Cost optimization of an offshore wind farm integrated to an oil & gas platform

REBEKKA HAMRE

SUPERVISOR
Joao Leal

University of Agder, 2020
Faculty of Engineering and Science
Department of Engineering Sciences

Abstract

This thesis is a case study of a wind and gas hybrid system, based on data from an existing oil and gas (O&G) platform field with seven gas turbines connected to a fictitious wind farm for a monthly time period for six years. The thesis uses the Python programming language to implement a genetic algorithm (GA) to optimize three different scenarios. The first scenario is to find the optimal number of wind turbines to gain the minimum total cost while meeting the load demand. The second scenario is to find the optimum number of wind turbines that give the optimum total cost if the O&G industry should contribute to cut 50 % of the total CO_2 emissions by 2050 in Norway (gas turbines at O&G installations accounted for 84.6 % of the direct CO_2 emissions from platforms in 2018). The third scenario is to find the minimum total cost and number of wind turbines while meeting the load demand, assuming the LCOE for floating offshore wind decreases by 30 % in 2030. From the results obtained from the GA, it shows from scenario 1, that the optimum number of wind turbines that gives the lowest total cost is 10 wind turbines. For scenario 2, it shows that it is needed to implement at least 7 wind turbines to gain the desired CO_2 reduction. From scenario 3, it shows that if the LCOE for floating offshore wind decreases with 30 %, the total cost is 249.818 mill USD, which is less than the total cost without wind turbines. From the obtained results from the GA, it is clear that floating offshore wind with today's LCOE prices is an expensive investment, but if the LCOE prices in the future reduce the total cost can be even cheaper than the cost without the wind turbines. Nevertheless, wind energy is intermittent by nature and needs other energy sources or storage to cover days with no or low wind. In this study are all costs based on research and findings due to limited time and much confidential material. For further work, it is recommended to do all calculations, such as LCOE, based on real numbers and get load profile, and costs and emissions for the specific model of gas turbines used on the O&G platform. It is also recommended to obtain the data on a daily or hourly basis to achieve more accurate results. For the optimization, it is also recommended to test other methods to compare the results.

University of Agder, Grimstad, June 21st, 2020

Rebekka Hamre

Rebekka Hamre

Preface

This thesis is the concluding work of my education on the Renewable Energy Master's program at the University of Agder (UiA) in Grimstad, Norway. Previously I have a bachelor's degree in Renewable Energy at UiA.

The Master's thesis corresponds to 30 study credits, which is equivalent to one semester, lasting from January to June 2020. This thesis addresses an optimization using a GA implemented in Python to minimize the total cost and find an optimal number of wind turbines for three different scenarios of a study case with a focus on reducing CO_2 emissions and excess wind power, using wind and load data for a monthly period of 6 years.

My interest in offshore wind energy started when I first worked as a summer intern at Aker Solutions, where I got to work with their offshore wind team for three summers. My interest in offshore wind energy combined with the knowledge in the O&G Industry while working at Aker Solutions was my background for wanting to write about this topic in my thesis.

When I first started working on my thesis, I had a basic understanding of optimization from a previous course at UiA, ENE409-G, Design and Optimization of Heat and Power Systems. I also had limited skills in programming, and I spent many hours learning the programming language Python.

The period working with the Master's thesis has been remarkably interesting due to the current situation with the COVID-19, which has led to online supervising, and the motivation has not always been on top.

I wish to thank my supervisor Joao Leal, professor at UiA Grimstad, for all his guidance regarding my master thesis and being a great motivator in this period. I also wish to thank my family and friends for all the support and encouragement during the period.

Individual/group Mandatory Declaration

The individual student or group of students is responsible for the use of legal tools, guidelines for using these, and rules on source usage. The statement will make the students aware of their responsibilities and the consequences of cheating. The missing statement does not release students from their responsibility.

1.	I/We hereby declare that my/our report is my/our own work and that I/We have not used any other sources or have received any other help than mentioned in the thesis.	☒
2.	I/we further declare that this thesis: - has not been used for another exam at another department/university/university college in Norway or abroad; - does not refer to the work of others without it being stated; - does not refer to own previous work without it being stated; - have all the references given in the literature list; - is not a copy, duplicate, or copy of another's work or manuscript.	☒
3.	I/we am/are aware that violation of the above is regarded as cheating and may result in cancellation of exams and exclusion from universities and colleges in Norway, see Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen §§ 31.	☒
4.	I/we am/are aware that all submitted theses may be checked for plagiarism.	☒
5.	I/we am/are aware that the University of Agder will deal with all cases where there is suspicion of cheating according to the university's guidelines for dealing with cases of cheating.	☒
6.	I/we have incorporated the rules and guidelines in the use of sources and references on the library's web pages.	☒

Publishing Agreement

Authorization for electronic publishing of the thesis.

Author(s) have copyrights of the thesis. This means, among other things, the exclusive right to make the work available to the general public (Åndsverkloven. §2).

All thesis that fulfill the criteria will be registered and published in Brage Aura and on UiA's web pages with author's approval.

Thesis that are not public or are confidential will not be published.

