FACULTY OF GEO-INFORMATION SCIENCE
AND EARTH OBSERVATION
ITC

SPATIAL OPTIMIZATION FOR WIND FARM ALLOCATION

Dragana Subotic

SUPERVISORS:

Prof.dr. Alexey A. Voinov
Dr. Nicolas Prechtel

UNIVERSITY OF TWENTE.

FACULTY OF GEO-INFORMATION SCIENCE AND EARTH OBSERVATION

ITC

SPATIAL OPTIMIZATION FOR WIND FARM ALLOCATION

DRAGANA SUBOTIC

Enschede, The Netherlands, September, 2017

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.

THESIS ASSESSMENT BOARD:

Prof.dr. Menno-Jan Kraak (Chair)
Prof.dr. Alexey A. Voinov (1st Supervisor)
Dr. Nicolas Prechtel (2nd Supervisor)

UNIVERSITY OF TWENTE.

ITC

Acknowledgments

I am especially fortunate to have Professor Alexey Voinov as my supervisor, who has showed unreserved support, guidance and patience in this research project. I am very grateful that you allowed me to explore my research interests, at the same time providing much needed instructions, criticism and depth of knowledge.

I would also like to extend my thanks to everyone from TU Munich, TU Vienna, TU Dresden and ITC, involved in the Cartography M.Sc. program, who made this amazing study experience possible for me.

Finally, I owe my deepest appreciation to my parents and my sister, for always being there for me, unconditionally.

Abstract

In order to secure a sustainable development and minimize the dependence on fossil fuels, energy production has to turn to alternative energy resources like wind, solar or bio energy. Renewable energy resources are not only naturally replenished, but also do not have the negative impact of environmental degradation and pollution, CO2 emission and so on. Wind energy has shown to be a serious competitor on the renewable energy market. According to the European wind agency, wind energy meets already 11% of the EU's total power demand, with €43bn investments made in this sector only in 2016.

Wind farm development and choosing suitable locations is a complex and iterative process that requires analysis of a number of criteria. Multi-criteria Decision Analysis (MCDA) methods, in conjunction with GIS (Geoinformation Systems) are used more and more as an effective support tool in renewable energy planning.

Following a methodology that first investigates different aspects of spatial optimization and wind farm development, this research tries to address the problem of assessing the potential of an area for wind energy production, by developing an algorithm that facilitates the search for an optimal scheme for wind farms. The algorithm considers shape and size of the available area, different turbine types and the wind resource. The results obtained in this process describe the applicability of the developed approach.

Keywords: spatial optimization, wind farm layout design, multi-criteria analysis, binary integer linear programming

List of figures

List of tables

List of abbreviations

AEP	Annual Energy Production
CoE	Cost of Energy
GIS	Geographic Information System
LCoE	Levelized Cost of Energy
LP	Linear Programming
MADA	Multi-Attribute Decision Analysis
MCA	Multi-Criteria Analysis
MCDM	Multi-Criteria Decision Making
MILP	Mixed Integer Linear Programming
MODA	Multi-Objective Decision Analysis
NSGA	Non-dominated Sorting Genetic Algorithm
PSO	Particle Swarm Optimization
SA	Simulated Annealing
WFL	Wind Farm Layout
WFLO	Wind Farm Layout Optimization

Table of Contents

1. Introduction

1.1. Background

An important part of the EU strategy for a sustainable future is the "20/20/20" clause, that aims at reducing the carbon dioxide emission by at least 20% (compared to the levels from 1990), an increase in the energy efficiency by 20% and increasing the total share of renewable energy sources in the energy consumption up to 20% (European Commission, 2010). In order to achieve this goal, the extensive use of fossil fuels in energy production must be substituted with alternative energy sources with low CO_2 emissions. Renewable energy sources, like wind or solar, are naturally replenished and have a far lower negative impact on the environment. More and more countries in Europe, and around the world, are starting to recognize the advantages of clean energy production and are working towards transitioning in this direction.

Wind energy has become very popular in recent years, and proved to be a serious competitor on the renewable energy market, with significant increases in installed capacity every year (Gigović *et al.*, 2017). For example, at the end of 2016 Netherlands had a total of 4.3 GW wind capacity installed, and is planning to reach 10 GW by the end of 2020 (Wind EUROPE, 2016). According to the European wind agency, wind energy meets already 11% of the EU's total power demand, with €43bn investments made in this sector only in 2016.

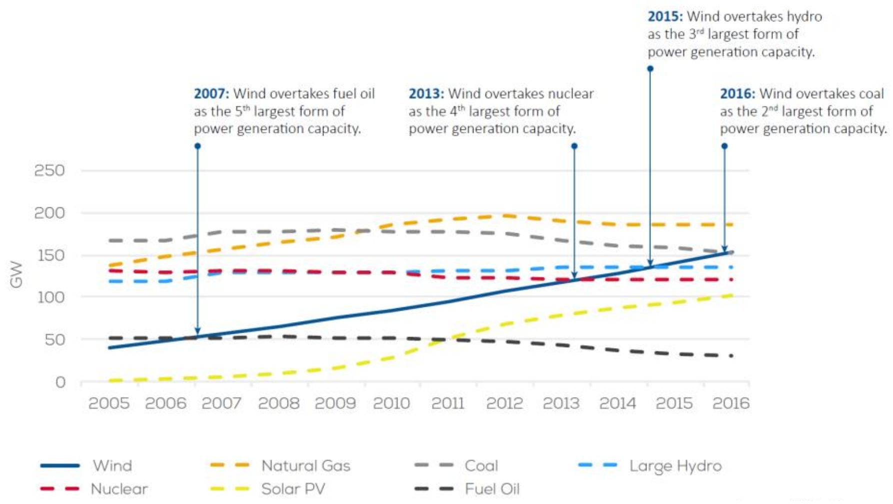


Figure 1-1: Types of energy production in Europe (Wind EUROPE, 2016)

1.2. Research problem

When it comes to building wind farms, especially in smaller countries with densely populated areas, choosing suitable locations is a complex and iterative process that requires analysis of a number of criteria (Grassi, Chokani and Abhari, 2012; Gigović *et al.*, 2017). Although the suitability depends foremost on the wind speed of an area, a number of other factors also must to be taken into consideration (environmental, cultural and visual impact, land use, land cover type and topography to name a few).

Managing sustainable energy is starting to rely more on the use of Multi-criteria Decision Analysis (MCDA) methods; that in conjunction with GIS (Geoinformation Systems) makes for an effective support tool in renewable energy planning (Pohekar and Ramachandran, 2004; Díaz Ignacio, 2016). Multi-criteria decision methods are generally categorized into two groups: multi-attribute decision analysis (MADA) and multi-objective decision analysis (MODA) (Malczewski and Rinner, 2015).

Multi-attribute decision analysis is used for identifying suitable areas for wind farms on a macro level, defining different criteria of suitability by which an area is then being rated.

Although less researched, the field of multi-objective optimization has shown to be a very promising tool in wind farm planning and design. Simply said, the optimization of a wind farm layout can be described as the process of finding the optimal position of a wind turbine that maximizes the value of an objective function (Tesauro *et al.*, 2012). Given a predefined area, the goal of the Wind Farm Layout Optimization (WFLO) is to determine where to locate wind turbines in order to maximize the output under certain conditions. Although a very important factor, the environmental impact of wind farms has not been addressed sufficiently in past studies and it has been treated as a minor criteria. But since more land is going to be exploited for future wind farm projects, wind farms are more likely to be located in proximity of residential areas. The impact of wind turbines on the landscape is a common objection for wind farm development. It is becoming a big concern for the wind farm owners, since many studies showed that the visual intrusion of wind turbines together with the noise that they produce and the shadow flickering effect are the biggest causes of skepticism in the local level towards building wind farms. Another concern is also the risk for bird populations who get killed by the spinning blades of wind turbines. This means that in the process of allocating a wind farm, these issues will also have to be addressed, in order to ensure maximum operation with a minimal negative impact to the surrounding area and its residents (Kwong *et al.*, 2012; Zhang, 2013; Mittal, Kulkarni and Mitra, 2016).

1.3. Research objectives

Number of studies that address the evaluation of locational suitability, or the optimization of sites for wind power development is still relatively low. The main goal of this research is to develop a method that will facilitate the search for the best possible layout scheme for wind farms in a defined region, and on a more general level, to investigate the possible solutions for applying Multi-Criterial Decision Making methods in conjunction with GIS in spatial optimization of wind farms. Specific objectives that require completion are as follows:

- Identify the main factors that influence wind farm development
- Develop an understanding of the spatial optimization methods and their main aspects
- Define the mathematical problem of placing wind turbines in a region of interest
- Implement an optimization algorithm for determining the most appropriate wind farm layout, at the same time assessing the cost and energy output, and considering different wind turbine types

In order to meet the above-mentioned objectives, the following research questions should be answered:

- How to formulate an objective function in order to maximize the AEP (Annual Energy Production)?
- Which multi-objective optimization techniques are best for a wind farm layout scenario?
- Should this model account for turbine wakes, and how can this effect be incorporated into the model?
- Which optimization algorithm will provide the best trade-off between accuracy of the output result and computing effort?
- How to assess the quality of the model?

2. Theoretical background

In this chapter, the basic terms related to optimization are explained, as well as the definition and context of spatial optimization. Methods for solving spatial optimization are presented, relating to the structure of a spatial optimization problem. This chapter also provides a review of the criteria and factors influencing wind farm development, and presents a discussion about the previous work related to this field.

2.1. What is optimization

Optimization is described in literature as a sub-field of operational research (OR), that applies techniques from mathematics and computer science to find the best decision in a set of alternatives, regarding some criteria (Scholz, 2016). According to Malczewski and Rinner (2015), multi-objective optimization belongs to the area of Multi-criteria Decision Making (or Multi-criteria Decision Analysis, MCDA) and is applied in many fields of science and engineering.

An optimization problem is defined using three elements:

- One (single-objective optimization) or a number of objective functions (multi-objective optimization) that define the goals of the optimization
- Constraints that define the necessary conditions to be satisfied, in order to make the solution acceptable for the problem at hand (Coello, Lamont and Van Veldhuizen, 2007; Malczewski and Rinner, 2015)
- Decision variables, that are the numerical quantities, for which values are to be chosen as a result of the optimization (Coello, Lamont and Van Veldhuizen, 2007)

The mathematical expression of this would be:

$$Maximize\ F(x) = \{f_1(x), f_2(x), \dots f_n(x)\}$$

$$subject\ to\ g_i(x)\ \geq 0,\ \ h_i(x) = 0$$

$$x \in X$$

where $F(x)$ is the n-dimensional objective function, $h_i(x)$ and $g_i(x)$ a set of equality and inequality constraints, X is the set of feasible alternatives and $x = (x_1, x_2, \dots, x_m)$ is the vector of decision variables $x_i \geq 0$, for $i = 1, 2, \ldots, m$ (Church, 2002; Tong and Murray, 2012; Zhang, 2013; Malczewski and Rinner, 2015).

2.2.1. Spatial optimization

In a spatial context, optimization plays a very important role in geography and GIS, used as a basic tool for land use planning, location and allocation models, transportation and vehicle routing, medical geography and many others (Tong and Murray, 2012).

What distinguishes a spatial optimization problem from any other type, is that the parameters, constraints and decision variables used in this case have geographic characteristics, that make them spatially interdependent. The decision to be made is usually in the context of where should something be placed or located. This allows for the solutions to be easily presented on maps, but it also means that spatial optimization models are specific and more often than not difficult to appropriately structure and solve (Tong and Murray, 2012; Lei, Church and Lei, 2015; Malczewski and Rinner, 2015).

It is important to explain the difference between spatial objective problems (or models) and the methods (or algorithms) for solving these problems, as well as their classifications. The objective functions determine whether the model is linear or nonlinear, and depending on the type of the decision variables, the problem at hand can be modelled as a discrete or as a continuous one (Coello, Lamont and Van Veldhuizen, 2007; Malczewski and Rinner, 2015). A discrete variable can take only a finite number of values, while continuous variables can take on any real value in a specified interval (Gropp and Moré, 1997). If all of its decision variables are discrete, the problem can be called an integer optimization problem, otherwise it is classified as an continuous optimization, unless there are both type of variables involved in which case we are calling it a mixed type optimization (Malczewski and Rinner, 2015).

2.2.1.1. Spatial optimization and GIS

Geographic Information Systems (GIS) can be described as information systems that possess a wide range of functions and toolboxes for spatial data collection, manipulation and visualization. In the context of spatial optimization problems and allocation models, the use of GIS tools is indispensable. This includes collection, processing and analysis of spatial data sets, assistance in identifying candidate sites, as well as generating graphical outputs and appropriate visualization of the solution (Church, 2002; Tong and Murray, 2012; Lei, Church and Lei, 2015).

2.2.2. How to solve an optimization problem

All methods for solving optimization problems fall into one of the two categories: exact (deterministic) methods and approximate (stochastic) methods (Coello, Lamont and Van Veldhuizen, 2007; Tong and Murray, 2012; Malczewski and Rinner, 2015).

Exact methods go through all possible scenarios for the problem solution, ensuring that the final output is the best and optimal result, and that no other values of the decision variables would result in a better objective. Enumeration and linear programming are commonly used types in this case. However, because of their complexity and intensive computation, spatial optimization problems are difficult to solve by exact methods (Tong and Murray, 2012; Malczewski and Rinner, 2015).

On the other hand, different heuristic methods have been developed, that present the approximate solution strategy to an optimization problem. Heuristic algorithms function as a trial and error search, which computes near optimal solutions but at the same time decreases computation time. The candidate solution is improved in a number of iterations, defined by the user, until a certain quality of the solution is reached. Metaheuristics (or advanced heuristics) are a group of methods based on natural processes, they are easy to implement and do not require continuous objective functions (Herbert-Acero *et al.*, 2014). A wide range of different metaheuristics have been described in literature and used for solving spatial optimization problems:

Genetic (evolutionary) algorithms are based on the principles of natural selection and survival of the fittest. These algorithms are population based, which means they operate on the whole set of individuals. In biology and evolution process, the genetic material of individuals is stored in chromosomes and combined in creating offspring, which means it changes from generation to generation. When applying genetic algorithms, a population represents an initial set of randomly generated solutions, a chromosome is one individual solution, and the genes (e.g. variables) are the characteristics of a single solution. Each iteration (generation) is formed by combining solutions from the previous one, that had the best characteristics (level of fitness), and this is done by using the selection, mutation and crossover operators. When a pre-defined condition (iteration limit) is reached, the process terminates (Malczewski and Rinner, 2015; Sunak *et al.*, 2015). One of the most popular genetic algorithms is the Non-dominated Sorting Genetic Algorithm II (NSGA-II), which produces finds the global optimum with high accuracy but requires high computing effort (Sunak *et al.*, 2015).

Particle Swarm Optimization (PSO) is another meta-heuristic method that applies population-based search. Swarm intelligence is inspired by swarm behavior, exhibited in nature by swarms of insects (e.g. bees, ants), flocks of birds etc. (Cortez, 2014; Malczewski and Rinner, 2015). Similar to genetic algorithms, PSO starts with a set of random solutions, where each particle has its own velocity and position in the search space (Zhang, 2013). In a cycle of iterations, a particle changes its position according its velocity, but also depending on the best position found by its neighboring particles and its own best position. The particles are linked so that one can inform the other about its memory of movement.

In contrast with the population search methods, local search algorithms focus on local neighborhoods of an initial solution. E.g. Simulated Annealing (SA), inspired by the phenomenon in metallurgy when a material is heated and slowly cooled, recreates the process of rearranging atoms from a disordered state and high energy to a solid state and low energy (Cortez, 2014; Malczewski and Rinner, 2015). SA iteratively improves the initial solution until certain convergence criteria are met, by using a control temperature parameter (T) that determines the probability of accepting inferior solutions (Cortez, 2014). Other commonly used types of local search algorithms are the tabu search method and hill climbing.

2.2. Wind Farm Development

As discussed in previous chapters, wind farm development is a complex process, that requires an iterative approach. However, two main steps can be distinguished here:

- **Site selection** - Identification of suitable land for building wind farms
- **Micrositing -** Determining the layout of the wind turbines, as well as their number and size

Wind farms are very versatile, which means they can be integrated into different types of landscapes (Herbert-Acero *et al.*, 2014). However, choosing suitable locations for wind farms is time-consuming, because of a number of factors that need to be considered at this stage. Wind potential of an area is of course the critical factor, but there are other criteria that have to be taken into account. As showed in Figure 2.1, these include distance from roads and urban areas, distance from the electricity grid, slope of terrain, land cover, distance to places of interest and natural reserves, forest areas and last but not less important social acceptability (Sunak *et al.*, 2015).

