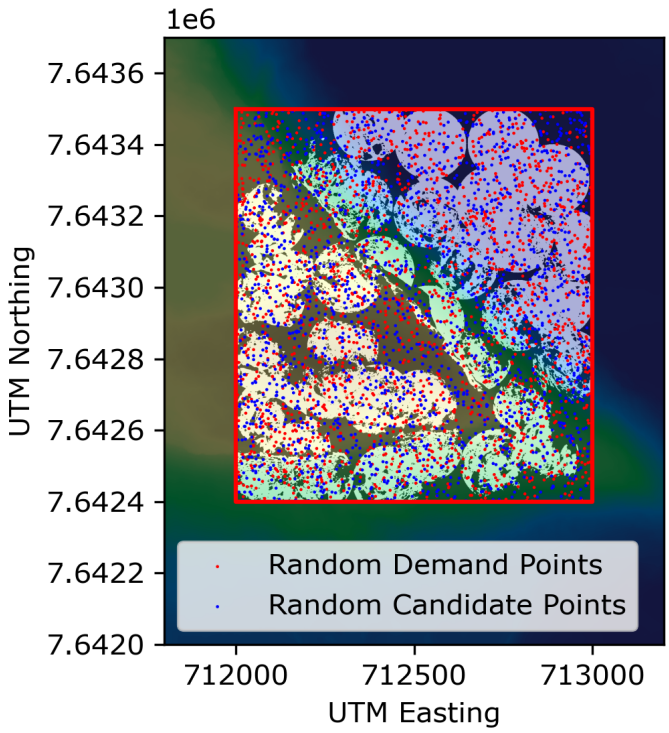

Figure 14: An example of the N.C.C. using ILP-method (Method 1). The brighter area marks the covered area. This particular run covered 66.65% of the DP, and 59.34 % of the DZ. The DZ/CZ is outlined with the red rectangle.

4.1.2 Method 1's Performance

In figure 14 an example of Network Configuration Coverage (N.C.C) is presented using Method 1, the ILP method.

Table 1 presents the results of 65 runs on the ILP. As discussed earlier, only the DP coverage is part of the ILP optimization problem. Nevertheless, the DZ coverage is interesting for at least two reasons. First, to compare the results with other methods. Second, the product of the thesis is to provide coverage over an area, not over certain

N=65	Expected Value	Variance
Process time (s)	348.00	15.18
Covered DP (%)	66.33	0.73
Covered DZ (%)*	59.03	0.53

Table 1: Key numbers from ILP method. *Solved by proxy.

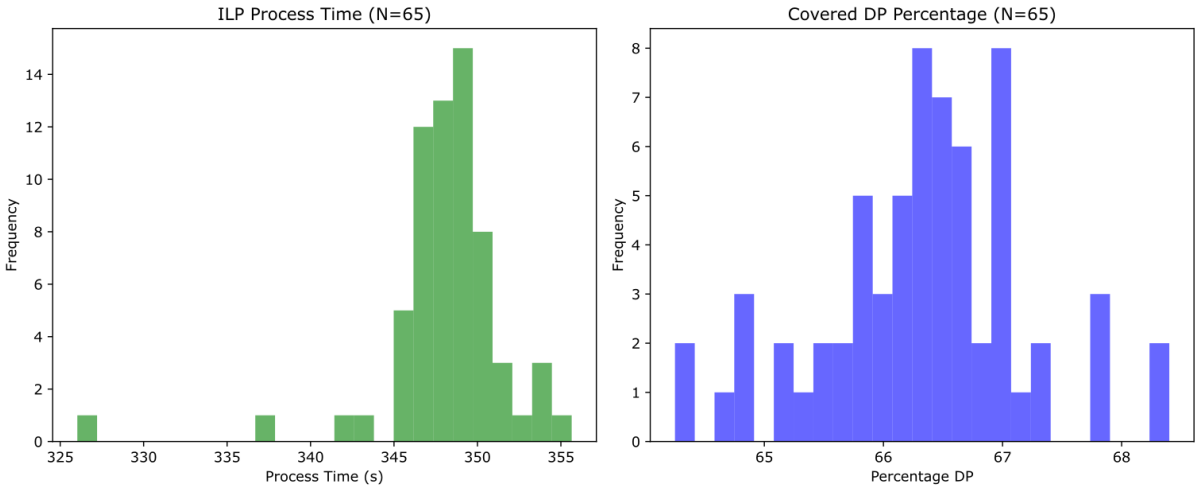

Figure 15: Histogram of the ILP-method's process time vs covered DP percentage.

points. Table 1 shows that the covered percentage of DP's average at 66.33% with a small variance of 0.73. It is interesting to see that the covered DZ percentage is fairly stable at 59.03% with only 0.53 in variance. The average process time is 348 seconds. The process time includes defining the DZ/CZ, randomly selecting DPs and CPs, performing viewshed analyses to create the CM, and solving the problem. The process time does not encompass the viewshed analyses performed on the selected CP's after the optimization is done, nor the plotting.

In figure 15 two histograms from the same 65 runs are presented. The coverage DP percentage seems to follow a Gaussian distribution, with the mentioned mean at 66.33 (0.73 in variance). The process time is more spread out, but the majority have a process time of 345 seconds and higher. The average is 348 seconds.

In figure 16, the relation between DP and DZ throughout all 65 runs are shown. Note that the y-axis in this figure is shifted. The figure shows that there is no fixed deviation between covered DP's and covered DZ. Instead, this deviation is varying with a mean of 7.3 percentage points. This demonstrates that using the ILP-method to optimize the

coverage of DPs with a suitable point density is a good proxy for optimizing DZ coverage. However, one must be aware of the deviation.

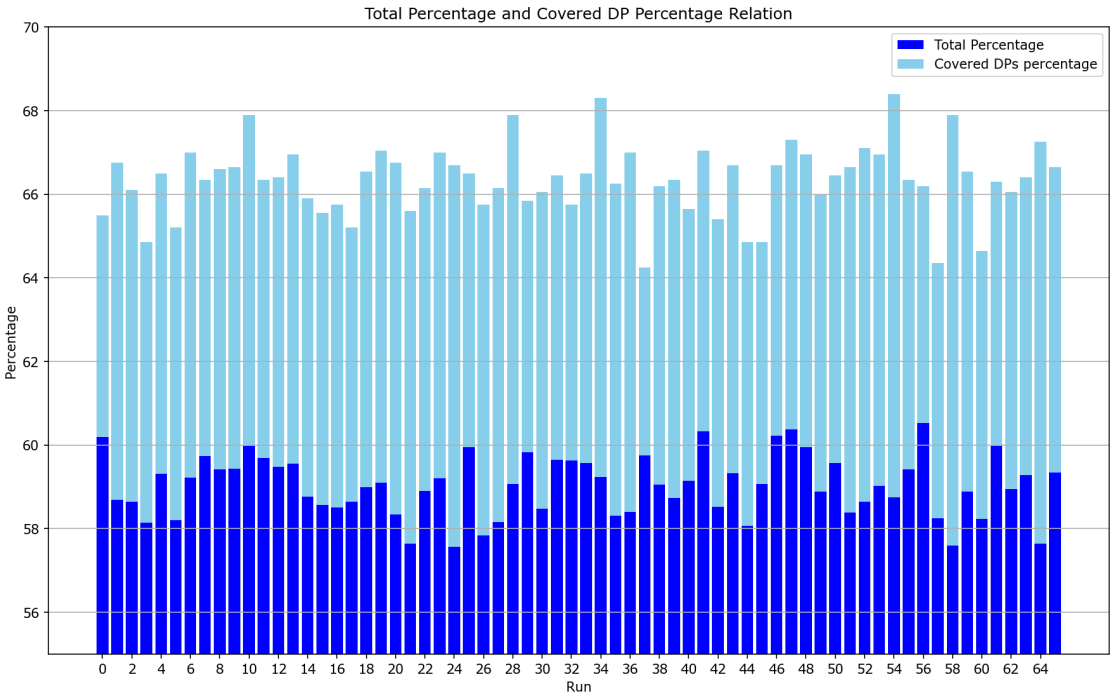

Figure 16: This barplot shows the coverage percentage of DPs and DZ for all the 65 runs. There is no fixed difference between DP and DZ coverage, however the average difference is 7.3 percentage points. Note that the y-axis in this figure is shifted.