I hereby give the University of Agder a free right to

make the task available for electronic publishing: ☒JA ☐NEI

Is the thesis confidential? ☐JA ☒NEI

(confidential agreement must be completed and signed by the Head of the Department)

- If yes:

Can the thesis be published when the confidentiality period is over? ☐JA ☐NEI

Is the task except for public disclosure? ☐JA ☐NEI

(contains confidential information. see Offl. §13/Fvl. §13)

Abbreviations

AC	Ant colony
CH_4	Methane
CO_2	Carbon dioxide
DP	Dynamic programming
EU ETS	European Emissions Trading System
GA	Genetic algorithm
GHG	Greenhouse gases
IEA	International Energy Agency
IRENA	International Renewable Energy Agency
LCOE	Levelized Cost of Energy
LP	Linear programming
MET	(Norwegian) Meteorological institute
Mill	Million
NCS	Norwegian Continental Shelf
NMVOCs	Non-methane volatile organic compounds
NO_x	Nitrogen Oxide
NLP	Nonlinear programming
O&G	Oil & Gas
PSO	Particle swarm optimization
PV	Photovoltaic
SA	Simulated annealing
Sm^3	Standard cubic meters
SO_2	Sulfur dioxide
UiA	Universitetet i Agder (University of Agder)
USD	U.S. Dollar

List of Figures

List of Tables

Table of Contents

1 Introduction

For Norway to meet its sustainability goals to reduce emissions by at least 50 % by 2030 (compared to 1990 level), it becomes necessary to replace more fossil fuel energy sources with renewables [1]. From 1990 to 2018, the emissions from O&G recovery increased from 8.2 mill tons yearly to 14.2 mill tons annually, respectively, the O&G industry's share of total emissions has thus increased from 15.9 % to 27.2 % in the period [2]. As the O&G industry is the main responsible for greenhouse gas emissions in Norway, especially from the burning of gas in the gas turbines, it is necessary to find solutions to reduce their emissions to meet Norway's sustainability goals.

Several O&G companies have started to invest in the renewable energy sector. An example of this is the energy company Equinor, who is going to power offshore O&G platforms with floating offshore wind turbines off the coast in Norway scheduled to start producing electricity at the end of 2020.

Using renewable wind power to reduce emissions and cost of carbon dioxide (CO_2) taxes at O&G installations, can look tempting, but is the price of implementing an offshore wind farm profitable? This study presents a case study of using a wind turbine farm as a supplement to an existing offshore O&G field.

In this research, wind speed and production data were retrieved from an O&G installation located in the North Sea. The study aimed to find the optimum total cost of a hybrid system consisting of offshore wind and gas turbines. For finding the optimum solution, a genetic algorithm (GA) was implemented.

GA is inspired by Charles Darwin's theory, where the fittest individuals survive and reproduce to the next generation. The algorithm is a proficient method for solving complex optimization problems fast.

The study is structured as follows: Chapter 2 provides a theoretical background. Chapter 3 provides the methodology used for each step of the progress on the model; Chapter 4 provides and discusses the results from the optimization. Chapter 5 concludes the work; Chapter 6 recommends further work.

1.1 Problem Definition

The main objective of this thesis is to minimize the total cost for a hybrid renewable and gas system, while always meeting the load demand.

The hybrid system consists of seven gas turbines and a varied number of 8 MW offshore wind turbines. Three different scenarios for the hybrid system is studied. To do the optimization of the system, a GA is implemented.

The goal of the thesis is divided into the following:

- Gain knowledge in offshore wind, O&G platforms, hybrid systems, optimization methods, and genetic algorithms.
- Gain knowledge in what methods have been used previously for optimizing hybrid systems by studying previous work.

- Implement a genetic algorithm for optimizing of the hybrid system.
- Find the number of wind turbines to gain the minimum total cost while meeting the load demand.
- Find how many wind turbines should be implemented if O&G contributes to cut 50 % of the CO_2. Thereby, find out how O&G installations can cut direct CO_2 emissions from O&G installations (gas turbines accounted for 84.6 % of the direct CO_2 emissions from O&G in 2018).
- Find the minimum total cost and number of offshore wind turbines, while meeting the load demand, assuming the LCOE for offshore wind decreases with 30 % in 2030.
- Evaluate the results obtained from the GA.

1.2 Limitations and Assumptions

Several limitations and assumptions were made in this thesis in order to achieve the desired goals. The method chapter provides a more in-depth explanation of the limitations and assumptions, while this section introduces the main limitations and assumptions. The primary limitations and assumptions are as following:

- The case study is based on a monthly time period for six years, so the results can only be seen as a rough estimate. For more accurate results, a daily or hourly time interval would be preferred, and an intraday analysis should have been implemented to address the high variability of the wind resources and its impact on always meeting the load.

- Due to the load profile from the O&G platform is confidential and it was not able to access in the amount of time for the thesis, the load profile is based on the O&G production from the platform and the average generation from seven gas turbines.

- The power from the wind by the wind turbine is based on the power curve, which is provided by the manufacturer. The power curve is created under standard conditions and might not be 100 % representative of this scenario. Nor does the wind power consider losses or downtime. It is also used average wind speeds for each hour, using average can smooth out fluctuations that can affect wind energy production.

- LCOE for the offshore wind turbine is based on an already calculated LCOE from another report based on data from Spain. The calculation of LCOE needs investment cost, operation, and maintenance cost, discount rate, and other costs in the lifetime of the wind turbine, which is numbers that are hard to access in a limited amount of time, and without connection to someone who can share this type of information. The LCOE for the gas turbines is also based on average available numbers, and not for this specific scenario.

- The amount of CO_2 emissions per MWh from the gas turbines is based on an average from available numbers.

- The GA generates population randomly, so the results vary a bit for every simulation.

2 Theory

In this chapter, a brief explanation of the essential characteristics of wind, different components of a renewable hybrid system, the optimization process, and a literature review is presented.

2.1 Oil and gas platforms

An offshore O&G platform usually consists of facilities such as drilling, accommodation, processing, exporting, and injection. These are facilities that are very energy consuming, platforms at the Norwegian Continental Shelf (NCS) usually have a consumption ranging from 10 MW to several of GW [3].

In Norway, the production of O&G started in 1971 [4]. The petroleum industry is Norway's most important industry, and since the production began, it has given Norway a value creation of approximately 14 000 billion NOK, measured in today's currency [5]. Numbers from 2017 shows that the Norwegian petroleum sector, both directly and indirectly, employs 225 000 [6]. There is no doubt that the petroleum industry will continue to be an important industry for Norway. Several industries have started to look at other possible energy resources, and companies have gone from calling themselves for petroleum companies to energy companies. There is no doubt that the future will depend on energy from fossil resources, but that there will be an increased number of renewable sources.