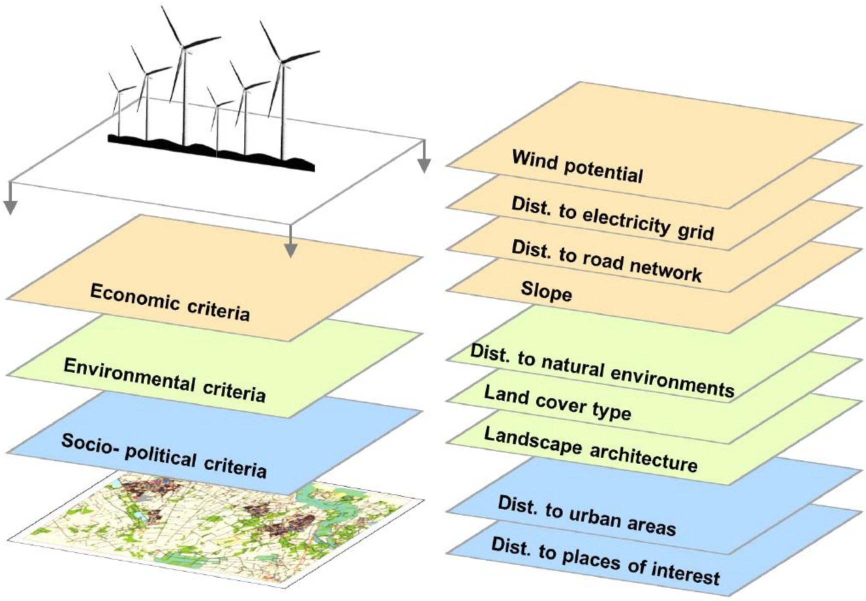


Figure 2-1: Wind farm development criteria (Sunak *et al.*, 2015)

Wind farm design and layout modelling refers to the process of determining the size of a wind farm and the arrangement and number of individual wind turbines. The size depends on the available area, but also on the expected energy output (Herbert-Acero *et al.*, 2014). Energy production of a wind turbine is a function of the wind speed at rotor height (Zhang, 2013). Proper turbine placement within a suitable area is of great importance, because the smallest change in wind speed can affect the energy output a great deal.

2.2.1. Wind turbines

Wind conditions of an area can also determine which wind turbines should be used. Technical specifications of turbines, that are also related to wind farm layout optimization are:

- Cut-in speed V_i
- Cut-out speed V_o
- Rated speed V_r
- Rated power P_r
- Power curve
- Rotor diameter D
- Hub height H

Cut-in speed is the wind speed needed for the turbine blades to start spinning, and generating power. If the wind speed is higher that the cut-out speed V_o, the turbine mechanism will shut down in order to prevent turbine damage. How much energy will the turbine produce with wind speeds between V_i and V_o depends on the power curve of the turbine. But if that speed is equal or higher than the rated speed V_r, the energy output will be constant and equal to the rated power P_r (Samorani, 2013).

2.2.2. Wind modelling and energy output

The wind phenomena can be represented with different regimes:

- Fixed wind direction and constant wind speed
- Variable wind direction and constant wind speed
- Constant wind direction and variable wind speed
- Variable wind direction and variable wind speed (Mittal, 2010).

When a wind farm is experiencing uniform wind speed, the power generated from the turbines can be calculated with this simplified expression:

$$P = \sum_{i=1}^{N} \frac{1}{3} u_i^3,$$

where u_i is the wind speed, and N is the number of wind turbines (Bã Nos *et al.*, 2010; Mittal, 2010). This is a generalized case, because the power output depends also on the power curve of the turbine used. Finally, the Annual Energy Production (AEP), which can be defined as integration of the total power (kW) produced over time (h), is calculated as

$$AEP = 8760 \sum_{i} \sum_{j} \sum_{k} F_{ijk} P_{ijk},$$

where 8760 is the number of hours in a year. F_{ijk} function presents the distribution of wind speeds and directions, and P_{ijk} is the corresponding power generated by that turbine (Gu and Ji, 2010; Zhang, 2013). Coefficients *i*, *j* and *k* represent the number of wind directions, wind speeds and number of wind turbines, respectively.

2.2.3. Cost modelling

Optimization of wind farm layout cannot rely only on energy production as main criteria, because the financial balance is always an important aspect to make a project feasible. The cost parameters mainly depend on the installation costs of the wind turbines, but there are also operation and maintenance costs and grid connection costs (IRENA (International Renewable Energy Agency), 2012; Tesauro *et al.*, 2012). According to Mosetti, Poloni and Diviacco, and many other authors that applied this approach later on, a very simple function can be used, that defines the cost of a wind farm as a function of the number of turbines:

$$\boldsymbol{Cost = CN\left(\frac{2}{3} + \frac{1}{3}e^{-0{,}00147N^2}\right)},$$

where N is the number of installed turbines, and C is the cost of one individual turbine. C can be defined in different ways, but can also be left out, in which case the cost would be a dimensionless parameter, depending only on the number of turbines. This expression assumes that the cost per turbine drops with the number of installed turbines.

2.2.4. Minimum distance between turbines

When wind passes through a turbine, it absorbs its energy, and its speed decreases in the area around the turbine rotor, resulting in production losses between neighboring turbines (Fagerfjäll, 2010). This effect is called the wake of the turbine. The wake is the reason why turbines cannot just be placed close as possible, but depend on the needed recovery of wind energy behind the neighboring turbine (Mustakerov and Borissova, 2010; Borissova and Mustakerov, 2017). Depending on the type of wind that is taken into account, uniform or predominant, different distance constraints can be defined that are usually function of the rotor diameter of the turbine (Yamani *et al.*, 2016).

2.3. Previous work

Over the years, different approaches were used for MCA (Multi-criteria Analysis) and decision making in wind farm development.

For analysis of land suitability, most studies are focusing on the AHP (Analytical Hierarchy Process) approach, that requires the hierarchy of criteria to be defined by using weights (Szurek, Blachowski and Nowacka, 2014; Latinopoulos and Kechagia, 2015; Höfer *et al.*, 2016; Villacreses *et al.*, 2017).

In order to address the NIMBY (Not In My Back Yard) syndrome, which refers to the negative attitude on a local level towards wind farm building, some authors focused on creating applications that would help in increasing the social acceptance for wind farm projects. The work of Grassi and Klein (2016) describes the advantages of a 3D interactive visualization platform in wind farm planning, or the case study by Slikker (2016), presents a web-application based on MCA, that allows users to manually set different values for suitability criteria and observe the results.

The WFLO (Wind Farm Layout Optimization) problem has been researched in different scenarios, is with different objectives. A literature review showed that these approaches differ in terms of the optimization method used, but also in modeling of wind conditions, type of search space representation, and so on. Some of the used criteria for the objective functions are first of all the energy production, AEP (Annual Energy Production) and the cost of energy (CoE, cost per KWh of energy produced, that also takes into account the costs of installation and maintenance), but also profit and net present value (Tesauro *et al.*, 2012; Herbert-Acero *et al.*, 2014).

The work by Mosetti, Poloni and Diviacco dating from 1994 is one of the first studies that applied optimization to the wind farm layout problem. The parameters and criteria that were set here, served as a guide for many of the studies done after. The location site was modelled as 10 x 10 square grid, and the authors considered two objectives: maximization of energy output and minimization of investment costs. Every cell of the grid is sized to 5D (5 rotor diameters), and the center of every cell is a possible location for a wind turbine. The test problem was solved with a genetic algorithm, for three wind speed scenarios: uniform wind speed, equally distributed (36 directions) with a 12 m/s wind speed, and non-uniformly distributed winds of 8, 12 and 17 m/s.

Sunak *et al.* (2015) made a GIS-based decision support system for the optimal sitting of wind farm projects. This project was performed in two steps: First a selection of best suitable sites by means of a spatial Analytic Hierarchy Process (AHP) modelling approach, and then a micrositing for optimal wind farm layouts simulation. Multi-criteria analysis was used to define exclusion areas and rated areas for wind farm development for the study area of city of Aachen, Germany, which was the first step in defining the optimization model. The authors used Levelized Cost of Energy (LCOE) as the optimization criteria, which presents the minimum cost of energy below which the total discounted revenues would be less than the total discounted investments. This formula takes into consideration investment expenditures, operation and maintenance costs, energy production and lifetime of the system.. Implementation of the optimization process is done with a gradient-based algorithm in Matlab, using the existing fmincon function.

McWilliam, van Kooten and Crawford (2012) developed an original approach to wind siting at large scale; that could serve more as a preliminary screening tool, than a model for wind farm layout optimization. This study incorporates the multiple factors that influence site suitability into an overall economic cost function. The model predicts the optimal wind farm radius and the spacing between installed turbines, and considers the electrical transmission network as well as the Annual Energy Production (AEP) of the installed turbines. Optimization process is implemented using a gradient type algorithm, Sequential Quadratic Programming (SQP), over a Cost of Electricity (CoE) function.

Chowdhury *et al. (*2010) presented a new method for unrestricted placement of turbines, with different rotor diameters, for maximum power generation. The wind farm dimensions and the minimum distance between turbines are treated as system constraints. The wind farm layout was treated as a continuous model, for three different cases: wind farm with identical turbines, wind farm with non-identical turbines and wind farm with identical turbines that can adapt to wind conditions. Each of these went through an optimization process using the Particle Swarm Optimization (PSO) algorithm.

Yamani *et al. (*2016) created a wind farm optimization model that represents a trade-off between energy generation and noise production, but also accounts for land-use and proximity constraints. This study investigates how can land use be incorporated as a constraint in the model itself, and what effect this has on the output result. The examination of this regulatory constraint is done by calculating the area of all triangles that the location of the turbine defines with vertices of a polygon under question. This study applied the Non-Dominated Sorting Genetic Algorithm (NSGA-II), which is a fast elitist algorithm, for a 3 km x 3 km square area.

Mittal, Kulkarni and Mitra (2016) proposed a hybrid methodology for wind farm micro-siting. The energy-noise trade-off is modelled with a two-step optimization – NSGA-II algorithm for determining the number of turbines and their layout (a discrete formulation), and in the second phase a gradient search based method to improve the results. The layout is divided into a finite number of grid points, and the optimal points found in the first stage are used as an input for the second stage, where their locations are improved.

Although most WFLO studies are focusing on modelling the influence of wind speed and direction change on the turbine energy production, and modelling the inter-turbine interaction, the approach being developed here aims at assessing the potential of an area for placing turbines, depending on their size and type, and taking into consideration zones where turbines cannot be placed, depending on land suitability criteria.

3. Methodology

3.1. Methodology flow

The work on this project is carried out in several phases, as showed in Figure 3-1. Phase 1 includes an analysis of all the prerequisites needed to start to tackle the problem, together with a comprehensive literature review and overview of the modelling approaches in wind farm scenarios used up to date. Phase 2 includes acquisition and preparation of the relevant data input. In order to test the algorithm on the study area, it is needed to perform an analysis of the available land, based on different land uses and restriction criteria. During phase 4 formulation and implementation of the optimization algorithm is performed, together with a validation and sensitivity testing. In the last phase, the algorithm is applied to the study region, for which the data were prepared in phase 2 and 3.

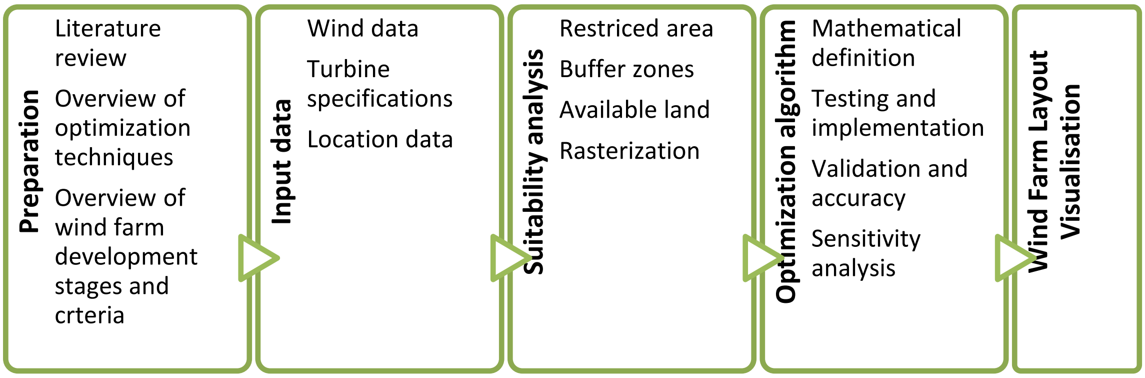


Figure 3-1: Methodological overview

3.2. Input data preparation

Different data types are needed to set up the optimization model. How was this data acquired and structured is described below.

3.2.1. Study area

Once the optimization procedure is established, its applicability can be tested on any given area. Enschede is located in the eastern part of the Netherlands, and it is the biggest municipality in the Overijssel province, covering an area of approximately 143 km^2. Unlike the western parts, this part of the country does not have such high annual wind speed averages, and falls in the category of 6.5 m/s – 7 m/s annual average wind speed (Senter

Novem, 2005). Considering that areas with 6 m/s and more can be considered for wind farm development, especially with constant technological improvements in turbine design that allow for even very low wind speeds, it is safe to say that a municipality like Enschede could have its first wind farm development projects in the near future.

In order to comprise the location data for the study area, Top10NL Cadastral Level Vector Shape files for 28 East, 29 West, 34 East and 35 West were used. These shapefile blocks were merged into a single layer, which was in the next step clipped to the boundary area of the municipality of Enschede. Different layer types used in this process are described in Table 1.

Land use class	Source	Sub-classes applied
Agricultural land	TOP10NL - terrein	typelandgebruik: ”akkerland”, “boomgard”, “boomkwekerij”, “fruitwekerij”, “populieren”
Built-up and residential area	TOP10NL - terrein	typelandgebruik: ”bebouwd gebiet”, “overig”, “dodenakker”
Forest	TOP10NL - terrein	typelandgebruik: ”bos: gemengd bos”, “bos:loofbos”, “bos: naaldbos”
Grassland and meadow	TOP10NL - terrein	typelandgebruik: ”braakligend”, “grasland”, “heide”
Railway	TOP10NL - terrein	typelandgebruik: ”spoorbaanlichaam”
Roads	TOP10NL – wegdeel	
Water	TOP10NL - waterdeel	
Natural reserve	Natura 2000	

Table 1: Location data sources

As seen in Figure 3-2, the study area demonstrates a densely populated space, with very mixed land use.

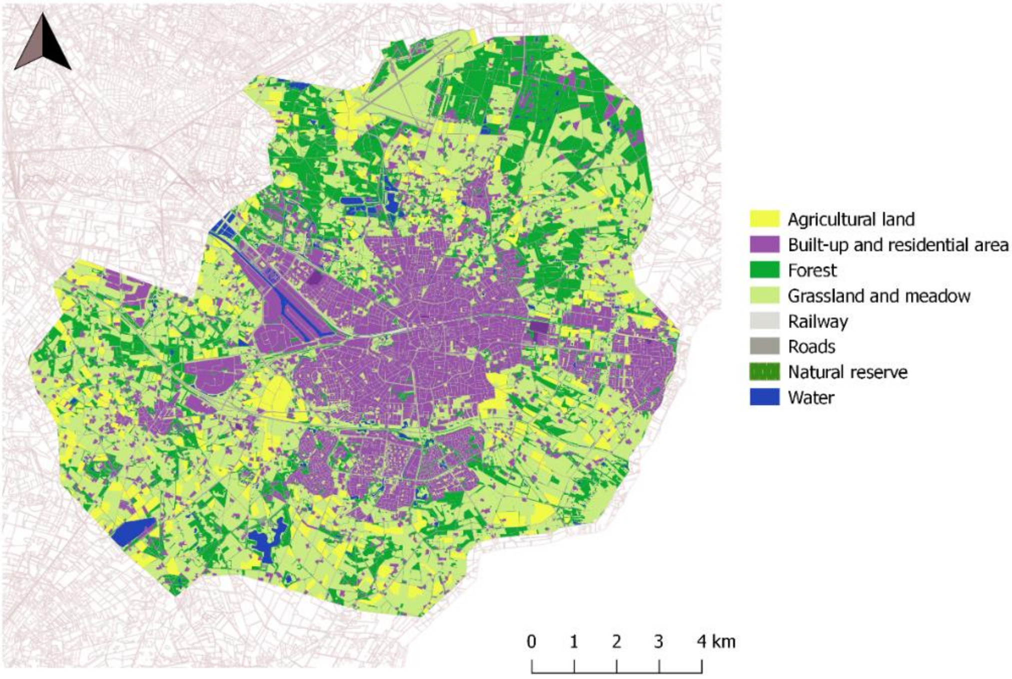


Figure 3-2: Land Use Map of Enschede

Because the goal here is to apply optimization techniques, the identification of available land is done by a simplified process of a multi-criterial analysis. The final output was created considering the criteria for restriction and buffer zones from Table 2, and only the left available space is taken into account as data input.