4.1.3 Method 1 on larger problems

In section 4.1.1 showed how increased CM size affects the process time and how the density of DPs/CPs impact the proxy evaluation. In this section, the behavior of Method 1 in larger problems is discussed. If keeping to the proxy evaluation, the point density has to stay fixed. Therefore, if the area is increased, the CM dimension will also increase. As shown in figure 12, the process time follows a quadratic relation with increased CM dimension. If the DZ were to be doubled, the CM size will also double. It can be read from the figure that the process time will then be $\sim$ 950 for [4000,4000] seconds, versus $\sim$ 350 seconds for [2000,2000]. As shown, the process time increases rapidly. If the DZ were to double again, to 4 times the size of the original, the regression line can be used to estimate the process time. The CM dimension would then be [8000,8000], hence $x = 8000$:

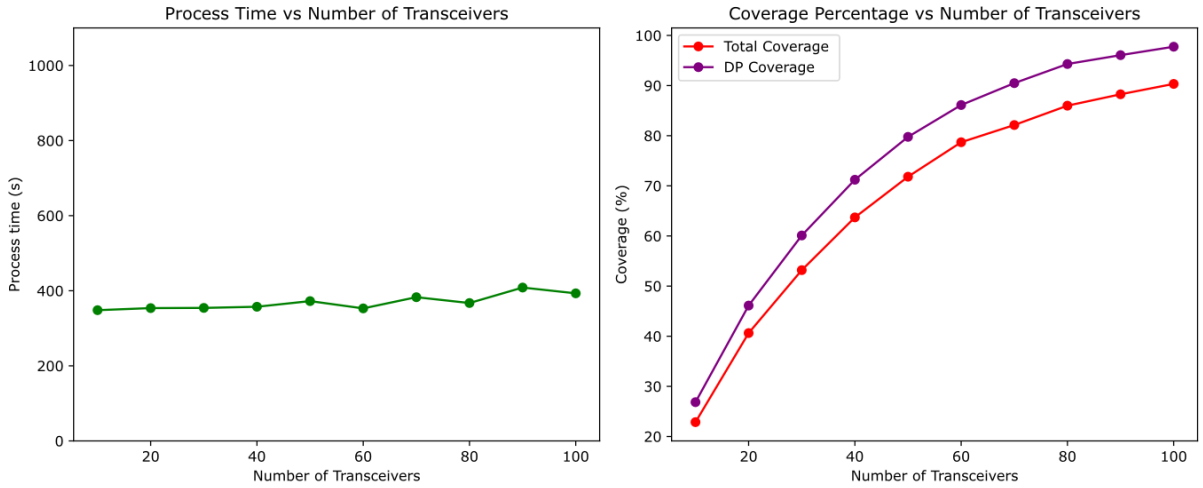

Figure 17: Method 1 behavior with increasing number of transceivers. The CM size is 2000x2000. The values are the average of 3 runs with the same number of transceivers.

$$y = 4.71 \times 10^{-5} \cdot 8000^2 + 7.70 \times 10^{-2} \cdot 8000 + 4.96 = 3635.36 \tag{35}$$

Method 1 will evidently use 3635 seconds, just over one hour to compute the coverage of 4400000m^2 DZ (∼[2100m x 2100m]). By 5000m x 5000m DZ, at 25000000m^2, Method 1 will use more than 24 hours with the same point density. This rapidly increasing process time can be reduced by lowering the number of DP's and CP's. However, then the use of this method as a proxy would have to be reevaluated.

The other test of problem enlargement is to increase the number of transceivers. In figure 17, the process time and coverage for up to 100 transceivers are shown. The process time remains close to 350 seconds for any number of transceivers. This shows that the number of transceivers does not affect the ILP calculation significantly and that the main factor for an increase in process time for Method 1 is related to increased CM dimension. In figure 17 it is also plotted how the coverage increases with an increasing number of transceivers. Deploying 40 transceivers deployed (5 more than this projects number) achieved ∼63% total coverage and ∼70% DP coverage. 80 transceivers (∼2.3 times 35) achieved ∼85% total coverage and ∼95% DP coverage. This again shows the strength of the MCLP vs LSCP. Even with 100 optimally placed transceivers ("CM optimal"), only 90% of the total DZ is covered.

4.2 Method 2: Successive Deployment using Nelder-Mead

In this section, Method 2's performance will be presented. Initially, the results from the project test (35 transceivers on the 1100000 m^2). Secondly, configuration patterns will be discussed, and finally the methods behavior on enlarged problems will be presented.

4.2.1 Method 2's Performance

N=60	Expected Value	Variance
Process time (s)	225.87	5.16
Covered DZ (%)	56.85	2.87

Table 2: Key numbers for the Successive Deployment method.

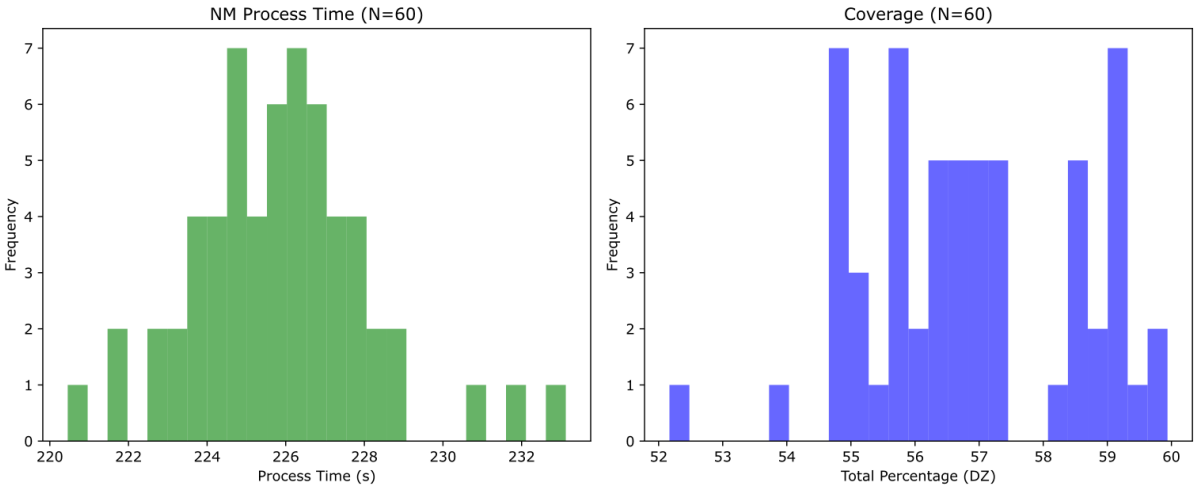

Figure 18: Histogram of the Successive Deployment data.

In this section we will review the results of Method 2, the method developed in this project. Remember that as described in the end of section 3.2, the DZ and CZ in this method are not identical, but the CZ is enlarged and similar in shape to the DZ. In table 2 the results of 60 runs are presented. As shown in Figure 18, the process time is normally distributed around its mean 225.87s with a variance of 5.16s. The coverage has a mean at 56.85 % with 2.87 in variance. The plot of a run is presented in figure 19.

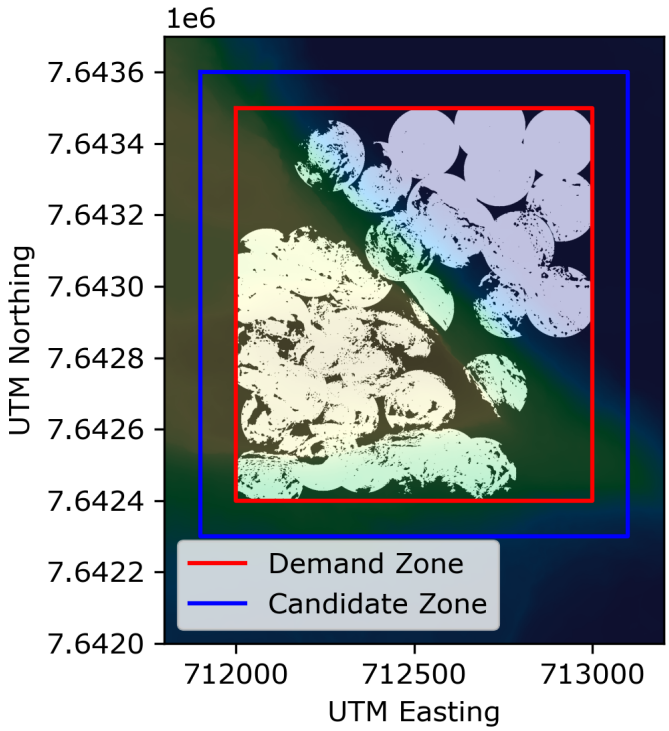

Figure 19: An example of the N.C.C using Successive Deployment (Method 2)