2.1.1 Energy supply (load)

The load demand at O&G platforms is very high and requires good energy sources for covering it. Currently, the O&G platforms are supplied with energy from gas turbines, which are operated by combusting natural gas or diesel oil. The primary source of emissions to the air on the O&G platforms comes from burning the natural gas/diesel oil [7]. In 2018 were the gas turbines responsible for 84.6 % of the direct CO_2 and 68 % of the nitrogen oxide (NO_x) emissions directly to air from O&G installations [7]. Due to increased focus on reducing greenhouse gas (GHG) emissions, electrification by replacing parts of the fossil fuel power source with renewable energy on the O&G platforms is necessary. Electrification can be done by either lay cables and get power from shore or replace parts of the gas turbines with renewable energy power. Another solution is to implement carbon capture and storage.

According to the International Renewable Energy Agency (IRENA) roadmap to 2050, electricity will, by 2050, become a significant energy carrier, growing from a 20 % share of final consumption to around 50 % share in 2050, where renewable power supplies two-thirds of the final energy [8].

2.1.2 Gas turbines

The efficiency of the gas turbines varies with the different models. The efficiency of modern gas turbines is around 36 to 37 % at full load [9]. At reduced load, modern gas turbines have an efficiency of about 30 to 36 %, giving an average efficiency of around 33 %.

The case study in this thesis is based on GE's LM2500 PE combined cycle gas turbine. The LM2500 PE is an aero-derivative gas turbine, a derivative of the General Electric CF6 aircraft engine. The LM2500 series is specially designed to serve the O&G industry as it is a light and compact version for generating electricity. Aeroderivative gas turbines also have the advantage over other typical industrial machines that it is more flexible to handle changes in the load more and that it has a short shutdown time [10]. These types of gas turbines also have cleaner emissions and better heat rates compared to other gas turbines. The LM2500 models are also designed to achieve a low level of emissions and do not exceed 15 ppm NO_x emissions [11]. The LM2500 PE has a total capacity of 23 MW and an operation time of 4000 hours [12].

2.1.3 GHG Emissions

By replacing all or parts of carbon-emitting fuel-based energy sources with renewable energy sources, air pollutions such as CO_2 and other GHG can be reduced substantially. GHG emissions are the main contributor to the greenhouse effect, which again contributes to global warming.

Since most countries have goals to reduce environmental emissions to become more sustainable, there has been a significant focus on electrification with among other renewable energy. In Norway, around 80 % of GHG emissions are taxed and/or regulated through the European Emissions Trading System (EU ETS)[13]. The EU ETS limits the total emissions among the joint countries by setting a quota for the companies. If the companies exceed the given quota, the companies have to purchase quotas from other companies. Vica versa can the companies with surplus quota, sell their remaining quota to other companies. The EU ETS is annually reduced to achieve the emission target.

The petroleum sector has to pay both the EU ETS, which is around NOK 200/tons CO_2 and the Norwegian CO_2 taxes, which is currently around NOK 500/tons CO_2 [13]. The taxation rate for fossil energy in Norway is among one the highest in the world [14]. Norway also has integrated environmental and climate considerations in its petroleum industry policy [15].

The combustion of natural gas and diesel in the turbines at the O&G platforms is the primary polluter, resulting in emissions such as CO_2, NO_x, methane (CH_4), sulfur dioxide ($SO_{2)}$, and non-methane volatile organic compounds (NMVOCs). About a quarter of Norway's GHG emissions comes from petroleum activities, and in 2018 around 13.4 million tonnes CO_2 equivalent was emitted from petroleum activities [15].

2.2 Offshore wind power

Offshore wind technology has, over the last decade, developed rapidly poised to make a significant difference in future energy systems, growing almost 30 % per year between 2010 and 2018 [16]. In 2018, turbines with record-level rated capacities ranging from 10 MW to 12 MW had been announced by manufacturers to be available for commissioning after 2020 [17].

The advantage of implementing offshore wind power to O&G platforms is that the average wind speeds are high, the turbulence intensity and wind shear are low compared to onshore wind power [3]. The wind speeds near offshore platforms on the NCS an average wind speeds 9-11 m/s. As offshore platforms on the NCS usually are at water depths exceeding 60 m, the traditional fixed-bottom offshore wind installations are not economically attractive [16]. At these water depths, floating offshore wind foundations are suitable and can harness the untapped wind resources. The wind power from floating wind could, according to the International Energy Agency (IEA), in 2040 meet the world's total electricity demand 11 times [16].

2.2.1 Floating offshore wind

The most mature offshore foundations are the monopile and the jacket; these are both currently restricted to waters less than 60 meters deep. The water depth restriction is a significant limitation as the wind resources are more abundant in deeper waters. Vibrant water sites also have some of the markets with considerable wind power potential [18]. For deeper waters, floating foundations can be applied and has in recent years, been maturing fast, and there have been significant technological developments. By 2050 floating wind farms could potentially cover around 5-15 % of the world's offshore wind installed capacity, according to IRENA [18]. The three main tested concepts for floating foundations are spar-buoy, semi-submersible, and tension leg platforms. Figure 1 illustrates the floating foundations.