Zones	**Pixel value**
Urban and residential areas and buffer zones within 1 km from urban and residential areas	0
Rural residential areas and buffer zones within 0.5 km from rural residential areas	0
Railroads and buffer zones within 100 m from railroads	0
Water bodies and buffer zones within 50 m of water bodies	0
Natural reserve areas and buffer zones within 1 km from natural reserve areas	0
Agricultural areas	0
Forest areas	0
Roads	0
Others	1

Table 2: Exclusionary zones for land availability estimation *(*Azizi et al., 2014*)*

After using the available buffer, overlay and intersection functions in QGIS, the available area is discretized into a raster map of 250 m x 250 m grid cell size, with a binary pixel value.

This specific cell size ensures enough area for placement of the smallest turbine with its proximity constraint, in the center of every raster cell. The map below shows potential areas for turbine placement.

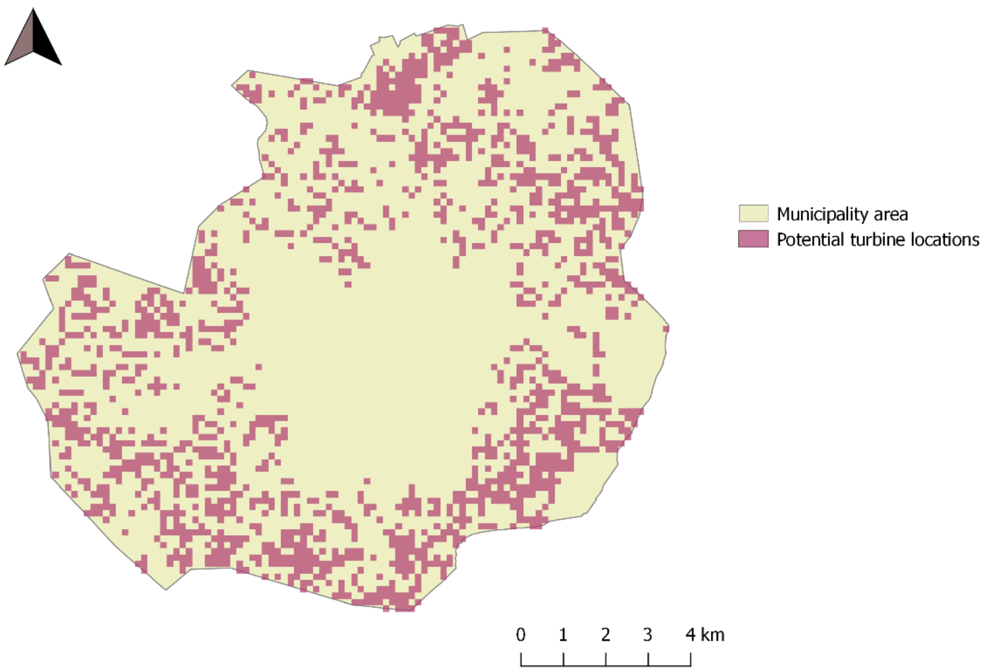


Figure 3-3: Raster grid map for potential turbine locations

3.2.2. Turbine specifications

Three different turbine types were taken as a reference, to describe the variability of characteristics that these machines have, and how it effects the production of energy (Enercon, 2017; PIERROT, 2017; Siemens, 2017; Vestas, 2017). What these parameters from Table 3 represent in practice, has already been explained in Chapter 2.

	Enercon	**Vestas**	**Siemens**
Vi [m/s]	4	4	3-5
Vo [m/s]	21.5	25	25
Vr [m/s]	11	12	12
Pr [kW]	800	2000	3150
H [m]	60	95	129
D [m]	53	90	142

Table 3: Turbine specifications

These specific turbine types are chosen because they can be considered very typical in onshore use. The Vestas V90-2.0 is the most commonly used onshore turbine worldwide, while the Siemens SWT-3.15-142 belongs to the group of large wind turbines and its design allows for production of energy even at low wind speeds.

3.2.3. Wind data

Wind observation data was taken from the KNMI (Royal Dutch Meteorological Institute). These datasets have hourly measurements wind speeds and directions, for a number of stations across Netherlands. For this purpose, the observations from the Twente station were taken, for the year 2015.

Measuring Station No	Measuring Station Name	LON (east)	LAT (north)	ALT (m)
290	TWENTHE	6.891	52.274	34.80

Table 4: Twente measuring station

Because the observations are made at the height of the measuring station, which is 10 m, the wind speeds have to be extrapolated to the specified turbine heights, in order to gain a more accurate estimate of the wind potential.

The wind speed values for different heights can be calculated with the following expression:

$$\nu_2 = \nu_1 \left(\frac{z_2}{z_1}\right)^{\alpha},$$

where ν presents wind speed at height z, and α is the wind shear exponent (Bauelos-Ruedas, Angeles-Camacho and Sebastin, 2011; Hadi, 2015). For the wind shear exponent value, we adopted the value 0.3 for the area of Enschede, which according to Bauelos-Ruedas et al. (2011) applies for the a smaller city area with trees and shrubs.

3.3. Problem definition

Given an urban or rural settlement of a certain size and land use classification, it is certainly challenging to assess to what degree could this area be exploited for wind farm projects. If we predefine a number of turbines types in potential use, the problem is to determine the most applicable type and number, and their locations, considering the shape and size of the available land, restricted areas for turbine placement, and turbine capacity and costs.

In terms of location modelling and optimization, this problem resembles a number of others, like the class of packing problems (packing in 2-dimensional containers, vertex and square packing), the knapsack problem, undesirable facility location problems and others (Savić, Šukilović and Filipović, 2011; Lei, Church and Lei, 2015).

An optimal placement scheme is determined by some optimization criteria. As explained in the previous chapters, when applying spatial optimization to the case of wind farm layout, the goal is to place the turbines in a way that maximizes a value of an objective function.

Among a number of factors described in the previous chapters, that influence wind farm design and development, in this case we are considering the following:

- Turbine type and number
- Turbine interdistance and location
- Turbine capacity, diameter size and height
- Wind speed
- Shape and size of the available land

3.4. Assumptions

During the model formulation and structuring of the algorithm, the following assumptions were made:

- Energy production of the specified turbines is modelled with a two-parameter logistic function, assuming that the wind turbines have identical power curves (Lu and Kim, 2014)
- Cost is a function of N number of turbines and Pr (installed power), assuming that the cost of a single turbine can be defined as 1 million € per MW of installed power
- The geographic area for turbine placement is a flat plane (which is certainly very likely to be the case in the Netherlands)
- The wind has uniform direction with hourly wind speed change, at turbine height. Because of the unpredictability of wind conditions, the annual energy production figures can be taken as a potential, not an exact calculation

- The model does not account for the effect of turbine wakes on energy production. In order to make this calculation, a set of flow and momentum theories would have to be adopted, which is out of the scope for this research. In this case, by adopting a 5 rotor diameter distance between the turbines, in both X and Y direction, a minimum wake influence on the energy production calculation is secured; moreover, for the sake of simplifying the calculation process and analysis in defining the cell size, the turbine rotor diameters were assigned values 50 m, 100 m and 150 m

3.5. Grid model

In a grid representation of space, the center of each cell is a potential location for placing a turbine. In Chapter 3.2.1 it is already mentioned that the discretization of the study area is done with a cell size of 250 m. As shown in the Figure 3.4 below, this secures the five diameter distance constraint for the smallest turbine. Center of each cell is assigned with a coordinate, that corresponds to the row and column number of this cell.

1, 1	1, 2	1, 3	1, 4	1, 5	1, 6
2, 1	2, 2	2, 3	2, 4	2, 5	2, 6
3, 1	3, 2	3, 3	3, 4	3, 5	3, 6
4, 1	4, 2	4, 3	4, 4	4, 5	4, 6
5, 1	5, 2	5, 3	5, 4	5, 5	5, 6
6, 1	6, 2	6, 3	6, 4	6, 5	6, 6

$5D_1$ $5D_2$ $5D_3$

Figure 3-4: Grid representation

In that sense, the bigger turbine will occupy the area of 2x2 grid cells, and the biggest turbine will occupy a 3x3 cell area. Turbine rotor diameters are not exactly the values of 50 m, 100 m and 150 m, but indeed 53 m, 90 m and 142 m, as shown in Table 5, so the distance constraint factors cx and cy, will have values slightly differing from number five, as shown in the table below:

Turbine	Raster cell size		D	5D	cx, cy
Enercon	1x1	250 m	53 m	265 m	4.71
Vestas	2x2	500 m	90 m	450 m	5.55
Siemens	3x3	750 m	142 m	710 m	5.28

Table 5: Turbine distance constraint factors calculation

These values of the distance constraint factors, cx and cy, will still insure a minimum influence of the wake effect on the energy production potential calculation.

3.6. Cost of Energy

Cost of Energy (CoE) criteria implies a solution that is a compromise between the investment costs and the amount of energy produced, the logic being that installing a higher number of turbines means more energy but at the same time more related cost. Using turbines with different capacities and rotor diameters complicates this choice a bit more, because bigger turbines can produce more energy, but they cost more and occupy more space.

In order to set up the framework for CoE calculation, the following parameters are introduced:

$A\,\{(i,j), where\ \ i = 1,2, \dots m, j = 1,2, \dots n\,\}\, set\ of\ X, Y\ coordinates\ for\ the\ study\ area$

$T -\ set\ of\ turbine\ types\ , and\ their\ characteristics$:

$T\ \{t1\ (Vi, Vr, Vo, Pr, D, H),\ \ t2\ (Vi, Vr, Vo, Pr, D, H),\ \ t3\ (Vi, Vr, Vo, Pr, D, H)\}$

$Vi -\ cut\ in\ speed$

$Vo -\ cut\ out\ speed$

$Vr -\ rated\ speed$

$Pr -\ rated\ power$

$v -\ wind\ speed\ at\ turbine\ height$

$D -\ turbine\ diameter$

$H -\ turbine\ height$

$N -\ number\ of\ turbines$

$cx, cy -\ distance\ constraint\ coefficients$

$AEP_{tot} -\ Annual\ energy\ production\ of\ N\ number\ of\ turbines$

$AEP\ (t) -\ Annual\ energy\ production\ for\ turbine\ of\ type\ t$

$C -\ cost\ factor\ for\ N\ number\ of\ turbines$

$CoE - Cost\ of\ Energy\ objective\ function$

Case 1 – Rectangular shaped wind farm area with one turbine type

When considering a square or rectangle shaped wind farm, designed with one turbine type, the following mathematical formulation should determine the number of turbines by minimizing the value of the objective CoE function:

$$minimize: CoE = \frac{Cost}{AEP_{tot}} \tag{1}$$

$Subject\ to$:

$$Cost = Pr * N * \left(\frac{2}{3} + \frac{1}{3} * e^{-0.00174*N^2}\right) \tag{2}$$

$$AEP_{tot} = N * AEP(t) \tag{3}$$

$$AEP(t) = \begin{cases} \frac{v - Vi}{Vr - Vi} Pr, & Vi \le v < Vr \\ Pr, & Vr \le v < Vo \\ 0, & else \end{cases} \tag{4}$$

$$N \le \left(\frac{Sx}{cx * D} + 1\right)\left(\frac{Sy}{cy * D} + 1\right) \tag{5}$$

$$Sx = Xmax - Xmin \tag{6}$$

$$Sy = Ymax - Ymin \tag{7}$$

Case 2 – Rectangular shaped wind farm area with three turbine types

If we are using different turbine types, the cost and energy output are calculated separately for each turbine type, depending on N number of each turbine installed, and the final result is a sum of these values:

$$CoE = \frac{Pr * N * \left(\frac{2}{3} + \frac{1}{3} * e^{-0.00174*N^2}\right)}{N * AEP(t)} \tag{8}$$

$$AEP_{tot} = N_{t1}AEP_{t1} + N_{t2}AEP_{t2} + N_{t3}AEP_{t3} \tag{9}$$

$$Cost = N_{t1}C_{t1} + N_{t2}C_{t2} + N_{t3}C_{t3} \tag{10}$$

The objective function to be minimized then takes on the following form:

$$CoE = \frac{N_{t1}AEP_{t1} + N_{t2}AEP_{t2} + N_{t3}AEP_{t3}}{N_{t1}C_{t1} + N_{t2}C_{t2} + N_{t3}C_{t3}} \tag{11}$$

Case 3 – Irregular shaped area with forbidden zones for turbine placement

In a more complicated case, the area for wind turbine placement is irregularly shaped, and raster cell values indicate what are the allowed locations for turbine placement, like in the example below.

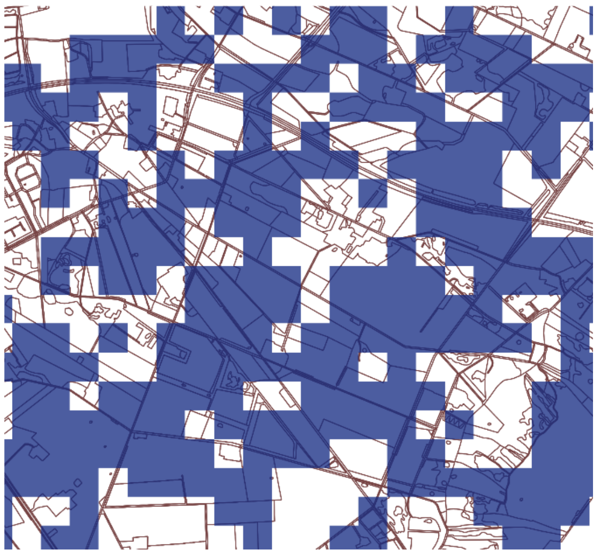


Figure 3-5: Irregularly shaped area with forbidden zones

The optimization procedure applied here should ”scan” the available area for turbine placement, and for a determined number of turbines calculate the value of the objective function.

3.7. Formulation of the optimization model

The combinatorial problem we are dealing with here can be put in the category of geometric packing problems. In the spirit of other GIS site-selection models, the proposed algorithm is based on a binary linear integer programming approach. Linear programming is commonly used to solve combinatorial optimization problem, and the binary type is a special case of the pure integer programming method (Motozawa, 2009; Chinneck, 2016). As already explained in Chapter 2, this implies decision variables that can take values 1 or 0 to indicate if a cell is selected for turbine placement. The formal mathematical expression for this would be:

$$maximize \sum_{i=1}^{n} c_i x_i \tag{8}$$

$$\sum_{j=1}^{m} a_{ij} x_i \leq b_j \tag{9}$$

$$x_i \in \{0,1\}, for\ i = 1,2,..,n \tag{10}$$

$$c_i > 0, for\ i = 1,2,..,n \tag{11}$$

The objective function has the form $maximize$, because the goal here is to maximize the N number of turbines placed in a designated area. The m number of inequality constraints serve as a condition to avoid overlapping of the turbine wake area. Last two conditions relate to defining the x_i as binary variables, and their coefficients c_i as non-negative numbers. In practical implementation number n represents the number of candidate turbine locations, enumerated by cell number from top left to bottom right, while m is equal to the number of cells in the input raster area. Algorithm operates with cell numbers, which are in the process converted to row and column numbers and finally point coordinates.

Implementation of the algorithm was done using R[1] programming language, and package lpSolve[2], that is an optimization framework in R for solving linear, integer and mixed integer programs.

[1] R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/ .

[2] Michel Berkelaar and others (2015). lpSolve: Interface to 'Lp_solve' v. 5.5 to Solve Linear/Integer Programs R package version 5.6.13. https://CRAN.R-project.org/package=lpSolve

3.8. Algorithm framework

The conceptual diagram shown in Figure 3-6, gives an overview of the implemented procedure and its specific steps.

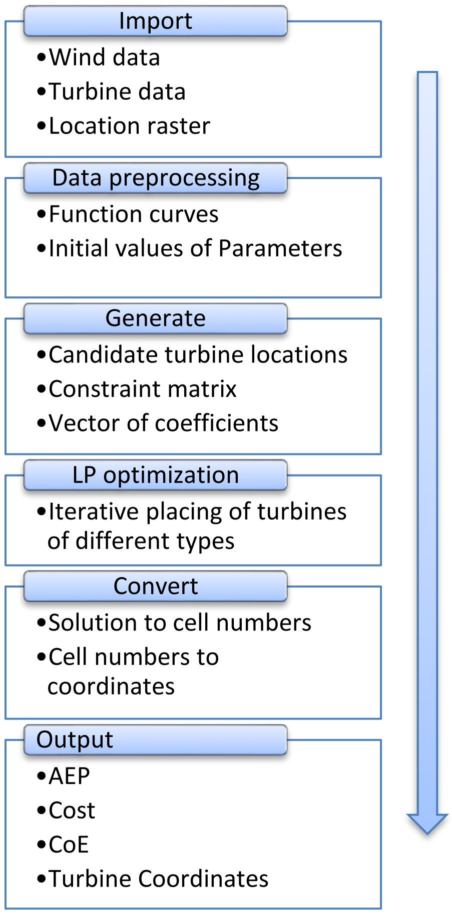


Figure 3-6: Algorithm framework

In order to explain how the allocation of turbines occurs, we are illustrating it on the following scenario:

1	1	1	0	0
1	1	1	1	1
1	1	1	1	1
1	1	1	0	0
0	0	1	1	1

We have an 5x5 area of 25 cells, out of which 19 cells are marked available for turbine placement (cells with value 1).