4.2.2 Network Configuration Patterns

A curiosity from when this method was developed was: When this program runs multiple times, will the Network Configuration (the solution) be very similar, even though it has different initial values? Or maybe not all similar, but only a limited number of patterns will unfold? The hypothesis was that the successive deployment algorithm will find some distinct (or strict) local optimums. If they early on find these optimums, then the "center of gravity" function will potentially put the next initial values in the same point, or at least the same area or basin of attraction. If these things happen, then it could be the beginning of a path that potentially could repeat itself. There might be many of these so-called paths. One argument for this not to happen is that the objective function is slightly changed every time. While true, it is only changed in a limited area around the deployed transceiver(s). For instance, if one transceiver is deployed in the southwest corner of the DZ, then the search for the second transceiver, for instance in the east, would be very similar to if the first transceiver is in the northwest corner. In figure 20 a collage of eight different runs is presented. One observation is that none of these runs have identical results. However, there are tendencies and patterns that are interesting. For instance, many of the runs provide solutions with clustering in the southwest corner, and more

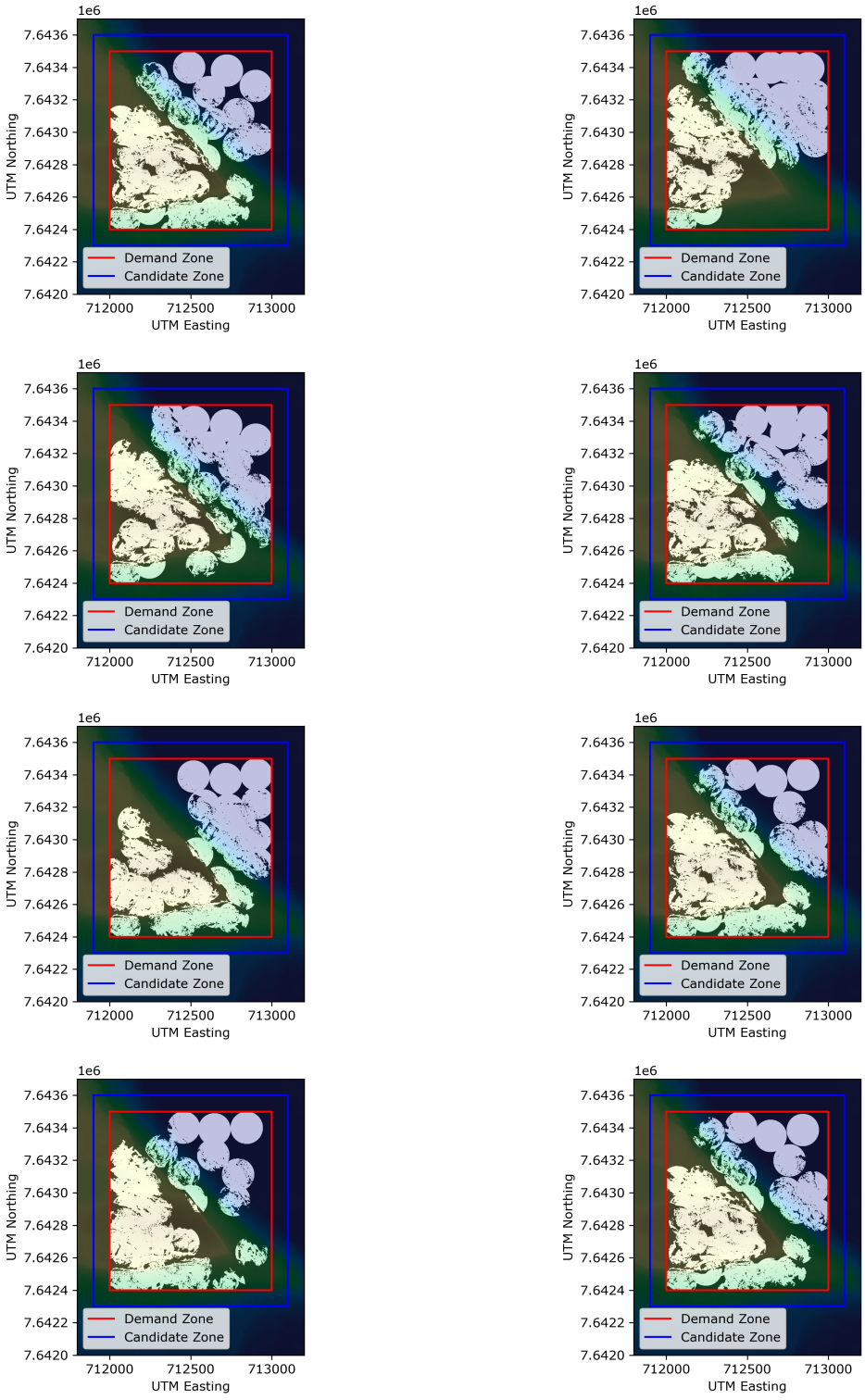

Figure 20: 8 different runs of Method 2. The brighter areas mark the covered area.

spaced-out transceivers in the northeast corner. The central cliff (from northwest corner to southeast corner), and its corners seems to be less covered than the two corners first mentioned. This is not necessarily an issue and might be close to the optimal solution of this problem. Note that description can also describe the ILP-run presented in figure 14, although with better covered corners. The terrain can be described as low flatland in the northeast, a V shaped elevated plateau with steep cliffs around them. The cliffs stretch from south west to south east (SE), before turning diagonally north west (NW). The more evenly spread-out transceivers in the north are maybe because this area lets the transceiver reach close to its full coverage potential, with little to no topographical interference. The clustering in the southwest corner is likely connected to the plateau, also a morphometric feature that the transceiver achieves a fairly large coverage. However, the topographical interference is seems higher here than in the flatlands, and the transceivers do not reach their full coverage potential. The clustering is probably a result of the partial "cover bubbles" being interwoven. There are also some positions that receive a deployed transceiver more often. Take for instance the coverage "bubble" in the southwest corner. This bubble is identical for every run in these 8 runs (9 including figure 19 (meaning a transceiver has been deployed there in all of these instances). This position is probably the optimum in a basin of attraction, that is not easily changed whenever the objective function "changes". The two lowest plots on the right, seem to have a very similar configuration. The tolerance of 0.1 (for terminating the optimization) might be a reason they are not identical. What the three areas that have least coverage (central cliff, and NW and SE corner) all have in common, is that they come with more challenging/hilly terrain. When an antenna has been deployed, and a new initial point is being calculated using the center of gravity principle, then this initial point will initially be somewhere in the central. As seen in all the terrain plots, and especially visible in the 3D plot (figure 5), the central area is dominated by the cliff. The successive deployment algorithm seems to have moved away from this cliff, either towards the flatland, or to the plateau. This is also illustrated in figure 10. This leads to the areas that generally are covered. The SE and NW corner are at the end of these cliffs, and few iterations seems to end up in this area, instead finding an optimum before they reach this area or simply searching in a different area. A possible solution to this is to change the terms of the initial value. Maybe the "center of gravity" of uncovered points is not ideal if the goal is to evenly distribute the

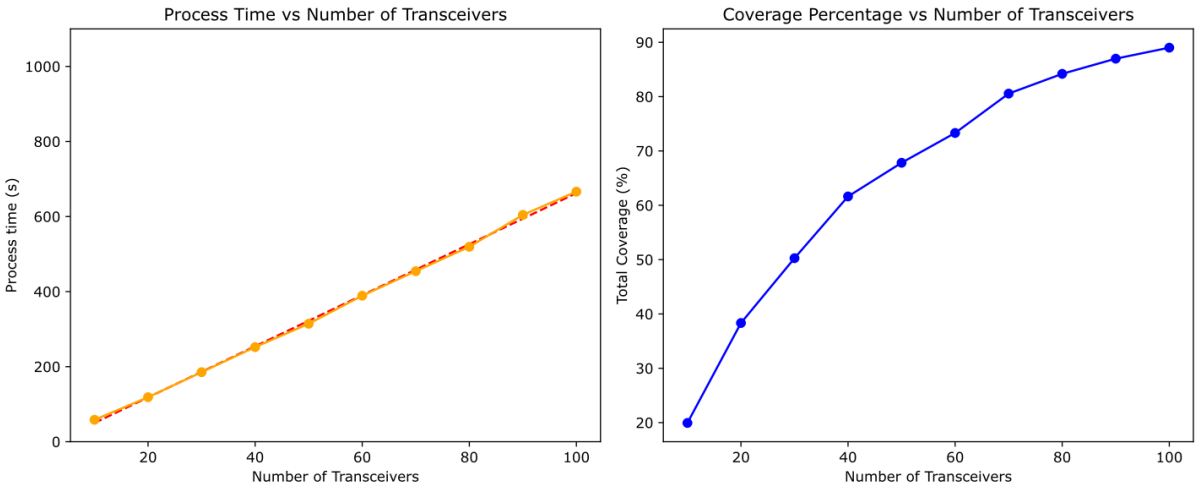

Figure 21: The second method's behavior with increasing number of transceivers. The values are the average of 3 runs with the same number of transceivers. The regression line is modeled by $y = 6.80x - 17.93$, where x is the number of transceivers and y is the process time.

coverage. Alternatives to initial position can be random initial point, center of the most uncovered quadrant etc.