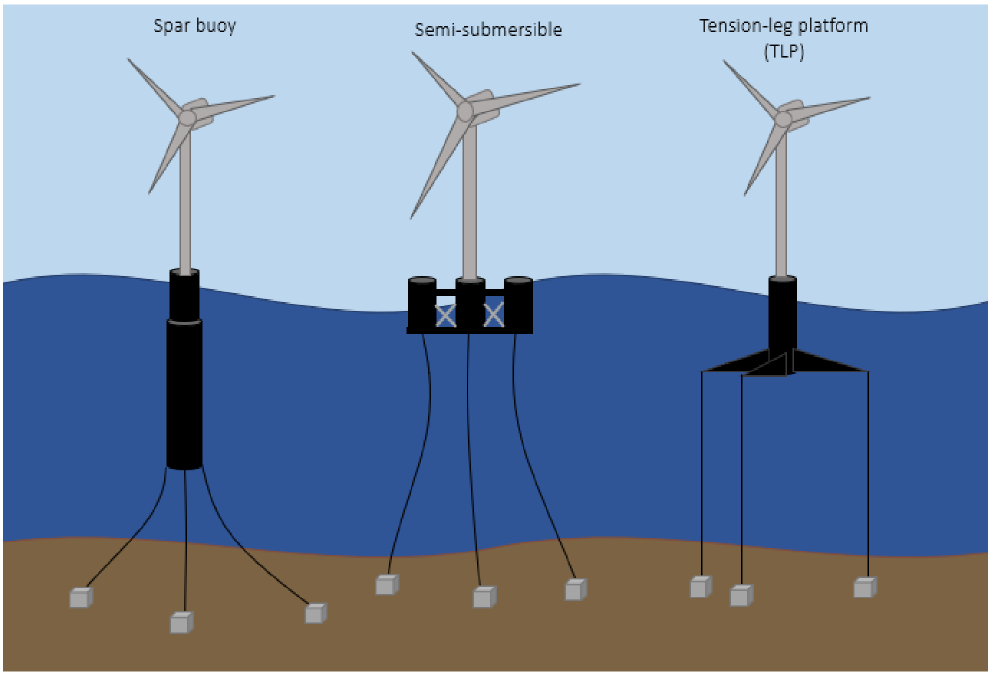

Figure 1: Floating wind foundation

Currently, there are mainly just demonstration and pre-commercial projects for floating offshore wind, but in 2017 Equinor's 30 MW Hywind project in Scotland commissioned being the world's first multi-unit installation of floating offshore wind. Equinor has also gotten approval for a 200 MW, and an 88 MW project. For the project off the coast of Norway, offshore wind turbines supplies power to five O&G platforms (Snorre and Gullfaks field) and accounts for around 35 % of the power supply [19]. Other demonstration projects are the 2 MW Floatgen in France and the 3 MW Hibiki in japan.

2.2.2 Wind Energy

Wind turbines create mechanical power by converting the kinetic energy in the wind. The kinetic energy in the wind is given as:

$$E = \frac{1}{2}mv^2 \qquad \textit{Eq. 1}$$

Where m is the mass [kg], v is the mean wind speed [m/s].

For a horizontal wind turbine, the mass is the contained volume that flows through the rotor (volume of a cylinder). For a wind turbine, the mass is air flowing; it is convenient to use mass per second ($\dot{m}$)[20], the kinetic energy per second ($\dot{E}$) is the same as the theoretical power P:

$$\dot{E} = P = \frac{1}{2}\rho Avv^2 = \frac{1}{2}\rho Av^3 \qquad \textit{Eq. 2}$$

Where $\dot{E}$ is the kinetic energy per second [J/s], P is the theoretical power of the wind [W], ρ is the density at a standard value of 1.25 [kg/m^3], A is the cross-section area [m^2], and v is the mean wind speed [m/s].

2.2.3 Power in the wind

The maximum theoretical amount of power that any wind rotor can extract is 59.3 % and is known as the Betz limit. The overall efficiency for most three-bladed rotors is around 50 % [20]. The maximum power extracted from the power available is also known as the power coefficient (c_p). The power extracted from the wind must, therefore, be factored in equation (for $\dot{E}$) and the power extracted from the wind is expressed as:

$$P = \frac{1}{2}\rho Av^3 c_p \qquad \textit{Eq. 3}$$

Where c_p is the power coefficient.

The power coefficient is not constant and varies with the tip speed ratio (λ) of the turbine.