4	6	5	4	2
6	9	8	7	4
6	9	8	7	4
4	7	7	7	4
2	4	4	4	2

In the first iteration, the algorithm wants to assign a maximum possible number of the biggest turbines (which occupy a 3x3 cell area) in the available space. In order to generate candidate locations, it performs a summing of values for a 3x3 cell area, in every cell. Candidate turbine locations are cells with the value 9, and in this case there are two.

Referring to (8) and (9), number of candidate locations n in the first iteration is 2, and number m is equal to the number of cells, which is 25 in this case. Everything stated until now is formulated using matrix notation in the following way:

- X is the vector of numeric coefficients of objective function, size n

$$X^T = \{1, 1\}$$

- F is the matrix of numeric constraint coefficients, size $m\ x\ n$

$$F^T = \begin{Bmatrix} \{1,1,1,0,0,1,1,1,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0\} \\ \{0,0,0,0,0,1,1,1,0,0,1,1,1,0,0,1,1,1,0,0,0,0,0,0,0\} \end{Bmatrix}$$

- B is the vector of numerical values for the right-hand sides of the constraints, size m

$$B^T = \{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1\}$$

- Vector of size m that defines the direction for every constraint; all constraint have the same inequality sign ”<=”

The constraints defined here ensure that the wake areas of two turbines do not overlap, by limiting the sum of every row in the constraint matrix to 1.

The solver is initialized with the matrix input, and because the search is performed from the top left corner, it takes the first candidate cell as the solution.

1	1	1	0	0
1	1	1	1	1
1	$\cancel{1}$	1	1	1
1	1	1	0	0
0	0	1	1	1

It is obvious that only one of the two candidate turbine locations could be used as a solution, since their wake areas overlap.

1	1	1	0	0
1	1	1	1	1
1	1	1	1	1
1	1	1	0	0
0	0	1	1	1

In the next step, the turbine location cell and its wake area are marked unavailable and the next iteration can start.

0	0	0	0	0
0	0	0	1	1
0	0	0	1	1
1	1	1	0	0
0	0	1	1	1

This time the algorithm performs a search on the rest of the available cells, to determine locations for the next turbine type (2x2 cell area).

0	0	0	0	0
0	0	0	1	1
0	0	0	1	1
1	1	1	0	0
0	0	1	1	1

Following the identical steps from the previous iteration, the next solution is generated. There is only one candidate location in this case, and thus only one turbine is allocated in the available area.

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	1	1	0	0
0	0	1	1	1

After marking the solution cells from previous iteration as unavailable, the rest of the available area is assessed for placement of the third turbine type.

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	1	1	0	0
0	0	1	1	1

In this last iteration, in each available cell one turbine can be placed, so the allocation is done by a simple search of the available space

The number of turbines allocated in each iteration serves as the input for calculating the parameters of the CoE function, while the turbine locations are converted from cell numbers to point coordinates.

3.9. Software used

This research was conducted by using the following software and tools:

- Operating system: 64-bit Windows 10 Home
- Desktop GIS: QGIS 2.18.7
- Programming Language: R 3.4.1
- Integrated Development Environment (IDE): RStudio 1.0.143
- Referencing: Mendeley Desktop 1.17.10
- Office suite: Microsoft Office 365 ProPlus Student

4. Results

Following the methodological framework from the previous chapter, this chapter demonstrates and explains the results generated in testing and applying the algorithm procedure in different cases. The model is first validated with a number of hypothetical scenarios, after which it is applied to the study area in question.

4.1. Preprocessing output

After inputting the necessary data into the R environment, as a result of preprocessing, the algorithm generates turbine power curves that graphically demonstrate their performance, and curves for the objective function in question.

Parameters for the power curves are calculated using the expression (4) from previous chapter. This also graphically demonstrates what was already explained in Chapter 2, regarding the performance of wind turbines at different wind speeds. The biggest turbine has the biggest production capacity and vice versa, which is also obvious from the graph below.

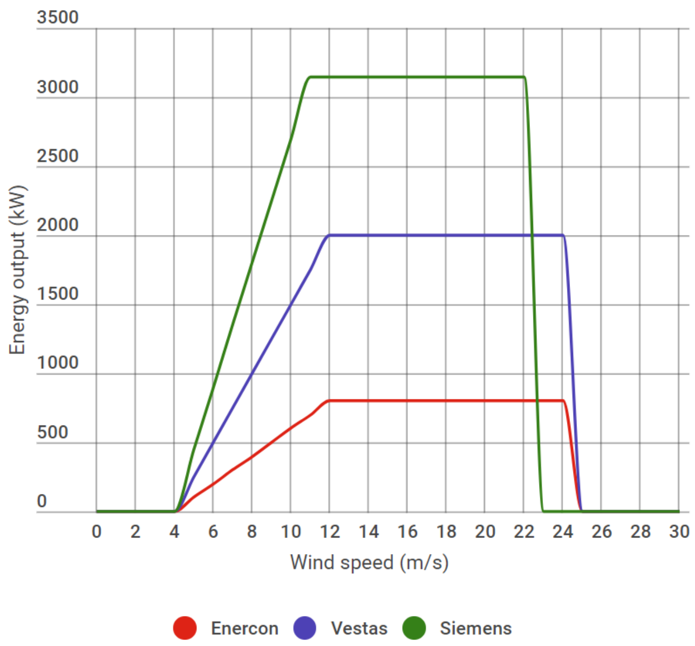


Figure 4-1: Energy output of three turbine types, for wind speed at hub height

Initial starting values of the cost and energy parameters, for one turbine, based on (2) and (4) are shown in the table below:

N = 1	Enercon	Vestas	Siemens
Cost (€)	800 000	2 000 000	3 150 000
AEP (kW)	2 477 536	7 531 257	14 363 679

Table 6: Initial values for cost and energy production

CoE function (8) is primarily in dependence of N (number of turbines), and as seen in Figure 4-2, using a turbine with a bigger capacity definitely results in lower values for the cost/energy ratio.

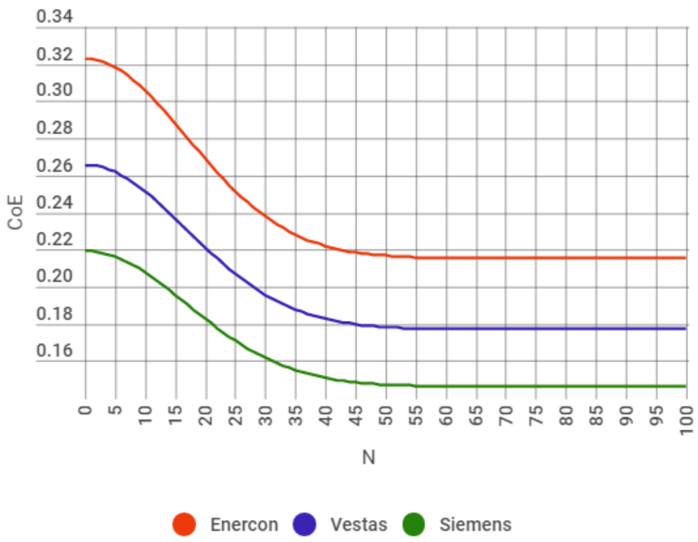


Figure 4-2: CoE function curves for three turbine types

This goes in favor of the "greedy" approach implemented here, where the algorithm first tries to place as many of the biggest turbines, then following the middle and smallest size.

4.2. Algorithm validation

In order to confirm the applicability of the implemented algorithm, its performance is tested on different cases, regarding area size and used turbines.

4.2.1. Case 1: One turbine type

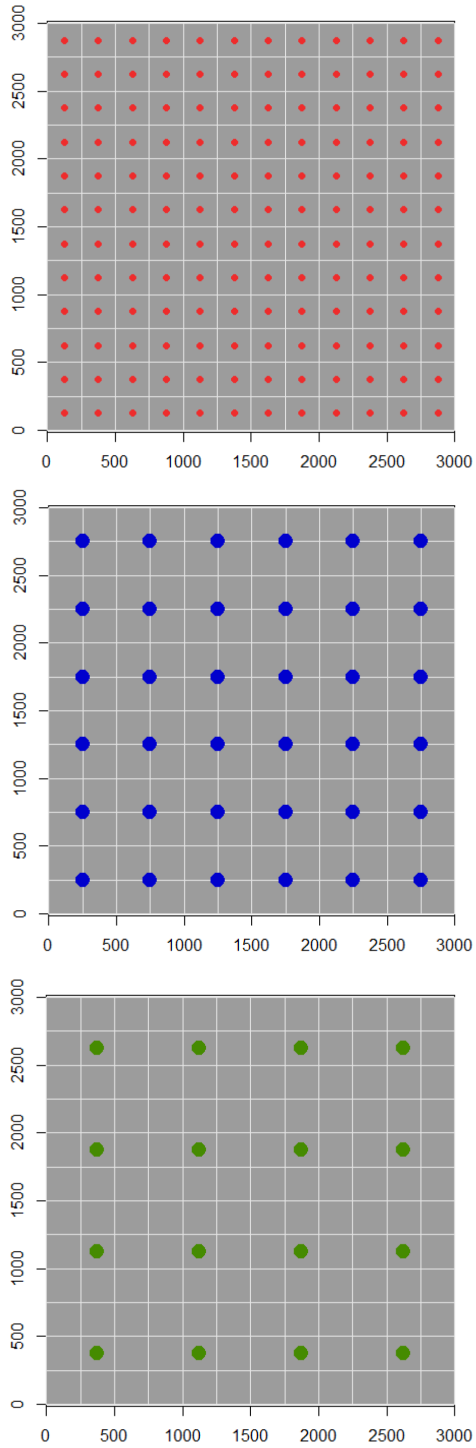
In the first scenario, we are analyzing the turbine layout for using just one type per area, with raster of 3000 m x 3000 m, 3000 m x 1000 m and 1000 m x 1000m size. Figure 4-3, 4-4 and 4-5 below, show the turbine placement for the given area sizes, respectively.

In this gridded representation, it is easy to determine the number of turbines, just by visual inspection. As explained in Chapter 3.5, the amount of space that every turbine occupies is defined by its wake area.

a)

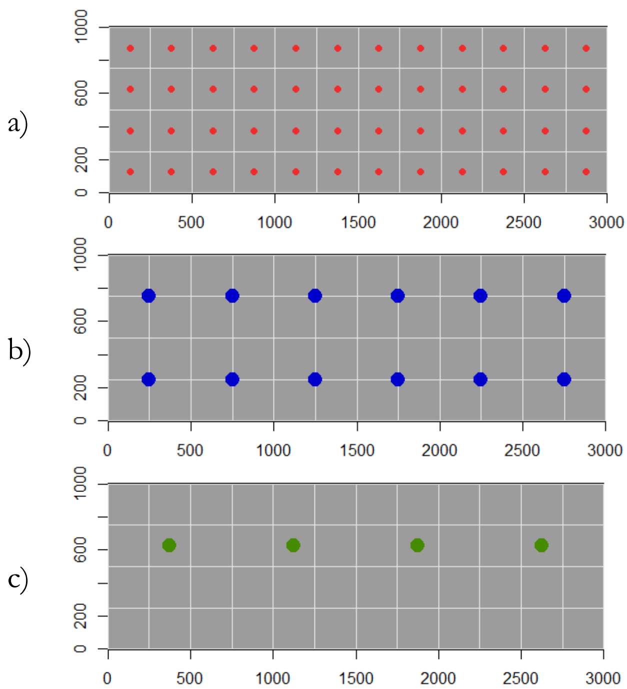
b)

c)



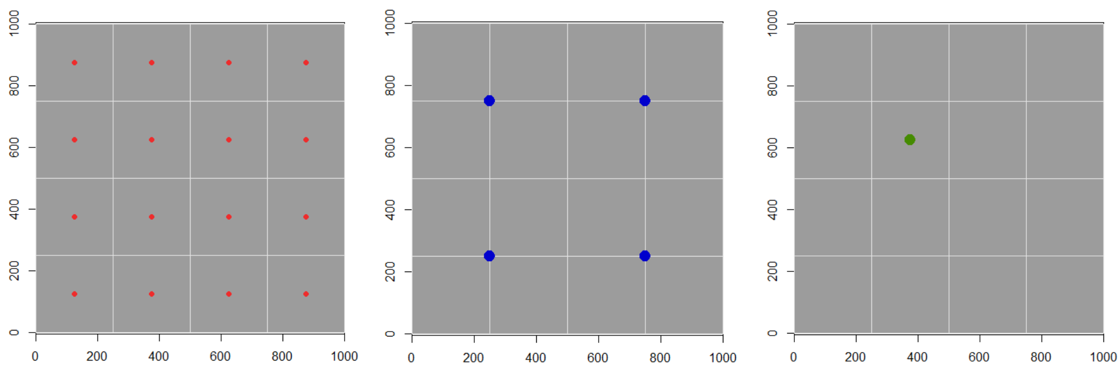
a) Enercon b) Vestas c) Siemens

Figure 4-3: Turbine layout for area 3000 m x 3000 m



a) Enercon b) Vestas c) Siemens

Figure 4-4: Turbine layout for area 3000 m x 1000 m



a) Enercon b) Vestas c) Siemens

Figure 4-5: Turbine layout for area 1000 m x 1000 m

Numerical values for the presented turbine settings are shown in Table 7. In any case, the number of placed Enercon turbines is the same as the number of available cells, since the cell size of 250 m x 250 m is already the size of the wake area for this smallest turbine type. The table allows for comparison of the parameter and CoE function values. It is obvious that using the smallest turbine type will result in highest cost per unit of energy produced, but it will yield the highest energy output. At the same time, in most cases using the biggest turbine type will have the lowest cost value of the CoE function, but at the expense of producing less energy. The middle-sized turbine then presents a satisfactory choice both in terms of cost and energy, so the question would be simply whether to put more weight on the cost or on the energy produced. In building commercial wind farms, this information is usually known upfront, meaning at least one of these values is a fixed one (financial limitations or required energy output).

Raster size	**1000 m x 1000 m**	**3000 m x 1000 m**	**3000 m x 3000 m**
Number of available cells	**16**	**48**	**144**
	Enercon		
CoE (€/kW)	0.284212	0.217221	0.215268
Cost (€)	11 266 314	25 832 349	76 800 000
AEP (kW)	39 640 582	118 921 746	356 765 237
N	16	48	144
	Vestas		
CoE (€/kW)	0.263130	0.245941	0.186323
Cost (€)	7 926 784	22 226 918	50 516 864
AEP (kW)	30 125 028	90 375 085	271 125 254
N	4	12	36
	Siemens		
CoE (€/kW)	0.219176	0.217296	0.193026
Cost (€)	3 148 175	12 484 685	44 361 112
AEP (kW)	14 363 679	57 454 716	229 818 866
N	1	4	16

Table 7: Values of CoE function parameters for given area sizes

4.2.2. Case 2: Three turbine types

In this case, the output was generated for using all turbine types on one area. This time the area sizes are 2500 m x 2500 m and 3500 m x 1500 m (chosen differently from previous case, because of the iterative structure of the algorithm that would result in the same turbine placement). The figures below demonstrate the iterative procedure, because the smallest and middle-sized turbines are placed only in the left available area after placing as many of the biggest turbines as possible. In the first test area (a), after placing the biggest turbines, there is no more place for the Vestas turbine type, so in the remaining cells the algorithm places the smallest turbine type. In Figure (b) it is the opposite case, so after the allocation of the first and second turbine type, there is no more place left for the smallest turbine size.

a)

b)

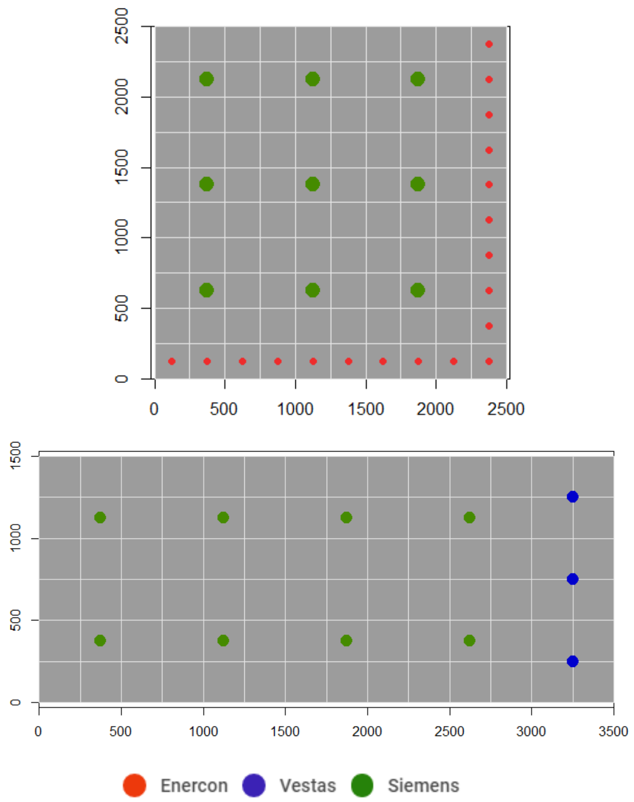


Figure 4-6: Turbine layout for 2500 m x 2500 m area size (a), and 3500 m x 1500 m (b)

Values of the parameters and CoE function here are calculated with (9), (10) and (11), and shown in Table 8.