4.2.3 Method 2 on Larger Problems

Figure 21 shows what happens to the process time and coverage when the number of transceivers increases. As expected, the process time increases linearly. The behavior can be approximated with the linear regression function denoted $y = 6.80x - 17.93$, where y is the process time, and x is the number of transceivers. This will not be good representation of the behavior closer to zero, but as x increases it represents the data in a satisfying manner. The regression line indicates that for every transceiver added, 6.8 seconds is added to the process time on average. The plot also shows that the coverage gradually converges and that a new transceiver's contribution reduces as the quantity of transceivers increases. It can also be seen that with 100 transceivers, Method 2 achieves a total coverage of ∼90% with a process time of ∼663 seconds.

Next, in figure 22, the process time behavior is illustrated when the demand zone expands. In addition, this data set seems to follow linearity and can be approximated with the regression line $y = 1.57x \cdot 10^{-4} + 115.19$, where y is the process time and x is the area of the demand zone. From the plot and the regression line, we can extrapolate

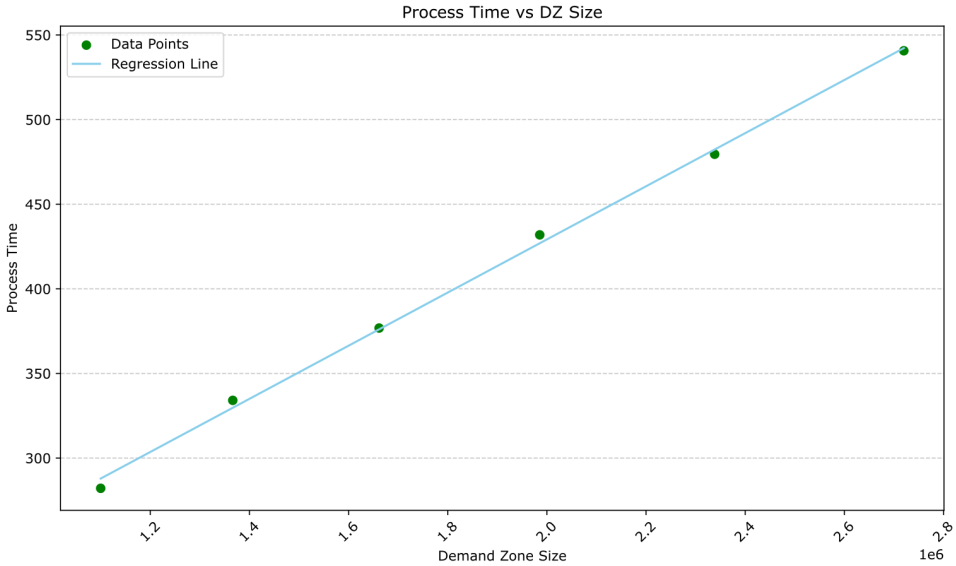

Figure 22: Method 2's behavior on expanded Demand Zone. The datapoints are an average of 10 runs. The regression line is denoted as $y = 1.57x \cdot 10^{-4} + 115.19$, where y is the process time and x is the area of the demand zone.

that by doubling the DZ from 1100000 to 2200000, the process time increases from ∼290 seconds to ∼460 seconds (an increase of 59% process time at 100% increase DZ.

4.3 Method 3: ILP on Successive Deployment Iteration Steps

In this section the results from Method 3 will be presented and discussed. This method combines features from the two previous methods, and the hope is that by applying the ILP method on the successive deployment steps, it will improve the result.

4.3.1 Method 3's Performance

N=108, mean(var)	Successive Depl.	Enhanced ILP*	Difference
Process time (s)	317.88 (33.34)	380.10 (53.48)	62.22
Coverage DZ (%)	56.27 (3.30)	59.49 (2.55)	3.22

Table 3: Comparing data of the combined solution. *Calculated by proxy, Average CM dimensions ([CP,DP]) were: [2493,4250].

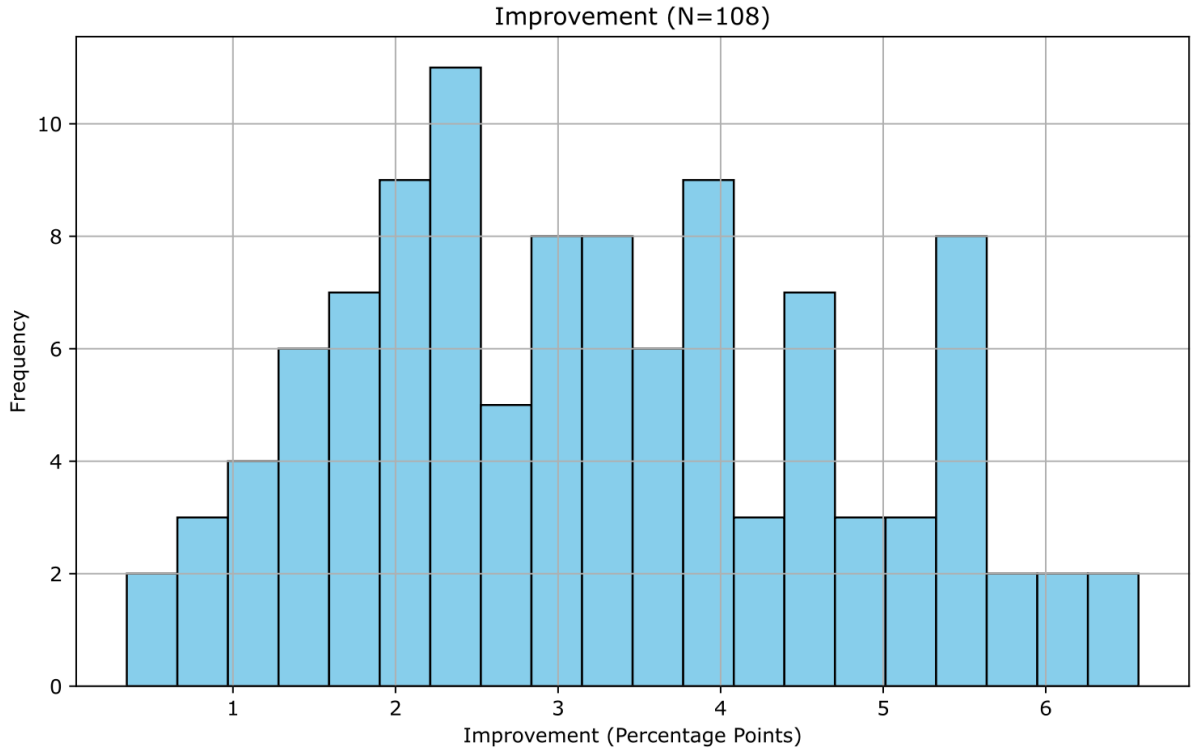

Figure 23: Histogram of improvement using Method 3. Note that 17 of the 108 runs improved more than 5 percentage points from Successive Deployment to the enhanced result!

First, note that for Method 3 the average CM dimension after the successive deployment part was performed were [2493,4250]. Table 3 shows that running ILP on the

generated CM from the successive deployment generates a 3.22 percentage point mean increase. This results in an average total coverage of 59.49%. As elaborated in Method 1's results, this number is calculated by optimizing the covered DPs. Figure 23 presents a histogram of the difference in cover of before/after applying the ILP was performed. Note that 17 out of 108 runs increased the coverage by more than 5 percentage points, and that not a single run got a lower coverage with ILP then successive deployment. Figure 24 presents a run of Method 3 . The left figure illustrates the coverage from the successive deployment part, and the right illustrates the coverage after performing the ILP program on the successive deployment steps. Although the difference between these two plots is not obvious, the final network configuration had an increase in coverage of 2.07 percentage points.