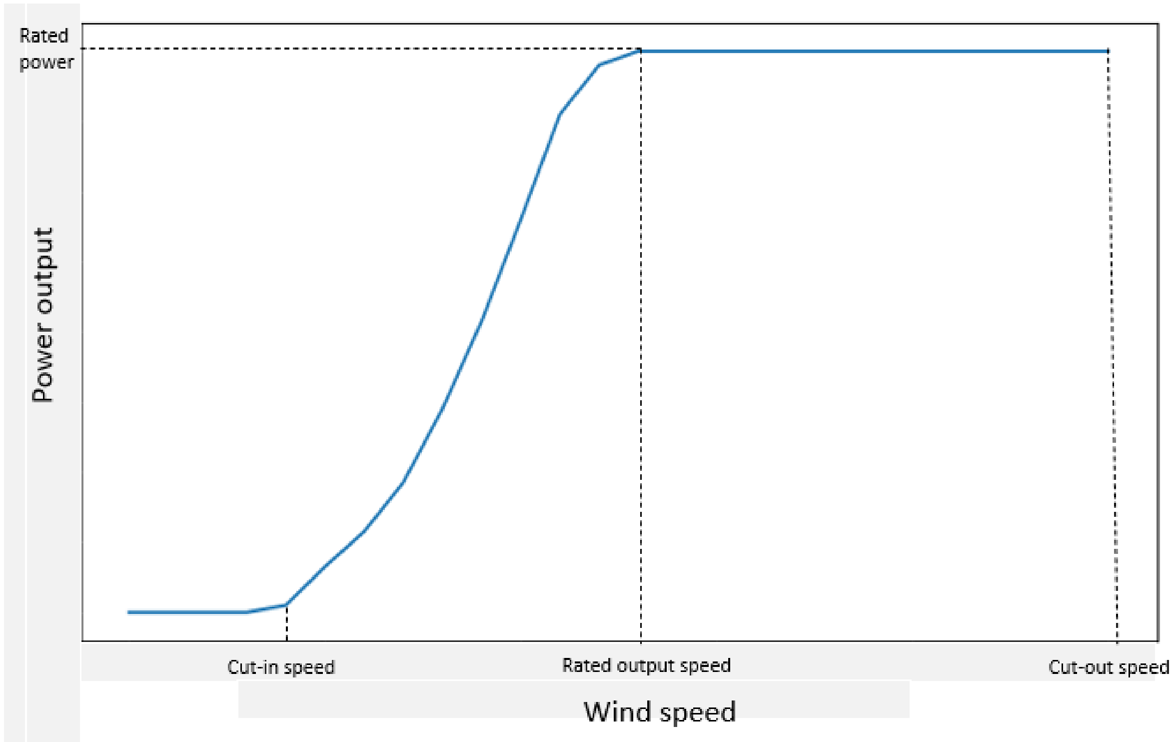

Figure 2: Typical wind turbine power curve

The electrical power output from the wind turbine from the different wind speeds can be shown in a power curve, see Figure 2. The three other parameters describe the power curve; cut-in speed, rated-speed, and cut-out speed. The cut-in speed typically ranges from 3 to 5 m/s; the rated speed typically ranges from 11.5 to 15 m/s, and the cut-out speed for most wind turbines generally is 25 m/s [20].

2.3 Cost of components

Experience together with the shift to deeper water, larger turbines, and better wind resources, the cost of offshore wind has, in the last decade, started to decline. For a wind turbine farm, the paramount price is the turbines, foundations, grid connection to shore, and the installation cost. The total installation cost for offshore wind projects commissioned from 2010-2018 has over the period had a 5 % cost reduction [21]. With wind farms moving from waters close to shore to deeper waters, the capacity factor for offshore wind increased up to 43 % in 2018, compared to 38 % in 2010 [18]. The wind resources at the deeper shore are often better and more stable, this together with the development of the wind turbine sizes, a better capacity factor has been achieved.

2.3.1 Levelized Cost of Energy (LCOE)

Levelized Cost of Energy (LCOE) refers to all the cost of generating electricity for a power plant's lifetime. The LCOE includes the cost of construction, operation and maintenance, fuel costs, taxes, and insurances. The LCOE is then found by dividing all the costs by the power plant's total lifetime energy production (MWh).

According to IRENA, offshore wind projects completed in 2018, had an average global capital cost of 4353 U.S dollar (USD)/kW with an average capacity factor of 43 % [16][22]. Capital investment (including costs of the turbine, foundation, internal cabling, substation, and offshore transmission assets) accounted for almost half of the LCOE for completed projects in 2018 [16]. Financing cost accounts for the other half. For the offshore wind in Europe, recent strike prices have indicated that there will be cost reductions in the nearest future. According to IEA, the new offshore wind turbines with rated capacities ranging from 10-12 MW planned commissioned after 2020 are to be auctioned at bids ranging from 55-75 USD/MWh [16][17]. IRENA has also predicted that the average installation cost for offshore wind projects could drop to between 1700-3200 USD/kW by 2030 and between 1400-2800 USD/kW by 2050 [18]. The average LCOE for offshore wind is around 0.115-0.127 USD/kWh [23][24][18]. IRENA has predicted that the average LCOE of offshore wind drops to an average between 0.05-0.09 USD/kWh by 2030 and 0.03-0.07 USD/kWh by 2050 [18][25].

The LCOE of floating offshore wind is higher than the LCOE for offshore wind, as it is a relatively new technology. A study done by Anders Myhr *et al.* [26] has analyzed and compared the LCOE of different floating wind turbine concepts. The study found that the LCOE for floating offshore wind was very dependent on the distance to shore and the depth at the chosen location. For wind farms with a longer length to shore, longer export cables are needed, and for wind farms at deep waters, the mooring costs increase. The study found that for an optimal site, the LCOE for floating offshore wind could range between 82.0-236.7 Euro/MWh, which is equal to 109.06-314.811 USD/MWh. Another study done by Laura Castro-Santos *et al*. [27] has also analyzed the economic aspects of floating offshore wind for the location of Galicia (Spain). The study shows that the LCOE for floating offshore wind varies with the installed MW of wind turbines. From the study the LCOE for a 100 MW wind farm varies from 117.05 to 1282.4 €/MWh; 200 MW wind farm varies from 100.31-1012.7 €/MWh; 300 MW wind farm varies from 96.29 to 944.33 €/MWh; a 400 MW wind farm varies from 93.70 to 902.28 €/MWh; 500 MW wind farm varies from 92.69 to 765.82 €/MWh; and a 600 MW wind farm will vary from 91.74 to 867.73 €/MWh. As one can see from the results, the LCOE will decrease for the higher MW installed of power.

The Norwegian Water Resources and Energy Directorate (NVE) collects cost figures for different technologies and has calculated the average LCOE for several plants in Norway. NVE has made an interactive solution to get an overview of the LCOE for the other technologies. From the interactive solution, it is possible to look at the development from 2018 to 2040 and the LCOE for investment, operation, and fuel. For wind power, it is only LCOE prices for land-based as Norway only has one offshore wind turbine at sea per today. Table 1 shows the average LCOE for land-based wind power and gas-fired combined cycle plant in 2018. The average LCOE for land-based wind power in 2018 for an average wind farm of 506 MW is 0.3439 NOK/kWh. By 2040, the expected LCOE is 0.2132 NOK/kWh. The average investment cost for wind power is 10687 NOK/kW, the average fixed operating and maintenance costs are 0.0 NOK/kW/year, the average variable operating, and maintenance costs are 0.10 NOK/kWh. For the gas-fired combined cycle power plant, the average LCOE for 2018 was 0.8206 NOK/kWh with a size of 450 MW, and the average investment cost was 7365 NOK/kW. Comparing these two, the average LCOE for wind is much lower, while the average investment cost for wind is much higher.