Raster size	**2500 m x 2500 m**	**3500 m x 1500 m**
Number of available cells	**100**	**84**
N (E)	19	0
N (V)	0	3
N (S)	9	8
N_{tot}	27	11
Cost (€)	39 944 538	30 283 704
AEP (kW)	176 346 303	137 503 204
CoE (€/kW)	0.226512	0.220240

Table 8: Values of CoE function parameters for given area sizes (three turbine types)

If we were to disregard the biggest turbine and start from iteration two, the result would be much different, for the same test area:

a)

b)

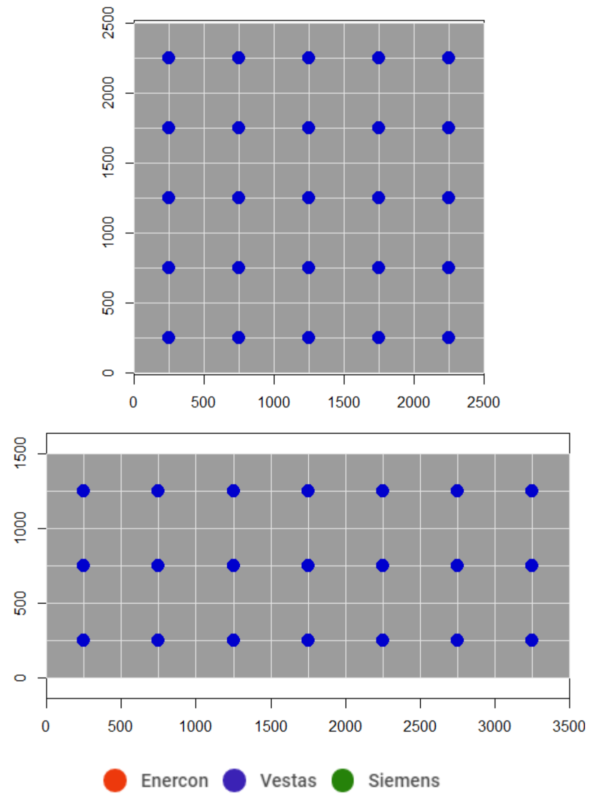


Figure 4-7: Turbine layout for 2500 m x 2500 m area size (a), and 3500 m x 1500 m (b) (without first iteration)

In this case the algorithm does not reach the third iteration, because the middle-size turbines take up all of the available space. By comparing the CoE parameters from Table 8 and Table 9, we can see that this setting actually results in a lower cost per unit of energy produced.

Raster size	**2500 m x 2500 m**	**3500 m x 1500 m**
Number of available cells	**100**	**84**
N (E)	0	0
N (V)	25	21
N (S)	0	0
N_{tot}	25	21
Cost (€)	38 950 968	34 499 449
AEP (kW)	188 281 426	158 156 398
CoE (€/kW)	0.2068763	0.218135

Table 9: Values of CoE function parameters for given area sizes (three turbine types, without first iteration)

4.2.3. Case 3: Three turbine types and area with forbidden zones

In the third case scenario, we are using a hypothetical raster of size 2500 m x 2500 m, that has approximately 70% of available area. This time the algorithm was able to place at least one of each turbine type. It is obvious from the figure below, that this placement depends primarily on the shape and size of the available area. By visual inspection of Figure (a) it is easily determined where could the biggest turbines be placed (because they require a 3x3 cell size area). In the remaining available space, only one Vestas turbine can fit, so for the rest of the available cells the algorithm assigns placement of turbine type three.

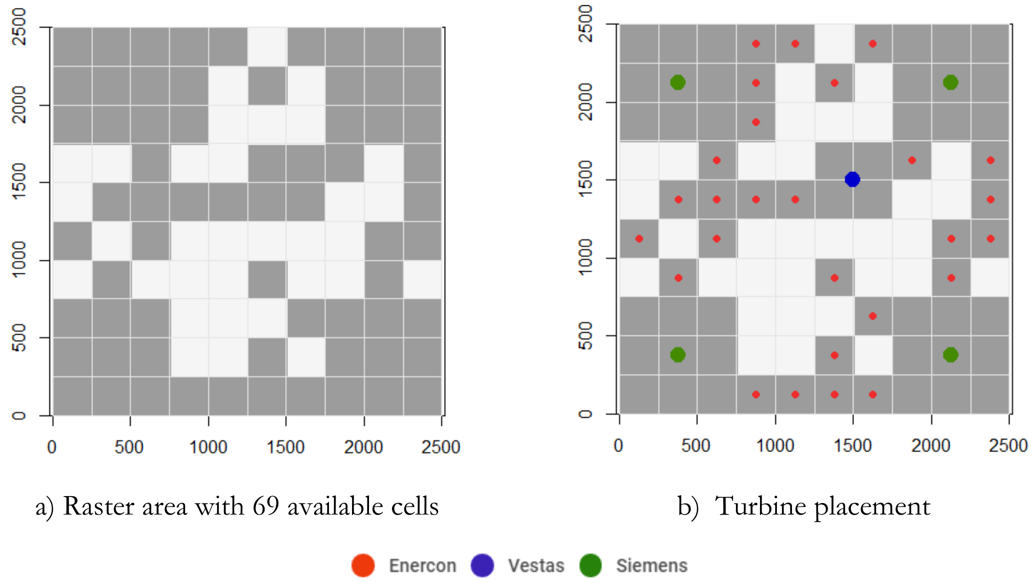


Figure 4-8: Turbine placement in a 2500 m x 2500 m area with forbidden zones

As described in previous cases, the values of the CoE parameters are calculated first for each turbine type, and the final value is a sum of this calculation.

Raster size	**2500 m x 2500 m**
Number of available cells	**69**
N (E)	27
N (V)	1
N (S)	4
N_{tot}	32
Cost (€)	39 944 538
AEP (kW)	131 879 455
CoE (€/kW)	0.234370

Table 10: Values of CoE function parameters for area with forbidden zones

Like in previous case, the algorithm was run first for all turbine types, and then just for the middle and smallest size. This again allowed for comparison of the CoE parameter values for both cases.

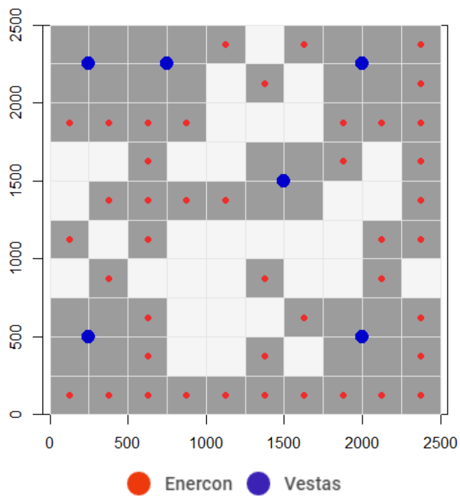


Figure 4-9: Turbine placement in area with forbidden zones (without first iteration)

The results again show that the second configuration has a higher energy production at a lower cost.

Raster size	**2500 m x 2500 m**
Number of available cells	**69**
N (E)	43
N (V)	6
N (S)	0
N_{tot}	49
Cost (€)	35 149 868
AEP (kW)	151 721 606
CoE (€/kW)	0.231673

Table 11: Values of CoE function parameters for an area with forbidden zones (without first iteration)

4.2.4. Summary

By running the algorithm on different test cases, it is established that the procedure developed here can correctly asses the size and shape of the available area, and place a number of turbines, or types of turbines, in this space.

Analysis of the numerical output also shows how different turbine configurations, on the same area, can have different CoE parameter values, that depend on the turbine type and number of turbines installed.

4.3. Applying the algorithm to the study area

After testing the algorithm performance on different hypothetical cases, the procedure was implemented on the study area in question. The complete R report for this case is presented in Appendix 1, together with turbine coordinates and other parameters. Here we are presenting the main parts of this report.

The solution took 35.728 seconds on a PC with Intel Core m3-5Y30 CPU at 0.90 GHz and 4GB RAM, under Windows OS. After importing the raster file into the R environment, as shown in Figure 4-10, it is converted in a rectangular form even though the initial input is clipped to the municipality border. This is necessary because the algorithm works on a matrix format, meaning the number of cells in rows and columns has to be the same. So in this case, the number of cells to be checked in each iteration goes up to 3472.

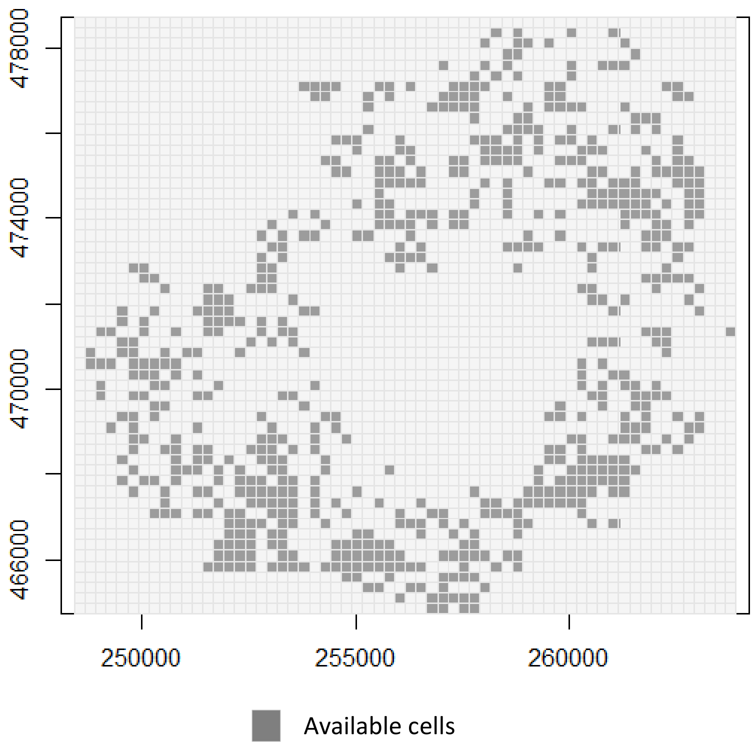


Figure 4-10: Input raster for the study area

As demonstrated before, the algorithm follows its iterative structure, and in that process, it allocated a total of 487 turbines in the study area.

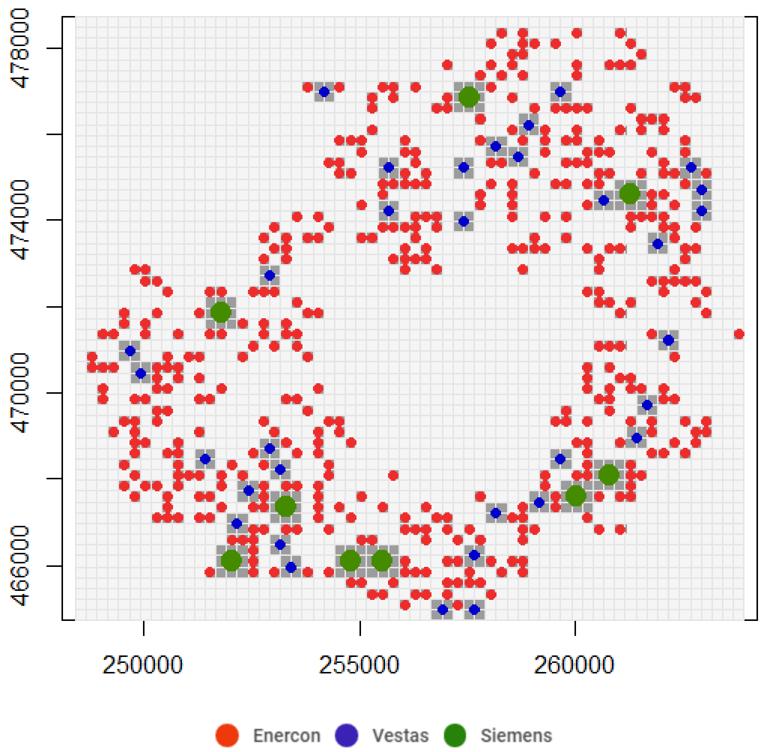


Figure 4-11: Turbine placement in the study area

Values of the numerical parameters for the study area are presented in the table below. They show a very high cost and energy output, which is reasonable considering the number of turbines placed.

Raster extent	**X_{max} = 263 908.6**
	X_{min} = 248 408.6
	Y_{min} = 464 717.4
	Y_{max} = 478 717.4
Number of cells (available/total)	**658/3472**
N (E)	445
N (V)	33
N (S)	9
N_{tot}	487
Cost (€)	311 748 515
AEP (kW)	1 480 308 280
CoE (€/kW)	0.210597

Table 12: CoE parameter values for study area implementation

After retrieving the turbine coordinates from the algorithm output, it was possible to indicate these locations on the land use map presented in Chapter 3. The result of this is presented in Figure 4-12.

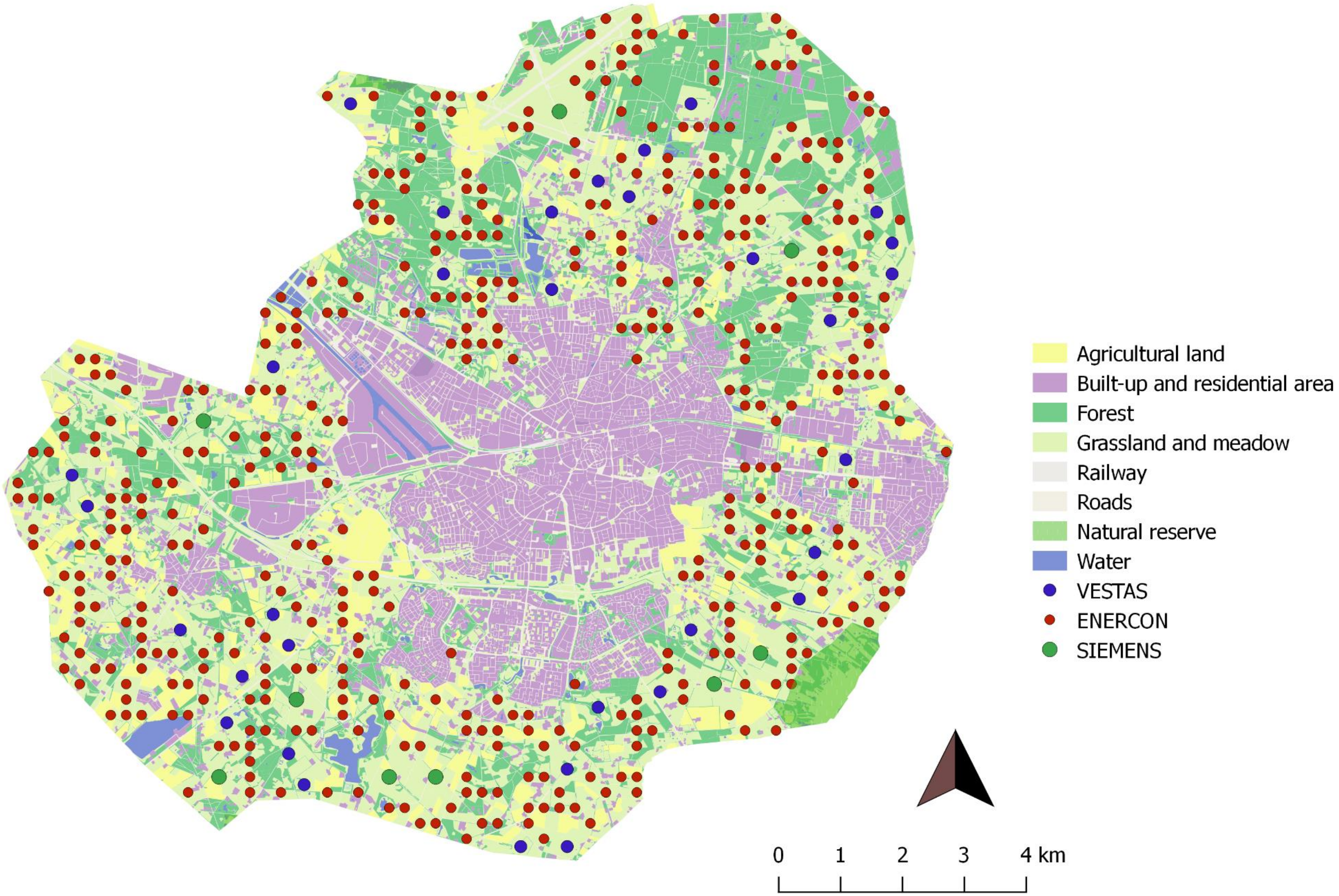


Figure 4-12: Land Use map of Enschede with turbine location

5. Discussion

In the previous chapters we presented the methodology that was followed in order to implement a spatial optimization model, as well as the output generated as a result of this implementation. Therefore it is also needed to conduct a critical examination of both, and see how and to what extent this approach answered the research questions posed in Chapter 1. This chapter tries to meet this requirement, by analyzing the results in different perspectives, stating the limitations of the research itself, as well as proposing ways of improvement in future work.