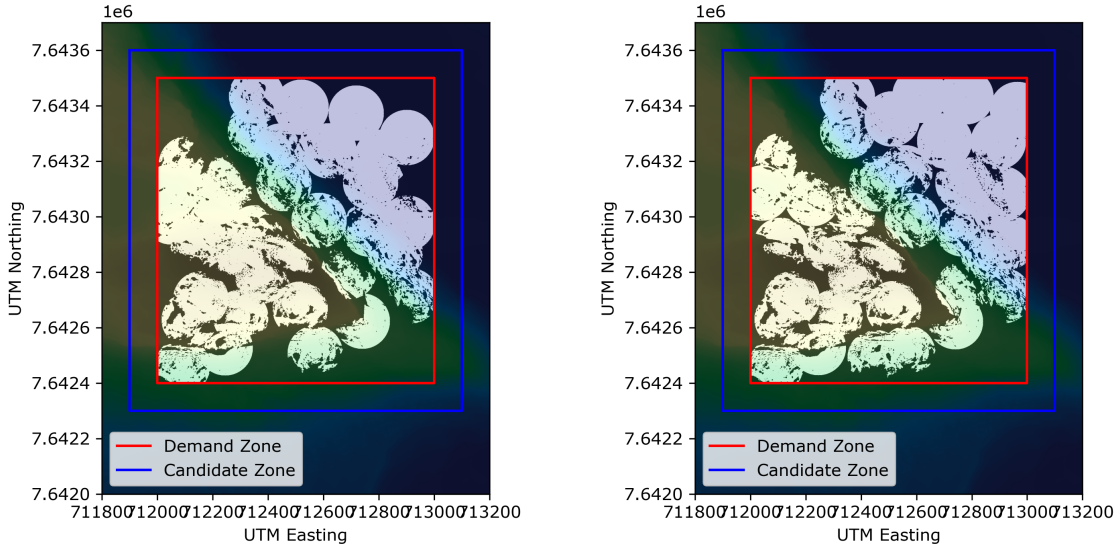

Figure 24: N.C.C using successive deployment vs enhanced coverage using ILP. In this particular run, the improvement is 2.07 percentage points.)

It is natural to compare the results from Method 3 presented in the table 3 with the two first methods' results. The expected value of successive deployment is naturally very similar, being the same program basis. The variance is marginally larger in Method 3 than in Method 2. The process time in Method 3 is 317.88 s vs. 225.87 s in Method 2. The reason for this difference is believed to be the creation of the CM. For every viewshed analysis performed, the randomly generated DP's is read from the result array, and stored as a new column in the CM. This process differs between Method 2 and Method 3. The variance is also greater in Method 3, easily explained since the successive deployment

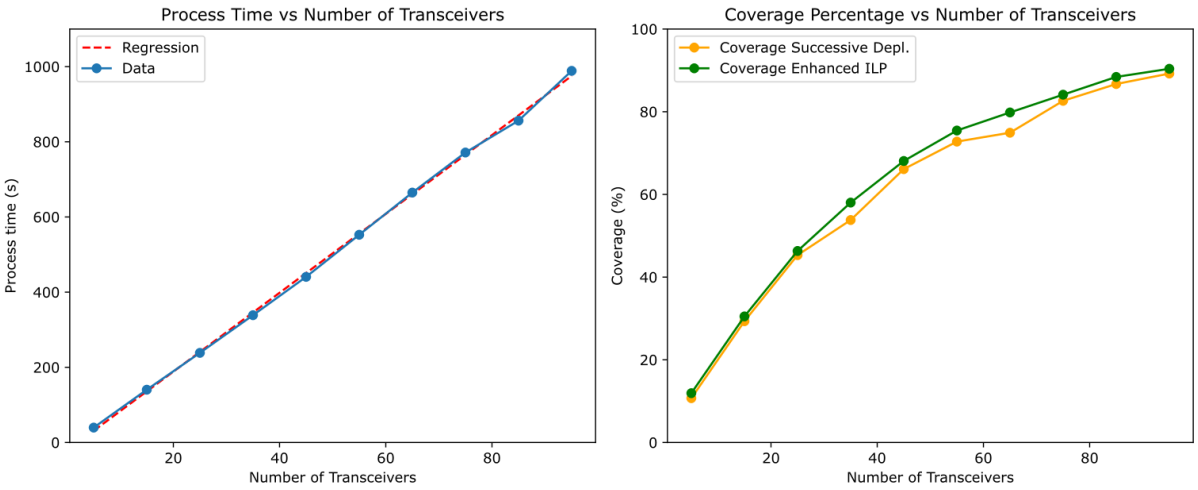

Figure 25: Method 3's behavior on an increasing number of transceivers. The data points are an average of 3 runs. The regression line is denoted as $y = 10.49x - 21.37$, where x is the number of transceivers, and y is the process time.

does not use the same number of iterations per deployment. The average, however, is 2493 iterations to deploy all 35 transceivers. The process time until the result has been enhanced using Method 1, is on average 380.10 seconds. This process time includes both the initial successive deployment, the creation of the CM and finally the ILP solving of the CM. In average, the enhanced result is produced 62.22 seconds after it the successive deployment is produced. The process time variance at this point is 53.48, however this is accumulated and therefore includes the variance from successive deployment.

4.3.2 Method 3 on Larger Problems

Figure 25 shows Method 3's behavior when the number of transceivers increase. Here the process time is also linear, and follows the regression line with function $y = 10.49x - 21.37$, where x is the number of transceivers, and y is the process time. This implies that 10.49 seconds are added to the process time for every added transceiver. It is also shown that the coverage generated by successive deployment and the enhanced coverage follow each other closely. The enhanced coverage also seems to follow a smoother path than the successive deployment solution. This may suggest that the enhanced solutions are stable and robust towards "bad" successive deployments (local optimum with not particularly good global objective value). On expanded DZ Method 3 also follows a linear increase. This is expected, as both expansions of Method 2 followed linearity and Method 3's behavior

on increased number of transceivers was linear. However Method 1 with increased CM demonstrates a quadratic behavior. This indicates that it is not the ILP solver that has the nonlinear behavior, but the generation of CM in method 1. In Method 3 the CM is made by storing viewshed data from the iteration steps in the successive deployment method. Figure 26 shows the relation between DZ size and process time for Method 3. Although the data points in the figure do not follow the regression line as closely as method 2's test, the behavior is identified as linear. However more data is need to improve the estimate on larger problems.

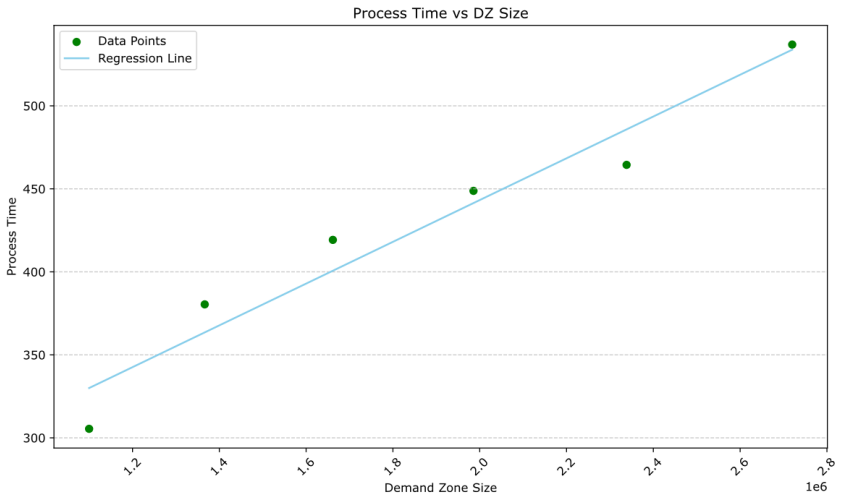

Figure 26: Method 3's behavior on expanded Demand Zone. The datapoints are an average of 5 runs. The regression line is denoted as $y = 1.26 * 10^4 x + 191.60$, where x is the DZ size, and y is the process time.

4.4 The Three Methods Compared

All together, the three methods have very interesting results. In table 4 the key numbers of all methods are gathered.

mean(variance)	**Method 1 (N=65)**	**Method 2 (N=60)**	**Method 3 (N=108)**
Process time (s)	348.00 (15.18)	225.87 (5.16)	380.10 (53.48)
CM dimension	[2000,2000]	N/A	[2493,4250] (mean)
Covered DZ (%)	59.03 (0.53)*	56.85 (2.87)	59.49 (2.55)*

Table 4: *Calculated using proxy

4.4.1 Performance on Project Problem

Let's define the project problem as the DZ with $1100000m^2$ and 35 transceivers. Overall, the best coverage is achieved with Method 3, which achieves 59.49% covered DZ, using a process time of 380.1 seconds. Marginally behind is Method 1, which covers 59.03 of the DZ in 348 seconds. The fastest algorithm is Method 2, using 225.87 seconds to compute a network configuration covering 56.85% of the DZ. Method 1 produces the second fastest configuration. The coverage difference between Method 1 and Method 2 is 2.18 percentage points in favor of Method 1, but Method 2 has a 35% reduction in process time. Methods 1 and 2 will be the two primary methods compared, due to their fundamental differences. As explained, Method 3 is essentially these two merged. Arguably, one of the most interesting findings from the project test results is that although Method 1 solved the problem as a combinatorial distribution problem, it only achieved 2.18 percentage points more coverage than Method 2. In practical terms, given that choosing Method 2 saves 35% process time, one might consider adding a few transceivers to Method 2, to equal the coverage gained with Method 1, if process time were the most important factor. The fact that Method 2 is so close in coverage compared to Method 1 is interesting from a development point of view. Method 2 and Method 3 were developed in this project. The methods are not polished, and it is the author's belief that there is more potential in the idea that what has been accessed. For instance, optimizing the optimization algorithm settings (in this case the Nelder-Mead settings) or improving the initial position are two areas that might bear fruit. In addition, a skilled programmer will also be able to optimize the code further, reducing the processing time even more.