Table 1: Average calculated LCOE of technologies obtained from the Norwegian market, source: NVE [28]

Technology	LCOE 2018 [NOK/kWh]	LCOE 2040 [NOK/kWh]	Investment cost [NOK/kW]	Fixed operating and maintenance costs [NOK/kW/year]	variable operating and maintenance costs [NOK/kWh]	Fuel Cost [NOK/kWh]
Wind power (land-based)	0.3439	0.2132	10687	0.0	0.100	0.0
gas-fired combined cycle power plant	0.8206	0.7796	7365	181	0.0210	0.6975

Depending on the size and model, and the amount of fuel burned, a gas-fired combined cycle power plant emits around 350-500 g CO_2/kWh electricity generated [29][30][31]. The petroleum sector must pay both the EU ETS price and the Norwegian CO_2 taxes, which is currently around NOK 200/tons CO_2 and NOK 500/ton CO_2, respectively. The CO_2 taxes and EU ETS gives a total of 700 NOK/tons CO2, which is equivalent to 84.848 USD/tons CO_2, using the exchange rate for USD in 2019. Combined cycle gas turbines have an average LCOE of 36 USD/MWh [23][24].

2.4 Hybrid systems

Offshore petroleum installations where gas turbines feed large loads directly, cannot be electrified by cables from shore, as this is too costly and difficult to implement [32]. One possible solution is to implement an offshore wind power park near the platform instead. The wind turbines are intermittent by nature and will, therefore, not alone be able to meet the energy needs; the gas turbines can consequently be used as a backup when the installed wind turbines are not able to meet the load demand.

When combining multiple sources of power generation or combined with storage, it is called a hybrid system. Linking wind- and gas turbines make it possible to save fuel by declutching and stopping the gas turbines when the wind turbines exceed the load demand. A disadvantage of implementing a hybrid wind system is the high initial cost of the wind turbines.

A hybrid system is often used in remote and isolated areas to provide a 100 % secure energy supply. The optimum solution for a hybrid system is dependent on the specific situation depending on different factors. Figure 3 illustrates an example of a hybrid wind and gas system. Examples of factors that are important for finding the optimum solution are existing generation assets, market structure, and transmission and distribution infrastructure [33].

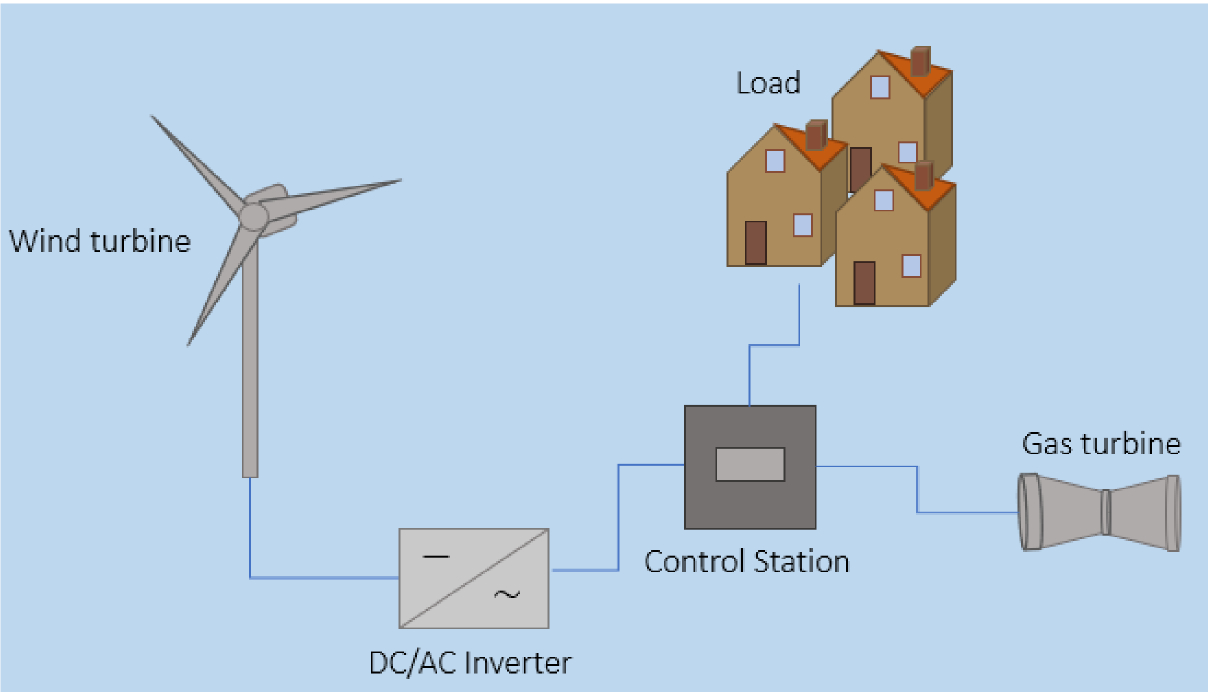

Figure 3: A hybrid wind-gas system

2.5 Optimization

An optimization problem consists of a group of variables, an objective function, constraints to limit the values of the variables and boundaries. An optimization problem is solved by finding the variables that result in the best (largest or smallest) value of the objective function within the given constraints. Solving an optimization problem requires to have an efficient optimization algorithm or an efficient search. There is a wide variety of algorithms. The algorithms are classified according to the goal of the optimization and characteristics of the algorithm.

2.5.1 Objective function

The task of the objective of an optimization problem is to minimize (or maximize) a numerical value. The value could be the cost of a project, an amount of a product, the weight of a product, etc. To be able to optimize the objective function, a vector minimizes (or maximizes) the value desired to be optimized. Generally, the following function describes an objective function:

$$maximize\ or\ minimize\ Z = \sum_{i=1}^{n} c_i X_i$$

Where c_i is the objective function coefficient equivalent to the ith variable and X_i is the ith decision variable. The coefficients of the objective function specify the value of the objective function of one unit of the corresponding variable, i.e., for cost optimization, c_i is the cost of using one unit of resource *i.* The decision variables are the variables that need to be determined to gain an optimized solution. The decision variable is the amount of resource *i* used to reach the optimized solution.