5.1. Results analysis

The WFLO problems are interdisciplinary, and they fall into the scope of different research fields. The optimization algorithm developed in this case, is one that tries to address the WFLO problem with a binary integer linear formulation. Linear programming is an exact optimization method, that searches for the optimal solution, but in more complex cases results in high computational times.

Solution in case of applying the algorithm to the study area took 35.728 s, but the number of cells to be checked in each iteration was 3472, which can to some extent explain the time of algorithm execution. Other reason could also be the speed of the computer processor in use. In other tests made on smaller raster sizes, the solution times took under 3 seconds. This means that although an exact method, binary linear programming still proved to be applicable to a more complex optimization problem like the WFO, and able to convey such formulation.

The accuracy of the assessment of the cost and energy production for different wind farm design scenarios, is constrained by the assumptions made in Chapter 3. That is why these figures should be understood as potential, and not exact values. On the other hand, the same assumptions were applied to all turbine types used in this case, which is why in a relative sense the accuracy can allow for comparison of the performance of these turbines.

Testing the turbine performance on areas of different shape and size, showed interesting results. If using a configuration with only one turbine type, it can be concluded that a bigger turbine will always be an optimal solution in terms of the CoE criteria. This is a reasonable conclusion, considering that the trends in this field are always towards constructing bigger and bigger turbines, and replacing the small ones. Results also showed that a configuration with turbines of different capacity (and different rotor diameters) can be a better option, than using just one turbine type on a designated area, because it allows for installment of more turbines and thus yields more energy. Depending on the size and shape of the available land, the results sometimes went in favor of using just the middle-sized turbine, Vestas, especially when testing on area with forbidden zones. The reason for this case is in the geometry of the available cell area, that did not allow a high number of the biggest turbines to be placed.

The algorithm proved to indeed find the optimal solution, in terms of fitting as many turbines as possible, in a designated available area. The potential limitation of this approach is that, although it generates an optimal solution, it generates one solution, even when there are more than one. Because the "scanning" of the available area is performed from top left to bottom right, it is not possible to know if a different arrangement of the Siemens turbines (with the same number of turbines), could result in placing more of the other turbine types. The possible workaround here would be in reordering the columns of the raster matrix, so that the shape and size of the available area stay the same, but the search would start and finish at different cells, thus maybe produce different optimal results.

As shown in Figure 4-12, when we placed the turbine coordinates on the original land use map from Chapter 3, it is obvious that some turbines fall on the border, or in the areas that were initially not defined as available. The reason for this is found in the different resolutions that we are dealing with. One is the resolution of the vector data set that was used, and the other is the resolution at which the rasterization is done, i.e. the cell size. Also because of the process of the rasterization itself, some vector features that are smaller than the prescribed raster cell size will still end up as an "available" cell in the output raster. On the other hand, larger available vector areas could be defined not available in the raster. This proves again that the results of any spatial analysis, in this case a MCA, is highly scale dependent. In this case, a possible answer would be to adjust the spatial resolution of the areal units, either by using a vector data set of smaller scale, or to define a bigger resolution in the process of rasterization. The latter would then of course increase the number of cells, and the computational time, as well as require redefining of the algorithm parameter setting.

5.2. Future work recommendations

In order to increase the performance and result usability of the developed approach, we believe that further research should be made. Because of the time shortage, only limited amount of parameters could be considered in the model implementation.

The wind farm cost is complex with a number of factors involved. In this case we defined the cost in function of the number of the installed turbines and their capacities, but in order to make a more realistic assessment the cost of different land could also be included as an input parameter. Right now the only information that the raster input has is the land availability, but different areas could also have different associated costs, or account for any other parameter like complex terrain. This information would then also be conveyed in the input raster cells, but would need a more powerful optimization solver to allow for such formulation, and come up with a result in reasonable computational time. Using a more powerful optimization method could then allow for adding parameters in any stage of the algorithm set, for example adding more turbine types as well as wind direction and wind speed distribution. Another improvement would be to use a more complex objective function, or a multi-objective formulation.

Although the primary objectives of this research were not directed at the actual visualization and presentation of the results as such, nor did time allow for it, in order to put the work done here in a broader context, it is necessary to mention this aspect also. Algorithm developed here could be incorporated into the already existing Desktop-based or Web-based SDSS (Spatial Decision Support Systems) and applications, used either in wind farm development and policy making, or in raising public awareness and social acceptability of wind farm projects. Of course that would require certain improvements and adjustment to the software architecture in question, and thus imply further research.

6. Conclusion

In this study, the primary focus was on developing a method that will facilitate the search for the best possible layout scheme for wind farms in a defined region, and on a more general level, to investigate the possible solutions for applying Multi-Criterial Decision Making methods in conjunction with GIS in spatial optimization of wind farms.

In order to satisfy the research goals stated in Chapter 1, the proposed methodology included a comprehensive literature review and overview of the modelling approaches in wind farm scenarios used up to date. After reviewing all the factors that influence wind farm development, and understanding the formulation of a spatial optimization problem, the WFL scenario was formulated into a binary integer linear problem and solved by the use of an appropriate optimization solver. Testing and analysis of different cases showed that binary integer linear programming and the use of GIS posses a great potential to aid in the process of development of wind farm projects. At the same time, these tools are highly adjustable, and in that sense can be adopted to many different scenarios. From the performed research and the output results, it can be concluded that the implemented methodology is effective in solving the given optimization problem.

The proposed algorithm aims to serve as a tool for preliminary screening in wind farm siting at a large scale. The assessment of energy production potential of an area and the associated cost, can assist in the decision-making process, providing necessary estimations in the earlier stages of the wind farm planning.

7. List of references

Azizi, A., Malekmohammadi, B., Jafari, H. R., Nasiri, H. and Amini Parsa, V. (2014) 'Land suitability assessment for wind power plant site selection using ANP-DEMATEL in a GIS environment: case study of Ardabil province, Iran', *Environmental Monitoring and Assessment*. Springer International Publishing, 186(10), pp. 6695–6709. doi: 10.1007/s10661-014-3883-6.

Bã Nos, R., Manzano-Agugliaro, F., Montoya, F. G., Gil, C., Alcayde, A. and Gómez, J. (2010) 'Optimization methods applied to renewable and sustainable energy: A review', *Renewable and Sustainable Energy Reviews*, 15, pp. 1753–1766. doi: 10.1016/j.rser.2010.12.008.

Bauelos-Ruedas, F., Angeles-Camacho, C. and Sebastin, S. (2011) 'Methodologies Used in the Extrapolation of Wind Speed Data at Different Heights and Its Impact in the Wind Energy Resource Assessment in a Region', in *Wind Farm - Technical Regulations, Potential Estimation and Siting Assessment*. InTech. doi: 10.5772/20669.

Borissova, D. and Mustakerov, I. (2017) 'Wind power plant layout design and assessment considering forbidden zones for location of turbines'.

Chinneck, J. W. (2016) 'Practical optimization: Binary and Mixed-Integer Programming', in.

Chowdhury, S., Messac, A., Zhang, J., Castillo, L. and Lebron, J. (2010) 'Optimizing the unrestricted placement of turbines of differing rotor diameters in a wind farm for maximum power generation'.

Church, R. L. (2002) 'Geographical information systems and location science', *Computers & Operations Research*, 29, pp. 541–562.

Coello, C. A. C., Lamont, G. B. and Van Veldhuizen, D. A. (2007) *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2nd edn.

Cortez, P. (2014) *Modern Optimization with R*.

Díaz Ignacio, G. J. (2016) *Multi-Objective Optimisation Algorithms for GIS-based Multi- Criteria Decision Analysis: An application for evacuation planning*.

Enercon (2017) *Enercon - Products and technical details*. Available at: http://www.enercon.de/en/products/ep-1/e-53/.

European Commission (2010) *A strategy for smart, sustainable and inclusive growth*.

Fagerfjäll, P. (2010) *Optimizing wind farm layout – more bang for the buck using mixed integer linear programming*. CHALMERS UNIVERSITY OF TECHNOLOGY.

Gigović, L., Pamučar, D., Božanić, D. and Ljubojević, S. (2017) 'Application of the GIS-DANP-MABAC multi-criteria model for selecting the location of wind farms: A case study of Vojvodina, Serbia', *Renewable Energy*, 103, pp. 501–521. doi: 10.1016/j.renene.2016.11.057.

Grassi, S., Chokani, N. and Abhari, R. S. (2012) 'Large scale technical and economical assessment of wind energy potential with a GIS tool: Case study Iowa'. doi: 10.1016/j.enpol.2012.01.061.

Grassi, S. and Klein, T. M. (2016) *3D augmented reality for improving social acceptance and public participation of wind farms*.

Gropp, W. and Moré, J. J. (1997) 'Optimization Environments and the NEOS Server', in *Approximation Theory and Optimization*. Cambridge University Press, pp. 167–182.

Gu, S. and Ji, L. (2010) 'Study on optimization of wind farm micro-layout', *Asia-Pacific Power and Energy Engineering Conference, APPEEC*, 96(2006), pp. 4–7. doi: 10.1109/APPEEC.2010.5448416.

Hadi, F. A. (2015) 'Diagnosis of the Best Method for Wind Speed Extrapolation Weibull scale factor at h 1 height k 1 Weibull shape factor at h 1 height c 2 Weibull scale factor at h 2 height k 2 Weibull scale factor at h 2 height', *International Journal of Advanced Research in Electrical*, 4(10). doi: 10.15662/IJAREEIE.2015.0410058.

Herbert-Acero, J., Probst, O., Réthoré, P.-E., Larsen, G. and Castillo-Villar, K. (2014) 'A Review of Methodological Approaches for the Design and Optimization of Wind Farms', *Energies*. Multidisciplinary Digital Publishing Institute, 7(11), pp. 6930–7016. doi: 10.3390/en7116930.

Höfer, T., Sunak, Y., Siddique, H. and Madlener, R. (2016) 'Wind farm siting using a spatial Analytic Hierarchy Process approach: A case study of the Städteregion Aachen', *Applied Energy*, 163, pp. 222–243. doi: 10.1016/j.apenergy.2015.10.138.

IRENA (International Renewable Energy Agency) (2012) *Renewable Energy Cost Analysis: Wind Power*.

Kwong, W. Y., Zhang, P. Y., Romero, D., Moran, J., Morgenroth, M., Amon, C., Romero, D., Amon, C. and Moran, J. (2012) 'Wind Farm Layout Optimization Considering Energy Generation and Noise Propagation', *Volume 3: 38th Design Automation Conference, Parts A and B*. ASME, pp. 1–10. doi: 10.1115/DETC2012-71478.

Latinopoulos, D. and Kechagia, K. (2015) 'A GIS-based multi-criteria evaluation for wind farm site selection. A regional scale application in Greece', *Renewable Energy*, 78, pp. 550–560. doi: 10.1016/j.renene.2015.01.041.

Lei, T. T., Church, R. L. and Lei, Z. (2015) 'A unified approach for location-allocation analysis: integrating GIS, didtributed computing and spatial optimization', *International Journal of Geographical Information Science*. doi: 10.1080/13658816.2015.1041959.

Lu, S. and Kim, H. M. (2014) 'Wind farm layout design optimization through multi-scenario decomposition with complementarity constraints'. doi: 10.1080/0305215X.2013.861457.

Malczewski, J. and Rinner, C. (2015) *Advances in Geographic Information Science Multicriteria Decision Analysis in Geographic Information Science*.

McWilliam, M. K., van Kooten, G. C. and Crawford, C. (2012) 'A method for optimizing the location of wind farms', *Renewable Energy*, 48, pp. 287–299. doi: 10.1016/j.renene.2012.05.006.

Mittal, A. (2010) *Optimization of the layout of large wind farms using a genetic algorithm*. CASE WESTERN RESERVE UNIVERSITY.

Mittal, P., Kulkarni, K. and Mitra, K. (2016) 'Multi-objective optimization of energy generation and noise propagation: A hybrid approach', in *2016 Indian Control Conference (ICC)*. IEEE, pp. 499–506. doi: 10.1109/INDIANCC.2016.7441181.

Mosetti, G., Poloni, C. and Diviacco, B. (1994) 'Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm', *Journal of Wind Engineering and Industrial Aerodynamics*, 51, pp. 105–116.

Motozawa, Y. (2009) *Analysis of Linear, Integer, and Binary Programming and their Applications*. University of Houston.

Mustakerov, I. and Borissova, D. (2010) 'Wind turbines type and number choice using combinatorial optimization', *Renewable Energy*, 35, pp. 1887–1894. doi: 10.1016/j.renene.2009.12.012.

PIERROT, M. (2017) *Windfarms by country - Netherlands*. Available at: http://www.thewindpower.net/country_windfarms_de_10_niederlande.php.

Pohekar, S. D. and Ramachandran, M. (2004) 'Application of multi-criteria decision making to sustainable energy planning—A review', *Renewable and Sustainable Energy Reviews*, 8, pp. 365–381. doi: 10.1016/j.rser.2003.12.007.

Royal Dutch Meteorological Institute (2015) *KNMI Kennis en datacentrum*. Available at: http://projects.knmi.nl/klimatologie/uurgegevens/selectie.cgi.

Samorani, M. (2013) 'The Wind Farm Layout Optimization Problem', in. Springer Berlin Heidelberg, pp. 21–38. doi: 10.1007/978-3-642-41080-2_2.

Savić, A., Šukilović, T. and Filipović, V. (2011) 'SOLVING THE TWO-DIMENSIONAL PACKING PROBLEM WITH m-M CALCULUS', *Yugoslav Journal of Operations Research*, 21(1), pp. 93–102. doi: 10.2298/YJOR1101093S.

Scholz, J. (2016) *IFG - Spatial Optimization*, *Graz University of Technology*. Available at: https://www.tugraz.at/institute/ifg/research/geoinformation/spatial-optimization/.

Senter Novem (2005) *Windkaart van Nederland*.

Siemens (2017) *Wind Turbines Portfolio*. Available at: https://www.siemens.com/global/en/home/markets/wind/turbines-and-services/former-portfolio.html.

Slikker, I. (2016) *A web-based application to increase public awareness about wind turbine site selection challenges: A case study for the Netherlands*.

Sunak, Y., Madlener, R., Höfer, T., Siddique, H. and De Doncker, R. W. (2015) 'A GIS-based Decision Support System for the Optimal Siting of Wind Farm Projects', *E.ON Energy Research Center Series*, 7(2).

Szurek, M., Blachowski, J. and Nowacka, A. (2014) 'GIS-based method for wind farm location multi-criteria analysis', *Mining Science*, 21(2014), pp. 65–81. doi: 10.5277/ms142106.

Tesauro, A. ;, Réthoré, P.-E., Larsen, G. C., Tesauro, A., Etho, P.-E. and Larsen, G. C. (2012) 'State of the art of wind farm optimization', in *Proceedings of EWEA 2012 - European Wind Energy Conference & Exhibition*.

Tong, D. and Murray, A. T. (2012) 'Spatial Optimization in Geography', *Annals of the Association of American Geographers*.

Vestas (2017) *V90-1.8/2.0 MW® at a Glance*. Available at: https://www.vestas.com/en/products/turbines/v90-2_0_mw#!aep.

Villacreses, G., Gaona, G., Martínez-Gómez, J. and Jijón, D. J. (2017) 'Wind farms suitability location using geographical information system (GIS), based on multi-criteria decision making (MCDM) methods: The case of continental Ecuador', *Renewable Energy*, 109, pp. 275–286. doi: 10.1016/j.renene.2017.03.041.

Wind EUROPE (2016) *2016 European statistics Wind in power*.

Yamani, S., Sorkhabi, D., Romero, D. A., Yan, G. K., Gu, M. D., Moran, J., Morgenroth, M. and Amon, C. H. (2016) 'The impact of land use constraints in multi-objective energy-noise wind farm layout optimization', *Renewable Energy*, 85, pp. 359–370. doi: 10.1016/j.renene.2015.06.026.

Zhang, P. Y. (2013) *Topics in Wind Farm Layout Optimization: Analytical Wake Models, Noise Propagation, and Energy Production*. University of Toronto.