4.4.2 Performance on Enlarged Problems

In this project two types of enlargements are evaluated. The first is increasing the number of transceivers and the second is DZ expansion. Let us first evaluate an increasing number of transceivers. In figure 27 the processing time and coverage percentage for an increased number of transceivers for the three methods are plotted. It is shown that with less than approximately 55 transceivers, Method 2 performs the configuration faster than Method 1. However, the ILP method does not increase in processing time with an increase in the number of transceivers. Method 2 on the other hand increases by 6.80 seconds per transceiver, according to the regression in figure 21. This means that with more than 55

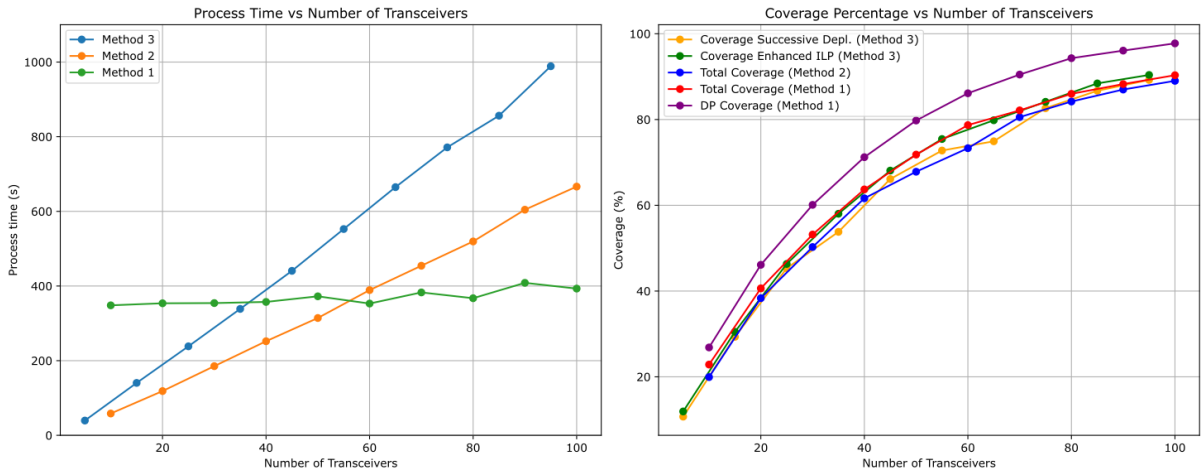

Figure 27: This figure compares all the increasing transceiver results in the same plot. Note that Method 3 run from 5 to 95 in intervals of 10, while Method 1 and 2 rund from 10 to 100 in intervals of 10.

transceivers Method 1 will configure the network of transceivers faster than Method 2, given the size of the CM does not change (2000x2000). Method 3 also performs faster than Method 1 for a low number of transceivers, and has a linear increase, steeper than Method 2, with approximately 10.49 seconds added per transceiver.

For the second type of enlargement, expanding the DZ, the points density in Method 1 is assumed to be the same. As presented in sections 4.1.3, Method 1's process time with increasing CM dimension is shown in figure 12. For a doubled DZ (giving CM size 4000x4000), Method 1 will use $\sim$ 950 seconds in process time. Method 2 on the other hand will use 460 seconds, according to figure 22. The tendency is clear: methods 1's nonlinear increase and Method 2's linear increase favors Method 2 in process time. Remember Method 1 exceeded 24 hours in process time with a demand zone of 5000m x 5000m. The regression from figure 21 can be used to estimate Methods 2 process time on $2500000m^2$(5000m x 5000m):

$$y = 1.57x \cdot 10^{-4} + 115.19 \tag{36}$$

$$y = 1.57 \cdot 25000000 \cdot 10^{-4} + 115.19 = 4040.19s, \tag{37}$$

where x is the area given in m^2. Method 3 also demonstrates linear behavior and may be a trade-off between Method 1 and 2.

4.4.3 Which Parameters are Most Important?

The two main parameters discussed in this thesis is process time and percentage covered. Naturally, coverage is one of the most important factors in an "optimizing coverage" problem, however is this also the case for process time? Three methods have been presented in this thesis. The conventional Method 1 provides the best coverage (61.61%, with an average process time of 302.36 seconds). Method 2 provides an average 56.85% coverage in only 225.87 seconds. This is a reduction of $\sim$ 35% process time. In many real-world scenarios, the coverage configuration of an area is something calculated once, and when you have the optimal configuration, there is no need to calculate it again. In these scenarios, if the time is not critical, the process time is arguably not the most important parameter. However, as discussed, the computational power needed to compute large MCLP's is significant. In section 4.1.1 the process time behavior with increasing coverage matrix dimension was presented, and section 4.4 discussed the behaviour of an enlarged problem using Method 2. If the DZ were to be enlarged and the same point density kept, Method 1's process time increases quadratically, while Method 2 follows linear behavior in the same scenario. With a large area increase, this can become a problem for Method 1. As mentioned, the project computer already pushed far above 90 % memory usage during results production. In the development of Method 3, the computer had problems with the handling of the large CM, and the author eventually reduced the data size (by randomly selecting points to store). For the 5000m x 5000m DZ used as an example, the CM size would be 45455x45455 if the point density were fixed ($\sim$ 29 hours processing time). With these large CM, even though processing time is not the priority, the problems might be too large to solve for an average computer.

One way to reduce the processing time is to reduce the point density. However, with randomized point selection, fewer points will reduce the quality of the representation. Therefore, another point-selection approach should be conducted. First, one does not have to have the same number of DPs and CPs. If reducing the CM is the goal, one could start with the CPs and select only points that are considered "good candidates". Two qualities within the set of CPs are that the point achieves a high percentage of its potential and that the points are well spread. As described in the literature, Bao et al. [4] discuss 3 strategies to define CPs. One of them is the randomized as performed in Method 1. The other two are selecting points at significant morphometric features

either by “handpicking” or the more advanced; by using GIS to filter. A fourth option might be to use Nelder-Mead or another DFO to search for points that achieve a high percentage of their coverage potential. If reducingDPs is required as well, one might start by prioritizing the areas that actually need coverage and discard the points outside these areas.

4.4.4 Why is Method 1 Not Superior?

Why does Method 1 not produce a far better N.C.C. than Method 2? As shown, there is only a 2.18 percentage point difference between Method 2 and Method 1. Given that Method 1 is a combinatorial distribution problem, this method would be expected to produce a significantly better configuration than Method 2, which only considers one transceiver at the time. Following are some discussions of why we get these results.

- Transceiver range: Given that these transceivers have only 100 meters radius, it is easier for the algorithm to locate them at points where they achieve near full coverage potential, than for long range transceivers, where topographical interference will be a greater problem. With more interference, only segments of the transceivers coverage potential is projecting the ground. In these scenarios, Method 1 is expected to perform better, selecting locations with segments that can be combined as a puzzle.

- Demand Zone size : Given the large DZ, the transceivers do not need to be interwoven to contribute with their full potential. The transceiver would prefer to be positioned somewhere it achieves a large percentage of its potential and no overlap. If the DZ on the other hand was smaller and the 35 transceivers did not have enough space to avoid overlap, the power of combinatorial distribution would be more visible.

- All versus a selection of CPs: Method 1 selects n random CPs while Method 2 can search the whole CZ for suitable CPs. Naturally, Method 2 can search for the "best location" every time; however, CP is restricted to the selected.

5 Conclusion

In this report three methods for computing a network configuration for low range transceivers in terrain have been presented, tested and discussed. The methods' behaviour on enlarged problems have been analyzed.

The first method is based on principles from the literature. It creates a coverage matrix based on viewshed analyses, and solves the problem as an ILP problem with the coverage matrix as its input. The second method is developed in this thesis. The method successively deploys transceivers at the best location in order to improve the current state. It uses an optimization algorithm for nonlinear programs to find these optimums. In this project the DFO algorithm Nelder-Mead was used. Method 2 does not solve the problem as a distribution problem, but rather a series of subproblems. The third method is essentially a combination of the first two. It saves viewshed information generated from the viewshed analyses conducted while performing method 2 into a coverage matrix. By applying method 1 on "pre-made" coverage matrix, the result is enhanced.