2.5.2 Decision variable

When formulating the optimization problem, establishing the decision variables are needed. The decision variables indicate what value the variables should have to gain the best result. Usually, the decision variables are presented as:

$$X_1, X_2, \dots, X_n$$

Where n is variables.

The decision variables are defined by setting upper and lower bounds, also known as the boundaries. It is essential to set reasonable boundaries to the decision variables so that the optimization algorithm does not waste time on finding solutions that are not useful.

2.5.3 Constraints

Constraints in an optimization problem define the values of the variables and represent the maximum or minimum of what the values can be. Usually, this function describes the constraints:

$$\sum_{i=1}^{n} a_{j,i} X_i \leq b_j \quad j = 1,2, \dots, m$$

Where j is an index corresponding to a constraint, $a_{j,i}$ is a coefficient that shows the amount of resource *j* is used for the unit of resource *i*, for the decision variable X_i. The constraint b_j is the remaining amount of resource j available. Constraints can also be written as greater-than-or-equal, as you can multiply by -1 and get less-than-or-equal constraint and vice versa. Constraints can also be written as inequalities, one greater-than-or-equal and a less-than-or-equal.

2.5.4 Genetic algorithm (GA)

A genetic algorithm is a widely used stochastic algorithm for solving optimization problems. Stochastic algorithms are referred to as metaheuristics. The word *heuristic* means *to find* or *to discover by trial and error*, and *meta* means *beyond* or *higher level* [34]. The trade-off in randomization and local search is apparent in all metaheuristic algorithms to a certain degree [34]. In 1960, John Holland introduced the algorithm based on Charles Darwin's theory of evolution, and in 1989 it was further developed by his student David E. Goldberg [34] [35]. The algorithm is based on the natural selection process where the fittest survive and reproduce.

Figure 4 shows the process of a GA. The optimization process using GA begins with generating an initial population. The initial population consists of individuals, which is defined by variables. In a GA optimization, the variables are commonly known as the genes. The genes function in the same way as for the human body; they form a string that together creates a chromosome. Gene can be seen as the recipe and chromosome as the book of recipes. In this context, the chromosome is the solution to the optimization. Usually, the strings forming the chromosome is in binary values. Figure 5 gives a simple picture of how genes, chromosomes, and populations are linked together.

The fittest individual is then selected by a fitness function that gives each individual a fitness score. The two with the best fitness score, also known as the parents, are then used to pass on the genes to the next generation.

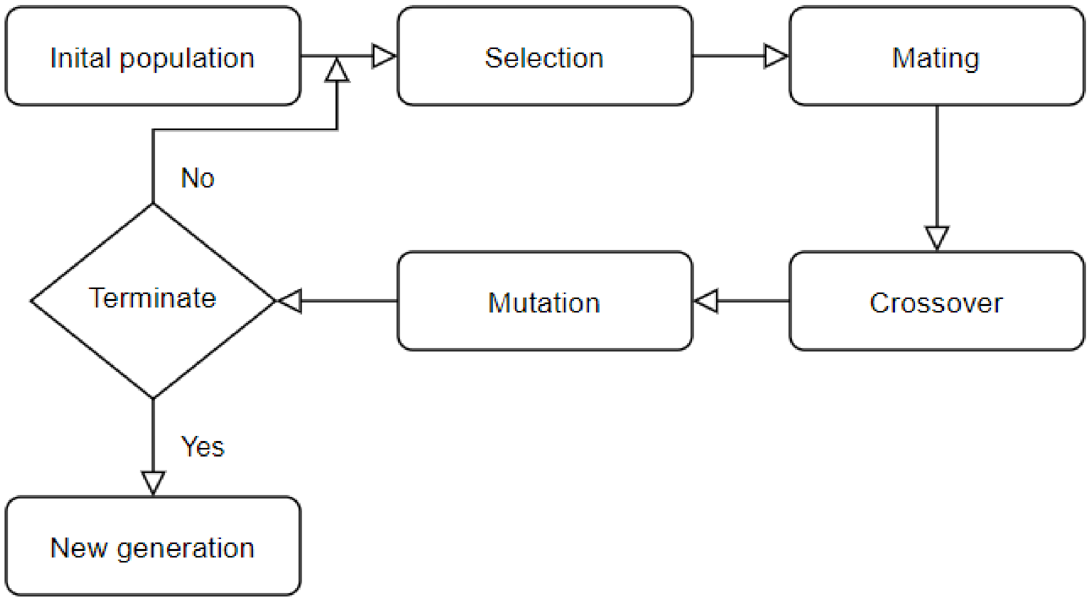

Figure 4: Flow chart showing the process of a genetic algorithm

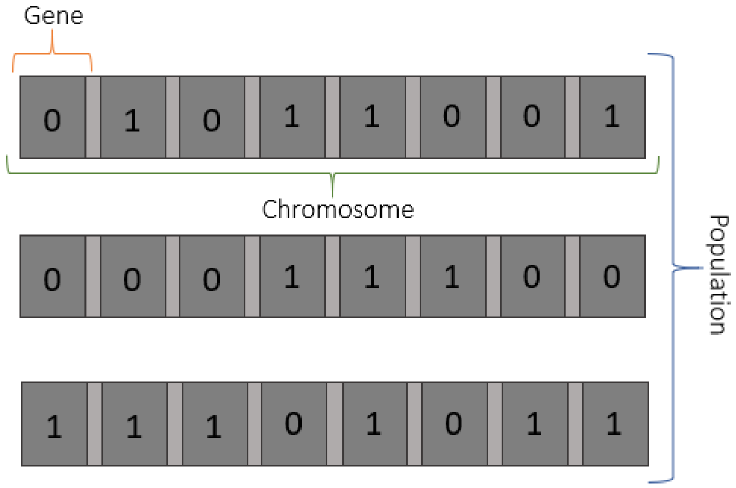

Figure 5: Initial population of a genetic algorithm

The algorithm then chooses a crossover point in the string of the genes from the parents. The next generation is formed by the mix of the genes from the parents within the crossover point, see Figure 6.