8. Appendices

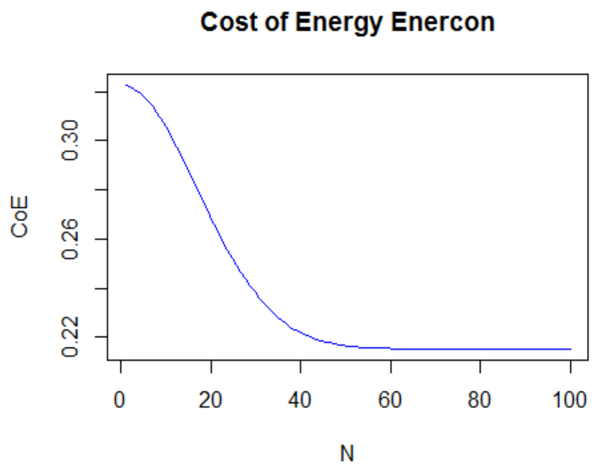
APPENDIX 1 – COMPILED R REPORT

wflo.R

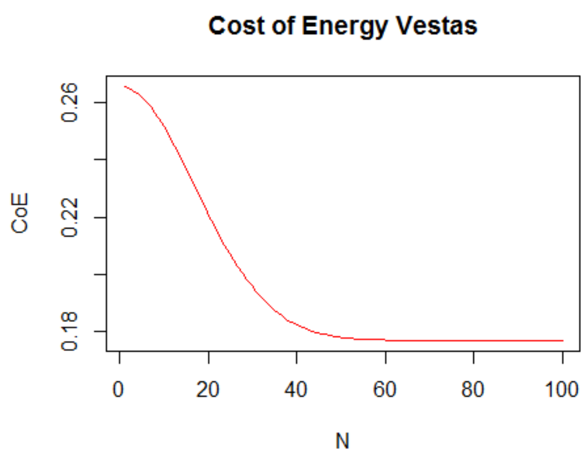
Mon Sep 11 20:23:43 2017

```
old <- Sys.time()
library(raster)
library(sp)
library(lpSolve)
library(ggplot2)
library(rgdal)
#Import wind data
ws10 <- read.csv("10.csv", header=FALSE, sep=",")
wsE <- ws10[,1]
wsV <- ws10[,2]
wsS <- ws10[,3]
#Import turbine specifications
wtur <- read.csv("wtur.csv", header=TRUE, sep=",")
row <- c("D", "Vi", "Vo", "Vr", "Pr", "H")
row.names(wtur) <- row
coll <- c("Siemens", "Vestas", "Enercon")
wtur <- (`colnames<-`(wtur, coll))
viE <- as.numeric(wtur["Vi","Enercon"])
vrE <- as.numeric(wtur["Vr","Enercon"])
voE <- as.numeric(wtur["Vo","Enercon"])
dE <- as.numeric(wtur["D","Enercon"])
hE <- as.numeric(wtur["H","Enercon"])
prE <- as.numeric(wtur["Pr","Enercon"])
viV <- as.numeric(wtur["Vi","Vestas"])
vrV <- as.numeric(wtur["Vr","Vestas"])
voV <- as.numeric(wtur["Vo","Vestas"])
dV <- as.numeric(wtur["D","Vestas"])
hV <- as.numeric(wtur["H","Vestas"])
prV <- as.numeric(wtur["Pr","Vestas"])
viS <- as.numeric(wtur["Vi","Siemens"])
vrS <- as.numeric(wtur["Vr","Siemens"])
voS <- as.numeric(wtur["Vo","Siemens"])
dS <- as.numeric(wtur["D","Siemens"])
hS <- as.numeric(wtur["H","Siemens"])
prS <- as.numeric(wtur["Pr","Siemens"])
#Calculate Energy Production for three turbine types
power.curveE <- function(x) {
  if (x>=viE && x<vrE)
  {P <- (x-viE)/(vrE-viE)*prE }
  else if (x>=vrE && x<voE)
  {P <- prE}
  else {P <-0}
  return(P)
}
power.curveV <- function(x) {
  if (x>=viV && x<vrV)
  {P <- (x-viV)/(vrV-viV)*prV}
  else if (x>=vrV && x<voV)
  {P <- prV}
  else {P <-0}
  return(P)
}
power.curveS <- function(x) {
  if (x>=viS && x<vrS)
  {P <- (x-viS)/(vrS-viS)*prS }
  else if (x>=vrS && x<voS)
  {P <- prS}
  else {P <-0}
```

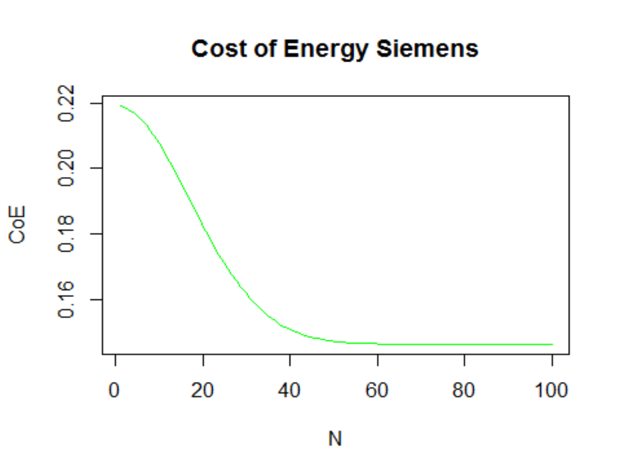
```
  return(P)
}
pwoE <- sapply(wsE, power.curveE)
pwoV <- sapply(wsV, power.curveV)
pwoS <- sapply(wsS, power.curveS)
aepE <- sum(pwoE)
aepV <- sum(pwoV)
aepS <- sum(pwoS)

#Cost of Energy functions
#ENERCON
obj.funE <- function(N) {
  C <- N*prE*1000*(2/3+1/3*exp(-0.00174*N^2))
  E <- N*aepE
  CoE <- C/E
  return(CoE)
}
curve(obj.funE, from = 1, to =100, xlab = "N", ylab="CoE", col="blue", main="Cost of
Energy Enercon")
```



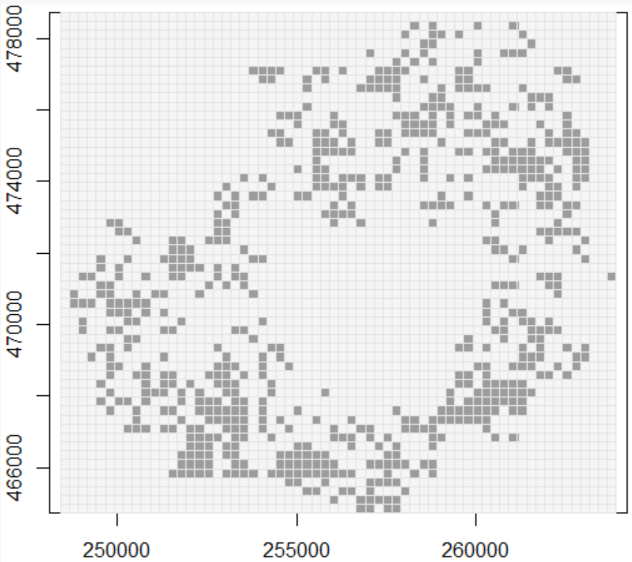

```
#VESTAS
obj.funV <- function(N) {

  C <- N*prV*1000*(2/3+1/3*exp(-0.00174*N^2))
  E <- N*aepV
  CoE <- C/E
  return(CoE)
}
curve(obj.funV, from = 1, to =100, xlab = "N", ylab="CoE", col="red", main="Cost of E
nergy Vestas")
```



```r
#SIEMENS
obj.funS <- function(N) {
  C <- N*prS*1000*(2/3+1/3*exp(-0.00174*N^2))
  E <- N*aepS
  CoE <- C/E
  return(CoE)
}
curve(obj.funS, from = 1, to =100, xlab = "N", ylab="CoE", col="green", main="Cost of
 Energy Siemens")
```

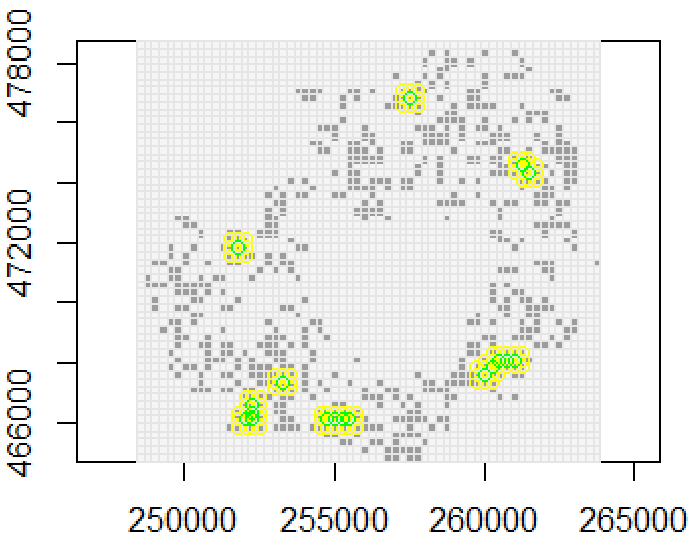



```r
#OPTIMIZATION PREPROCESSING


#raster input
enschede1 <- raster::raster('11111.tif')
qgis <- enschede1
A <- values(qgis)
plot(qgis, breaks=c(0,0.2,1),col=c("gray96", "gray61", "gray61"), legend=FALSE)
plot(rasterToPolygons(qgis), add=TRUE, border='gray90', lwd=1)
```



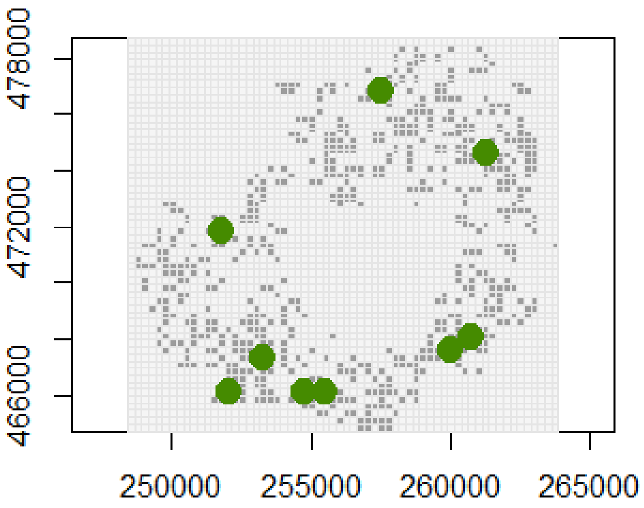

```r
#First iteration - Siemens turbines

#generation of candidate locations

coordqgis <- coordinates(qgis)
matrix <- as.matrix(qgis)
```

```r
result <- matrix(0, nrow=nrow(qgis), ncol=ncol(qgis))
for (i in 2:(nrow(qgis)-1)) {
  for (j in 2:(ncol(qgis)-1)) {
    result[i,j] <- sum(matrix[i,j], matrix[i-1,j-1], matrix[i,j-1], matrix[i+1,j-1],m
atrix[i-1,j],matrix[i+1,j],matrix[i-1,j+1], matrix[i,j+1], matrix[i+1,j+1])
  }
}

inds_posible <- which(result==9, arr.ind = TRUE)
count_posible <- NROW(inds_posible)
options_cellnumber <- cellFromRowCol(qgis, inds_posible[,1], inds_posible[,2])
plotcoord <- coordqgis[options_cellnumber,]
points(plotcoord, col="green")
M <- ncell(qgis)
N <- count_posible
row_pos <- matrix (0, nrow=9, ncol=N )
col_pos <- matrix (0, nrow=9, ncol=N )
for (n in 1:N) {
  row_pos [,n] <-  c(inds_posible[n,1], inds_posible[n,1]-1,inds_posible[n,1]-1,inds_
posible[n,1]-1,inds_posible[n,1],inds_posible[n,1],inds_posible[n,1]+1,inds_posible[n
,1]+1,inds_posible[n,1]+1)
  col_pos [,n]<-  c(inds_posible[n,2],inds_posible[n,2]-1, inds_posible[n,2], inds_po
sible[n,2]+1, inds_posible[n,2]-1,inds_posible[n,2]+1, inds_posible[n,2], inds_posibl
e[n,2]-1,inds_posible[n,2]+1)
}
options_neighbourhoodcellnumber <- matrix(0, nrow=9, ncol=N )
for (n in 1:N) {
options_neighbourhoodcellnumber[,n] <- cellFromRowCol(qgis, row_pos[,n], col_pos[,n])
}
plotcoord_neighbourhood <- coordqgis[as.vector(t(options_neighbourhoodcellnumber)),]
points(plotcoord_neighbourhood, col="yellow")
points(plotcoord, col="green")
```



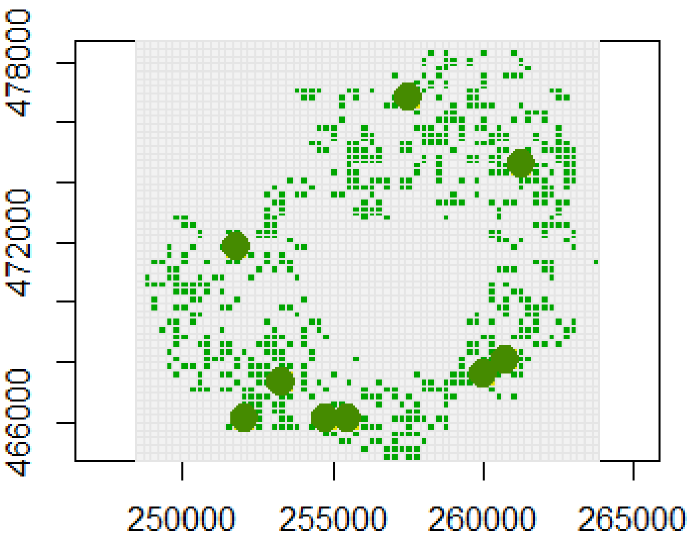

```r
resultbin <- matrix (0,nrow=M, ncol=N)
for ( n in 1:N) {
 resultbin[options_neighbourhoodcellnumber[,n],n] <- 1
}
#generate input for optimization algorithm
f.obj <-  rep(1,N)
f.con <- resultbin
f.dir <- rep("<=", M)
f.rhs <- rep (1,M)
```

```
#initialize LP solver
lp ( "max", f.obj, f.con, f.dir, f.rhs)

## Success: the objective function is 9

solution <- lp ( "max", f.obj, f.con, f.dir, f.rhs)$solution
solution

##  [1] 1 1 0 0 0 1 1 0 0 1 1 1 0 0 1 0 1 0

obj.value <- sum(solution)
obj.value

## [1] 9

solution_cellno <- options_cellnumber[which(solution==1, arr.ind = TRUE)]
solution_coord <- coordqgis[options_cellnumber[which(solution==1, arr.ind = TRUE)],]
solution_coord #coordinates of Siemens turbines

##              x        y
##  [1,] 251783.6 471842.4
##  [2,] 252033.6 466092.4
##  [3,] 253283.6 467342.4
##  [4,] 254783.6 466092.4
##  [5,] 255533.6 466092.4
##  [6,] 257533.6 476842.4
##  [7,] 260033.6 467592.4
##  [8,] 260783.6 468092.4
##  [9,] 261283.6 474592.4

plot(qgis, breaks=c(0,0.2,1),col=c("gray96", "gray61", "gray61"), legend=FALSE)
plot(rasterToPolygons(qgis), add=TRUE, border='gray90', lwd=1)
points(solution_coord, col="chartreuse4", bg="chartreuse4", pch=21, cex=2)
```




```
# Second iteration - Vestas turbines

#Preprocessing

solution_rowcol <- rowColFromCell(qgis, solution_cellno)
solution_neigh_row_pos <- matrix (0, nrow=9, ncol=obj.value )
solution_neigh_col_pos <- matrix (0, nrow=9, ncol=obj.value )
for (k in 1:obj.value) {
  solution_neigh_row_pos [,k] <-  c(solution_rowcol[k,1], solution_rowcol[k,1]-1,solu
tion_rowcol[k,1]-1,solution_rowcol[k,1]-1,solution_rowcol[k,1],solution_rowcol[k,1],s
olution_rowcol[k,1]+1,solution_rowcol[k,1]+1,solution_rowcol[k,1]+1)
  solution_neigh_col_pos [,k]<-  c(solution_rowcol[k,2],solution_rowcol[k,2]-1, solut
```

```
ion_rowcol[k,2], solution_rowcol[k,2]+1, solution_rowcol[k,2]-1,solution_rowcol[k,2]+
1, solution_rowcol[k,2], solution_rowcol[k,2]-1,solution_rowcol[k,2]+1)
}
solution_neighbourhoodcellnumber <- matrix(0, nrow=9, ncol=obj.value )
for (k in 1:obj.value) {
  solution_neighbourhoodcellnumber[,k] <- cellFromRowCol(qgis, solution_neigh_row_pos
[,k], solution_neigh_col_pos[,k])
}

A [solution_neighbourhoodcellnumber] <- 0.5
A [solution_cellno] <- 0.2

values(qgis) <- A
plot(qgis,  legend=FALSE)
plot(rasterToPolygons(qgis), add=TRUE, border='gray90', lwd=1)
points(solution_coord, col="chartreuse4", bg="chartreuse4", pch=21, cex=2)
```




```
A [solution_neighbourhoodcellnumber] <- 0
values(qgis) <- A
matrix2 <- as.matrix(qgis)

#generation of candidate locations
result2 <- matrix(0, nrow=nrow(qgis), ncol=ncol(qgis))
for (i in 1:(nrow(qgis)-1)) {
  for (j in 1:(ncol(qgis)-1)) {
    result2[i,j] <- sum(matrix2[i,j], matrix2[i+1,j+1], matrix2[i,j+1], matrix2[i+1,j
])
  }
}

inds_posible2 <- which(result2==4, arr.ind = TRUE)
count_posible2 <- NROW(inds_posible2)
options_cellnumber2 <- cellFromRowCol(qgis, inds_posible2[,1], inds_posible2[,2])
plotcoord2 <- coordqgis[options_cellnumber2,]
points(plotcoord2, col="red")
M <- ncell(qgis)
N2 <- count_posible2
row_pos2 <- matrix (0, nrow=4, ncol=N2 )
col_pos2 <- matrix (0, nrow=4, ncol=N2 )
for (n in 1:N2) {
  row_pos2 [,n] <-  c(inds_posible2[n,1], inds_posible2[n,1],inds_posible2[n,1]+1,ind
s_posible2[n,1]+1)
  col_pos2 [,n]<-  c(inds_posible2[n,2],inds_posible2[n,2]+1, inds_posible2[n,2]+1, i
nds_posible2[n,2])
}
```