The CM for method 1 was created by selecting a number of random DPs and CPs. 2000 points of each were chosen to represent the DZ. This number is a trade-off between processing time and closeness in DP coverage and DZ coverage. More points lower the gap between DP and DZ coverage.

The methods were initially tested on a rectangular DZ with size $1100000m^2$, and configured the location of 35 transceivers. The results show that method 3 provides the highest coverage with 59.49%, and method 1 marginally behind with 59.03%. Method 2 deploys 35 antennas in 225.87 seconds, a $\sim 35\%$ reduction in time compared to Method 1 (348.00 seconds), but with a coverage 2.3 percentage points less than Method 1's (59.03 with method 1, 56.85 with method 2). Method 3 on the other hand produces its coverage of 59.49% in 380.1 seconds, thereby improving the Nelder-Mead with 3.22 in average difference.

Two types of enlargement were tested on the three methods: Expanded DZ and increased number of transceivers. For the expanded DZ, Method 1 is set to have the same density of DPs and CPs. It is found that Method 1's process time models quadratic behavior. While it uses 348 seconds in a CM size of $n = m = 2000$ and ~ 950 seconds in double the area ($n = m = 4000$ representing $2200000m^2$), it would use more than 24 hours if the DZ were 5000m x 5000m ($25000000m^2$). Method 2 has linear behavior that can

be estimated with a regression function. It is estimated that method 2 would need 4040 seconds process time to configure a network on a DZ of 5000m x 5000m ($25000000m^2$). Method 1's process time does not seem to be affected by the number of transceivers. Method 2 and 3 increase linearly with approximately 6.8 seconds for Method 2 and 10.49 seconds for Method 3. For real world enlargements, both of these enlargements would likely be performed simultaneously. In that situation, the process time of Method 1 will increase quadraticly while method 2 and 3 will increase linearly.

Process time is not necessarily the most important parameter, and in many scenarios there might only be a need to compute the network configuration once. However, the methods' process time behaviors do indicate if the problem is viable or even possible for large problems. Method 1 may have a processing time issue if the CM size becomes to large. However, measures can be taken to reduce the CM. For instance, one can select CPs that are good candidates (covering much of the transceiver's potential), or only having DP's from the area that actually need the coverage. Method 2 may be a good alternative to the conventional method 1 if process time is important, and at least in problems with similar characteristics as the project test: low transceiver range and "plenty" of room to put them.

The two methods developed in this project are still not established, and do have potential for future applications. The author is convinced that a skilled programmer will be able to rewrite the program to be more effective and possibly reveal the full potential of these methods.

6 Further Research

In this section, some thoughts are presented regarding further research. The first two sections present ideas that will improve the methods, and especially Methods 2 and 3. These two suggestions are also considered achievable. The third section on the other hand, is more of an intellectual experiment of how the methods used in this project would perform on an LSCP if they were modified as such.

6.1 General Program Development

Although all three methods have provided interesting results, they likely hold more potential than what has been accessed. Method 2 and 3 were mainly developed during this project. A skilled programmer will likely be able to improve the program to perform faster, and therefore reduce process time. Although Method 3 provided the best coverage with current programs, it was only marginally better than Method 1. If all programs were optimized, this result might change, and the process time will likely be significantly reduced.

6.2 Weights and Constraints

In this thesis the DZ has been treated as a homogeneous valued area, where no subareas have been given a different value. However, in real-world problems some areas are more important to cover than others, so why not also prioritize them in the optimization problem. By adding weights in the vector c_y one can easily manipulate certain points to be more valued. One way of doing this is by connecting a relevant quantity of that area to c_y, for instance, the historical ROV activity in that area, or the areas proximity to infrastructure. For the Successive Deployment, this is not as easily done as with ILP. However, it should be achievable by assigning weight to the cells in the DEM. This needs more thoroughly explanation. In the successive deployment procedure, the covered DZ is initially a array of zeroes. Each 0 represent $1m^2$ of the DZ. When the viewshed from a transceiver is added, all the points that are seen from the transceiver will be assigned the value 1. To calculate the coverage percentage (which has been the objective value in for method 2), all the values of the array are summed (in practice all the ones), and divided by the number of cells in the array. Now, what could be done is to created a "weight

array" with the same dimension as the array of zeros. The default of this array is that all the cells are 1. If some areas should be weighed different from others, assign this area another value. Higher than 1 if important, lower than 1 (for instance 0) if not important. Now instead of applying 1's to the seen cells of the covered DZ array, the corresponding value of the weight array is applied. That way when calculating the covered state, some cells will contribute more to the objective function. The objective value may be changed to be the sum of the values in the covered DZ array.

In this project there were no constraints on where the transceivers could be put, nor was there any cost to deploying them. For Method 1, it is also easy to associate a cost with the deployment of transceivers by assigning a negative value (the cost) in c_x. For the successive deployment it is natural to do so, as it is ordered to place out a fixed number of transceivers. If it was necessary, a if-test could be applied, with the logic:

```
if covered area > 40%:
   stop.
```

Stating that "we have reached our coverage goal, and do not want to deploy more transceivers." If there are areas that are not adequate for transceivers, measures to prohibit or make the candidate unfavorable may be applied. For Method 1, the CP's of such positions may simply be removed. Or cost of deploying transceivers in such positions may be applied. This would be applied the same way as explained for weights in all methods, with a negative sign indicating it to be a negative weight.

6.3 LSCP

In this thesis, the problem has been solved as an MCLP, with a maximum number of transceivers (35 in the project problem). How would these three methods perform if they were used to solve a similar setup as an LSCP, where the whole area should be covered? All the programs would have to be modified to become an LSCP. The methods that solve the problem as combinatorial problem (ILP) are expected to perform this challenge more elegantly than the successive deployments. However, 100 % DZ coverage will not be easy with current DP and CP selection. In figure 28, 100 transceivers are configured with ILP. The covered DP percentage is 97.8% while it covers 90.18% of the DZ. By the time all DPs are covered, there will be parts of DZ left uncovered, but the ILP will acknowledge this as successful, given it is optimizing the DP coverage. This mismatch needs to be

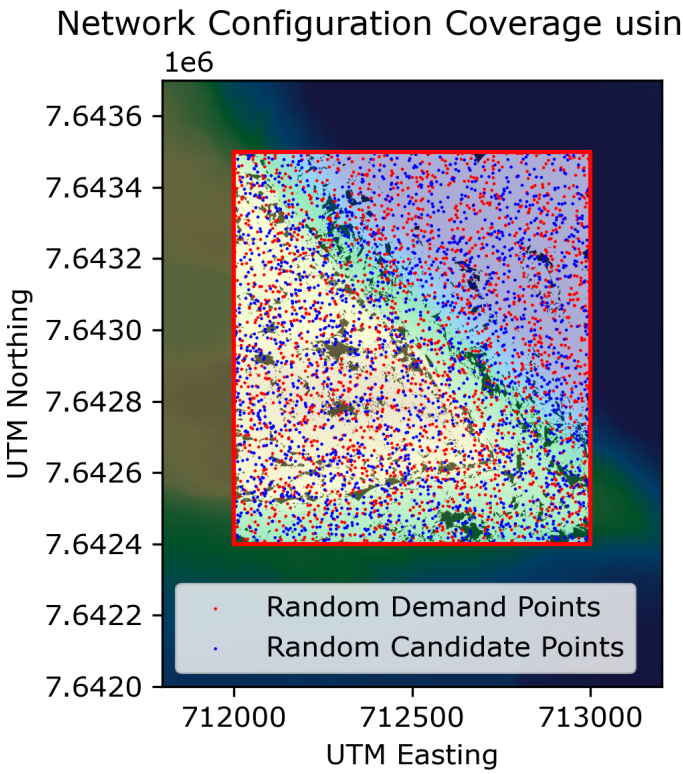

Figure 28: 100 deployed transceivers using Method 1. Covered DP percentage 97.8, total coverage: 90.18.

given a careful thought. There is more concern regarding Method 2. Since Method 2 successively deploys transceivers and does not consider the combinatorial problem, this method will likely end up with several smaller uncovered gaps between the transceivers. If this is the case, Method 2 will need to deploy a significant number of transceivers to cover up these gaps. Method 2 will be able to finish the problem, but possibly with a disproportionately large number of transceivers. Method 3 on the other hand is believed to be able to solve the LSCP problem better, as it has Method 1's ILP solver on the iteration steps.

References

[1] R. Church and A. Murray, *Location Covering Models: History, Applications and Advancements*. Jan. 2018, ISBN: 978-3-319-99845-9. DOI: 10.1007/978-3-319-99846-6.

[2] Y. Lou and N. Ahmed, *Underwater Communications and Networks* (Textbooks in Telecommunication Engineering). Cham: Springer International Publishing, 2022, ISBN: 978-3-030-86648-8 978-3-030-86649-5. DOI: 10.1007/978-3-030-86649-5. [Online]. Available: https://link.springer.com/10.1007/978-3-030-86649-5 (visited on 04/26/2024).

[3] A. Bahr, F. Schill, and I. Martin, "Practical applications of free-space optical underwater communication," *2021 Fifth Underwater Communications and Networking Conference (UComms)*, pp. 1–5, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:244273106.

[4] S. Bao, N. Xiao, Z. Lai, H. Zhang, and C. Kim, "Optimizing watchtower locations for forest fire monitoring using location models," *Fire Safety Journal*, vol. 71, pp. 100–109, 2015, ISSN: 0379-7112. DOI: https://doi.org/10.1016/j.firesaf.2014.11.016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0379711214001672.

[5] F. Welscher, R. Bulbul, J. Scholz, and P. Lederer, "The antenna coverage location problem in the context of cattle tracking in the austrian alps," *International Journal of Applied Earth Observation and Geoinformation*, vol. 122, p. 103 414, 2023, ISSN: 1569-8432. DOI: https://doi.org/10.1016/j.jag.2023.103414. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569843223002388.

[6] Hydromea, "Can ROVs go wireless?," Jun. 2023. [Online]. Available: https://www.oceannews.com/featured-stories/can-rovs-go-wireless (visited on 04/26/2024).

[7] N. M. A. NMA. "Bestill høyoppløselige dybdedata." (2024), [Online]. Available: https://kartverket.no/api-og-data/bestille-dybdedata (visited on 04/26/2024).

[8] G. Laporte, S. Nickel, and F. S. da Gama, *Location Science*, 2nd ed. Springer Cham, Mar. 16, 2020, 767 pp., https://doi.org/10.1007/978-3-030-32177-2, ISBN: 978-3-030-32177-2.

[9] R. L. Church and C. S. Revelle, "The maximal covering location problem," *Papers of the Regional Science Association*, vol. 32, pp. 101–118, 1974. [Online]. Available: https://api.semanticscholar.org/CorpusID:154435809.

[10] J. Nocedal and S. J. Wright, *Numerical Optimization* (Springer Series in Operations Research and Financial Engineering). Springer New York, 2006, ISBN: 978-0-387-30303-1. DOI: 10.1007/978-0-387-40065-5. [Online]. Available: http://link.springer.com/10.1007/978-0-387-40065-5 (visited on 04/16/2024).

[11] "Local vs. global optima," Mathworks. (), [Online]. Available: https://se.mathworks.com/help/optim/ug/local-vs-global-optima.html (visited on 04/23/2024).

[12] *Integer programming*, in *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/Integer_programming.

[13] Gurobi. "Mixed integer programming (MIP) - a primer on the basics." (), [Online]. Available: https://www.gurobi.com/resources/mixed-integer-programming-mip-a-primer-on-the-basics/ (visited on 04/16/2024).

[14] B. C. Williams, "Integer programming and branch and bound," Nov. 15, 2004. [Online]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://web.mit.edu/16.410/www/lectures_fall04/L18-19-IP-BB.pdf.

[15] S. Singer and J. Nelder, "Nelder-mead algorithm," *Scholarpedia*, vol. 4, no. 7, p. 2928, 2009. DOI: 10.4249/scholarpedia.2928.

[16] D. M. Olsson and L. S. Nelson, "The nelder-mead simplex procedure for function minimization," *Technometrics*, vol. 17, no. 1, pp. 45–51, 1975, Publisher: Taylor & Francis _eprint: https://www.tandfonline.com/doi/pdf/10.1080/00401706.1975.10489269. DOI: 10.1080/00401706.1975.10489269. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00401706.1975.10489269.

[17] N. M. A. NMA. "Høydedata." (2024), [Online]. Available: https://hoydedata.no/LaserInnsyn2/ (visited on 01/18/2024).

[18] GDAL. "GTiff – GeoTIFF file format." (2024), [Online]. Available: https://gdal.org/drivers/raster/gtiff.html (visited on 02/13/2024).

[19] OGC. "OGC GeoTIFF standard." (2024), [Online]. Available: https://www.ogc.org/standard/geotiff/ (visited on 02/13/2024).

[20] L. Mæhlum and J. K. Rød, *UTM*, in *Store Norske Leksikon*, Accessed: 3. april 2024, 2005. [Online]. Available: https://snl.no/UTM.

[21] V. Marianov and C. ReVelle, "Siting emergency services," in *Facility Location*, Z. Drezner, Ed., New York, NY: Springer New York, 1995, pp. 199–223, ISBN: 978-1-4612-5357-0 978-1-4612-5355-6. DOI: 10.1007/978-1-4612-5355-6_11. [Online]. Available: http://link.springer.com/10.1007/978-1-4612-5355-6_11 (visited on 04/15/2024).

[22] M. F. Goodchild and J. Lee, "Coverage problems and visibility regions on topographic surfaces," *Annals of Operations Research*, vol. 18, no. 1, pp. 175–186, Dec. 1989, ISSN: 0254-5330, 1572-9338. DOI: 10.1007/BF02097802. [Online]. Available: http://link.springer.com/10.1007/BF02097802 (visited on 02/14/2024).

[23] Y.-H. Kim, S. Rana, and S. Wise, "Exploring multiple viewshed analysis using terrain features and optimisation techniques," *Computers & Geosciences*, vol. 30, no. 9, pp. 1019–1032, 2004, ISSN: 0098-3004. DOI: https://doi.org/10.1016/j.cageo.2004.07.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098300404001372.

[24] W. R. Franklin and C. Ray, "Higher isn't necessarily better: Visibility algorithms and experiments," in *Advances in GIS research: sixth international symposium on spatial data handling*, vol. 2, Citeseer, 1994, pp. 751–770.

[25] lp_solve. "Introduction to lp_solve 5.5.2.11." (Dec. 2, 2024), [Online]. Available: https://lpsolve.sourceforge.net/5.5/.

[26] J. R. Aasvold, "Topografi i optimalisering," University of Tromsø, 2023.

[27] "Huawei matebook." (), [Online]. Available: https://consumer.huawei.com/no/laptops/matebook-13/specs/.

[28] "Intel® core™ i7-8565u processor." (), [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/149091/intel-core-i7-8565u-processor-8m-cache-up-to-4-60-ghz.html (visited on 04/18/2024).

[29] "What is NumPy?" https://numpy.org/. (), [Online]. Available: https://numpy.org/doc/stable/user/whatisnumpy.html.

[30] "Matplotlib," Matplotlib: Visualization with Python. (), [Online]. Available: https://matplotlib.org/.

[31] "Shapely." (), [Online]. Available: https://shapely.readthedocs.io/en/stable/index.html (visited on 04/26/2024).

[32] "GDAL documentation," GDAL. (), [Online]. Available: https://gdal.org/index.html.

[33] "Rasterio introduction," Rasterio. (), [Online]. Available: https://rasterio.readthedocs.io/en/stable/intro.html.

[34] S. Mitchell, A. Kean, A. Mason, M. O'Sullivan, A. Phillips, and F. Peschiera. "PuLP," PuLP. (), [Online]. Available: https://coin-or.github.io/pulp/ (visited on 05/10/2024).

[35] "Scipy," Scipy. (), [Online]. Available: https://scipy.org/.

[36] OpenAI. "ChatGPT," ChatGPT. (), [Online]. Available: https://chat.openai.com/.

List of Acronyms

ACLP Antenna Coverage Location Problem.

AUV Autunom Underwater Vehicle.

CM Coverage Matrix.

CP Candidate Point.

CZ Candidate Zone.

DEM Digital Elevation Model.

DP Demand Point.

DZ Demand Zone.

GDAL Geospatial Data Abstraction Library.

GeoTIFF Geographic Tagged Image File Format.

GIS Geographic Information Systems.

ILP Integer Linear Programming.

LP Linear Programming.

LSCP Location Set Covering Problem.

MCLP Maximal Covering Location Problem.

MI Magnetic Induction.

MILP Minimum Impact Location Problem.

MILP Mixed Integer Linear Programming.

N.C.C Network Configuration Coverage.

NMA Norwegian Mapping Autority.

ROV Remotely Operated Vehicle.

UWC Underwater Wireless Communication.