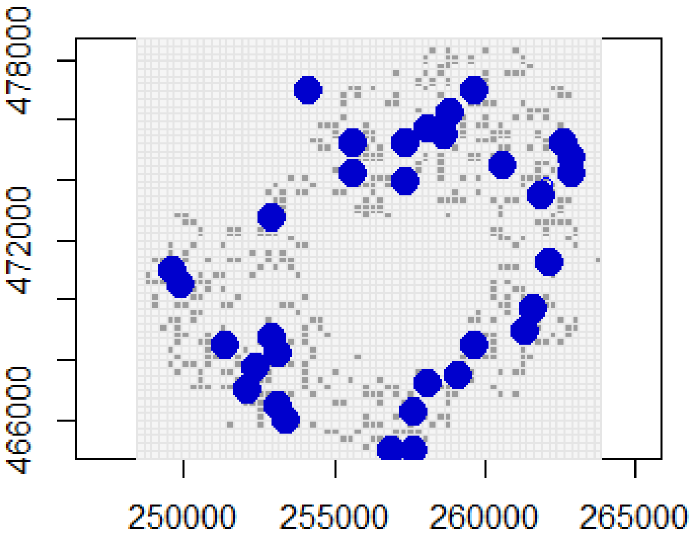
```
options_neighbourhoodcellnumber2 <- matrix(0, nrow=4, ncol=N2 )
for (n in 1:N2) {
  options_neighbourhoodcellnumber2[,n] <- cellFromRowCol(qgis, row_pos2[,n], col_pos2
[,n])
}
plotcoord_neighbourhood2 <- coordqgis[as.vector(t(options_neighbourhoodcellnumber2)),
]
resultbin2 <- matrix (0,nrow=M, ncol=N2)
for ( n in 1:N2) {
  resultbin2[options_neighbourhoodcellnumber2[,n],n] <- 1
}


#generate input for optimization algorithm
f.obj2 <-  rep(1,N2)
f.con2 <- resultbin2
f.dir <- rep("<=", M)
f.rhs <- rep (1,M)
f.rhs[options_cellnumber] <- 1

#initialize LP solver
lp ( "max", f.obj2, f.con2, f.dir, f.rhs)

## Success: the objective function is 33

solution2 <- lp ( "max", f.obj2, f.con2, f.dir, f.rhs)$solution
solution2

##  [1] 1 1 0 1 1 0 1 0 0 0 1 0 1 0 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 0 1 0 1 1 1
## [36] 0 1 0 1 1 1 1 1 0 1 1 0 0 1 0 1

obj.value2 <- sum(solution2)
obj.value2

## [1] 33

solution_cellno2 <- options_cellnumber2[which(solution2==1, arr.ind = TRUE)]
solution_coord2 <- coordqgis[options_cellnumber2[which(solution2==1, arr.ind = TRUE)]
,]
solution_coord2

##              x        y
##  [1,] 249533.6 471092.4
##  [2,] 249783.6 470592.4
##  [3,] 251283.6 468592.4
##  [4,] 252033.6 467092.4
##  [5,] 252283.6 467842.4
##  [6,] 252783.6 472842.4
##  [7,] 252783.6 468842.4
##  [8,] 253033.6 468342.4
##  [9,] 253033.6 466592.4
## [10,] 253283.6 466092.4
## [11,] 254033.6 477092.4
## [12,] 255533.6 475342.4
## [13,] 255533.6 474342.4
## [14,] 256783.6 465092.4
## [15,] 257283.6 475342.4
## [16,] 257283.6 474092.4
## [17,] 257533.6 466342.4
## [18,] 257533.6 465092.4
## [19,] 258033.6 475842.4
## [20,] 258033.6 467342.4
## [21,] 258533.6 475592.4
## [22,] 258783.6 476342.4
## [23,] 259033.6 467592.4
## [24,] 259533.6 477092.4
```

```
## [25,] 259533.6 468592.4
## [26,] 260533.6 474592.4
## [27,] 261283.6 469092.4
## [28,] 261533.6 469842.4
## [29,] 261783.6 473592.4
## [30,] 262033.6 471342.4
## [31,] 262533.6 475342.4
## [32,] 262783.6 474842.4
## [33,] 262783.6 474342.4
```

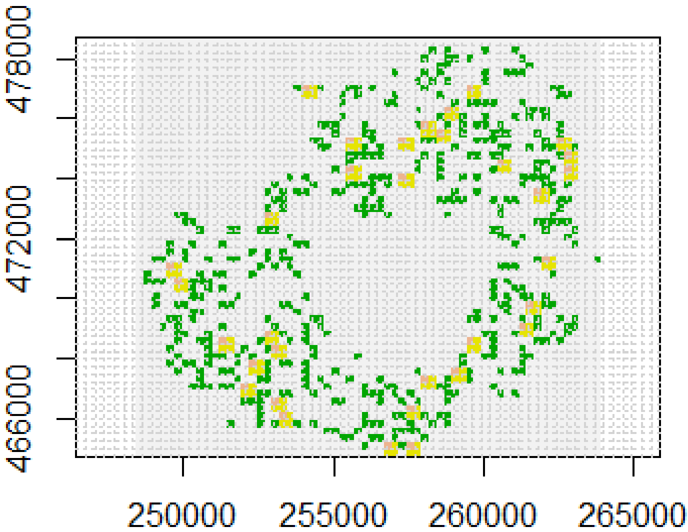
```
solution_coord2_cent <- solution_coord2
for (s in 1:obj.value2) {
  solution_coord2_cent[s,'x'] <- solution_coord2_cent[s,'x']+125
  solution_coord2_cent[s,'y'] <- solution_coord2_cent[s,'y']-125
}
solution_coord2_cent #coordinates of Vestas turbines
```

```
##              x        y
##  [1,] 249658.6 470967.4
##  [2,] 249908.6 470467.4
##  [3,] 251408.6 468467.4
##  [4,] 252158.6 466967.4
##  [5,] 252408.6 467717.4
##  [6,] 252908.6 472717.4
##  [7,] 252908.6 468717.4
##  [8,] 253158.6 468217.4
##  [9,] 253158.6 466467.4
## [10,] 253408.6 465967.4
## [11,] 254158.6 476967.4
## [12,] 255658.6 475217.4
## [13,] 255658.6 474217.4
## [14,] 256908.6 464967.4
## [15,] 257408.6 475217.4
## [16,] 257408.6 473967.4
## [17,] 257658.6 466217.4
## [18,] 257658.6 464967.4
## [19,] 258158.6 475717.4
## [20,] 258158.6 467217.4
## [21,] 258658.6 475467.4
## [22,] 258908.6 476217.4
## [23,] 259158.6 467467.4
## [24,] 259658.6 476967.4
## [25,] 259658.6 468467.4
## [26,] 260658.6 474467.4
## [27,] 261408.6 468967.4
## [28,] 261658.6 469717.4
## [29,] 261908.6 473467.4
## [30,] 262158.6 471217.4
## [31,] 262658.6 475217.4
## [32,] 262908.6 474717.4
## [33,] 262908.6 474217.4
```

```
points(solution_coord2_cent, col="mediumblue", bg="mediumblue", pch=21, cex=2)
```




```
# Third iteration - Enercon turbines

#Preprocessing

solution_rowcol2 <- rowColFromCell(qgis, solution_cellno2)
solution_neigh_row_pos2 <- matrix (0, nrow=4, ncol=obj.value2 )
solution_neigh_col_pos2 <- matrix (0, nrow=4, ncol=obj.value2 )
for (k in 1:obj.value2) {
  solution_neigh_row_pos2 [,k] <-  c(solution_rowcol2[k,1], solution_rowcol2[k,1],sol
ution_rowcol2[k,1]+1,solution_rowcol2[k,1]+1)
  solution_neigh_col_pos2 [,k]<-  c(solution_rowcol2[k,2],solution_rowcol2[k,2]+1, so
lution_rowcol2[k,2]+1, solution_rowcol2[k,2])
}
solution_neighbourhoodcellnumber2 <- matrix(0, nrow=4, ncol=obj.value2 )
for (k in 1:obj.value2) {
  solution_neighbourhoodcellnumber2[,k] <- cellFromRowCol(qgis, solution_neigh_row_po
s2[,k], solution_neigh_col_pos2[,k])
}
A [solution_neighbourhoodcellnumber2] <- 0.5
A [solution_cellno2] <- 0.2values(qgis) <- A
plot(qgis)
grid(nx=ncol(qgis), ny=nrow(qgis))
```

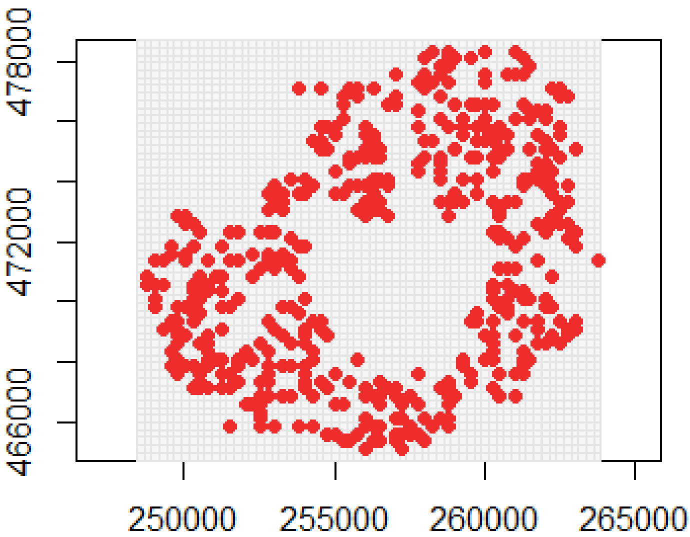



```
A [solution_neighbourhoodcellnumber2] <- 0
values(qgis) <- A
```

```
matrix3 <- as.matrix(qgis)
inds_posible3 <- which(matrix3==1, arr.ind = TRUE)
count_posible3 <- NROW(inds_posible3)
count_posible3
[1] 445


options_cellnumber3 <- cellFromRowCol(qgis, inds_posible3[,1], inds_posible3[,2])
plotcoord3 <- coordqgis[options_cellnumber3,]
plotcoord3
            x        y

  [1,] 248783.6 470842.4

  [2,] 248783.6 470592.4

  [3,] 249033.6 471342.4

  [4,] 249033.6 470592.4

  [5,] 249033.6 470092.4

  [6,] 249033.6 469842.4

  [7,] 249283.6 471342.4

  [8,] 249283.6 470592.4

  [9,] 249283.6 469092.4

 [10,] 249533.6 471842.4

 [11,] 249533.6 471592.4

 [12,] 249533.6 469342.4

 [13,] 249533.6 468342.4

 [14,] 249533.6 467842.4

 [15,] 249783.6 472842.4

 [16,] 249783.6 469842.4

 [17,] 249783.6 469342.4

 [18,] 249783.6 469092.4

 [19,] 249783.6 468842.4

 [20,] 249783.6 468592.4

 [21,] 249783.6 468092.4

 [22,] 249783.6 467592.4

 [23,] 250033.6 472842.4

 [24,] 250033.6 472592.4

 [25,] 250033.6 471592.4

 [26,] 250033.6 471342.4

 [27,] 250033.6 469842.4

 [28,] 250033.6 468842.4
```

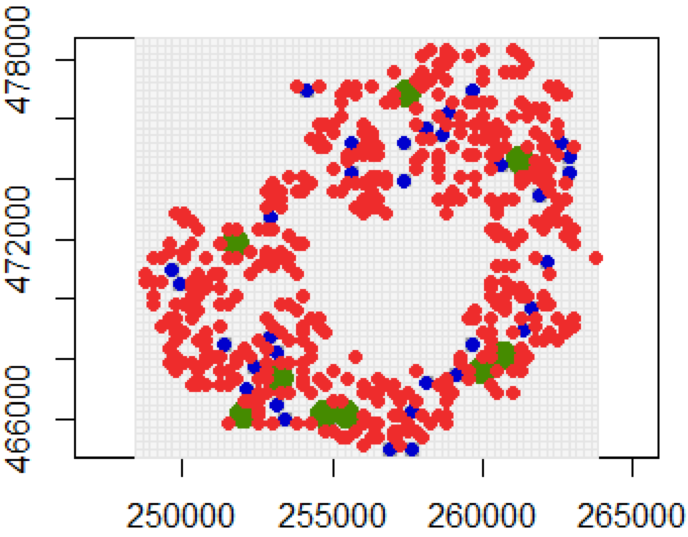
```
 [29,] 250033.6 467842.4
 [30,] 250283.6 472592.4
 [31,] 250283.6 471842.4
 [32,] 250283.6 470592.4
 [33,] 250283.6 470342.4
 [34,] 250283.6 470092.4
 [35,] 250283.6 469592.4
 [36,] 250283.6 469342.4
 [37,] 250283.6 467842.4
 [38,] 250283.6 467092.4
 [39,] 250533.6 472342.4
 [40,] 250533.6 470842.4
 [41,] 250533.6 470592.4
.
.
.
[444,] 263033.6 469092.4
[445,] 263783.6 471342.4
```

```
points(plotcoord3, col="firebrick2", bg="firebrick2", pch=21, cex=1)
```




```
obj.value3 <- count_posible3
obj.value3

## [1] 445

#Calculate CoE parameter values
```

```r
COE <- function(Ne, Nv, Ns) {
  Ce <- Ne*prE*1000*(2/3+1/3*exp(-0.00174*Ne^2))
  Ee <- Ne*aepE
  Cv <- Nv*prV*1000*(2/3+1/3*exp(-0.00174*Nv^2))
  Ev <- Nv*aepV
  Cs <- Ns*prS*1000*(2/3+1/3*exp(-0.00174*Ns^2))
  Es <- Ns*aepS
  CoE <- (Ce+Cv+Cs)/(Ee+Ev+Es)
  return(list(Ne, Nv, Ns, Ce, Cv,Cs, Ee, Ev, Es, CoE))
}
list_coe <- COE(obj.value3,obj.value2,obj.value)
list_coe

## [[1]]
## [1] 445
##
## [[2]]
## [1] 33
##
## [[3]]
## [1] 9
##
## [[4]]
## [1] 237333333
##
## [[5]]
## [1] 47307466
##
## [[6]]
## [1] 27107716
##
## [[7]]
## [1] 1102503685
##
## [[8]]
## [1] 248531483
##
## [[9]]
## [1] 129273112
##
## [[10]]
## [1] 0.210597

# Plot area with all turbine locations

qgis <- enschede1
plot(qgis, breaks=c(0,0.2,1),col=c("gray96", "gray61", "gray61"), legend=FALSE)
plot(rasterToPolygons(qgis), add=TRUE, border='gray90', lwd=1)
points(solution_coord, col="chartreuse4", bg="chartreuse4", pch=21, cex=2)
points(solution_coord2_cent, col="mediumblue", bg="mediumblue", pch=21, cex=1)
points(plotcoord3, col="firebrick2", bg="firebrick2", pch=21, cex=1)
```




```
write.csv(solution_coord, file = "Siemens.csv")
write.csv(solution_coord2_cent, file = "Vestas.csv")
write.csv(plotcoord3, file = "Enercon.csv")
new <- Sys.time() - old
print(new)

## Time difference of 35.72827 secs
```