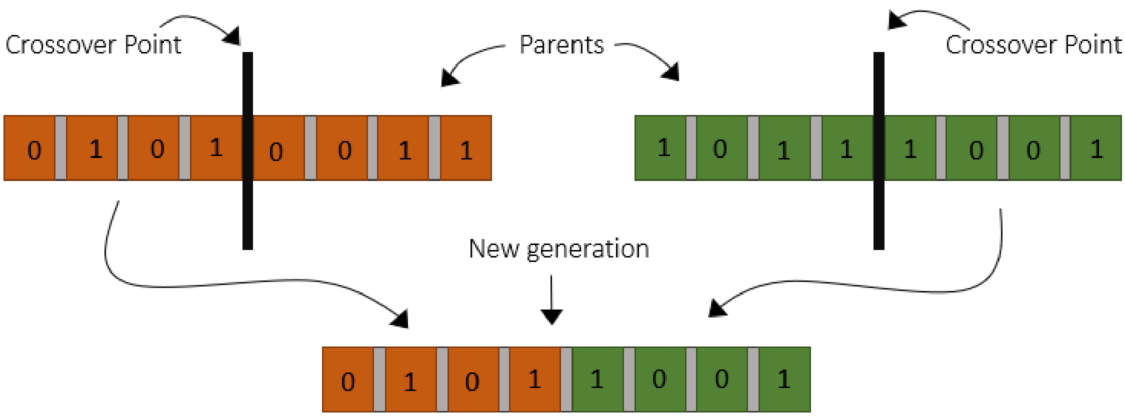

Figure 6: How genes are transmitted from parents to the next generation in genetic algorithm

Mutations with a low random probability can occur in the genes of the new offspring; this is to create variation in the population. If the next generation has converged, the algorithm should be terminated, and the optimization problem has gained a solution.

Optimization using GA has many advantages. The most useful advantages are the algorithm's ability to handle very complex problems fast, and its ability to run in parallel [36]. That the algorithm runs in parallel means that the algorithms are not dependent on each other, running the algorithm in parallel can avoid it being stuck in a local optimal solution. Optimization using GA does also have some disadvantages. One disadvantage of GA is that it might not give the most local optimum [36].

The reason why the GA does not always give the local optimum is due to the population is generated each time randomly, but compared to the time it takes to compute all possible outcomes, it is a proficient method.

2.6 Previous studies

Several studies have been investigating the hybrid system optimization matter. The most widely used methods for the optimization of hybrid systems are classical techniques and metaheuristic techniques [37]. Classical optimization algorithms find the optimum solution by utilizing differential calculus [37]. Examples of classical optimization are linear programming (LP), dynamic programming (DP), and nonlinear programming (NLP). Metaheuristic optimization is nature-inspired and usually uses stochastic operations in the search process [38][37]. Examples of metaheuristic optimization methods are generic algorithm (GA), particle swarm optimization (PSO), simulated annealing (SA), and ant colony (AC) algorithm.

Linear programming has been used for several hybrid renewable system optimization studies, as this method has the availability to perform reliability and economic analysis. An example of studies that have used this method is the study done by Saif *et al.* [39]. The study is a multi-objective capacity planning of a photovoltaic (PV)-Wind-Diesel-battery Hybrid power system. It has two objectives: minimizing the total cost and minimizing total CO_2 emissions while limiting the expected unserved energy. The study is based on a case study that involves designing a hybrid power system for a city with 50,000 inhabitants. The data used in the model were extracted from the real environment and technical data. The results from the study were used to construct a Pareto front, which represented the best trade-off between cost and emissions under different reliability conditions.

Another study that has used linear programming for optimization is the study done by Nagabhushana *et al.*[40]. The study utilizes linear programming to minimize the present worth of capital and operating costs of a hybrid renewable energy system by using MATLAB software. The constraints used in the study are the energy demand, components of the system, technological options, cost of components, satisfying battery autonomy hours, and keeping a limit on renewable energy resources. This has been done to find the optimum generation capacity and storage needs for a stand-alone hybrid wind/PV system for three specific locations in Karnataka (India).

The nonlinear programming model is when both the objective functions and the constraints or one of them contain nonlinear part. An example of a study that has used NPL for the optimization of hybrid systems is the study done by Ashok [41]. The study investigates the optimal combination of energy components for a typical rural community with the objective is to minimize the life cycle costs while having a reliable system operation. The study is based on a case study of a typical farming village of Western Ghats in Kerala, India. The optimization model that is developed in the study has used the hourly load profile of the town and found the optimal number of different renewable energy units and the optimal schedule by using a quasi-newton algorithm. The three other renewable sources considered are micro-hydro, wind, and solar PV.

A genetic algorithm is an evolutionary population-based algorithm to find the optimal solution for a given problem [37]. The genetic algorithm is implemented to a variety of different programming tools, such as MATLAB, Python, and java. Several additional studies have used this method for the optimization of hybrid systems. An example of a study that has used GA for optimization is the study published by Riboldi *et al*. [42]. The study investigates the possibilities to integrate a wind farm into an offshore combined cycle power plant. The study has used a multi-objective optimization with three objective functions; the cumulative CO_2 emissions, the total cost to supply energy to the plant, and the weight of the onsite power cycle. The study is based on existing wind turbine technologies and relative wind speeds. The wind speeds obtained were initially in a 20-minute resolution. It was desirable to increase the resolution to 1 minute. This was done by generating average wind speeds instances using a distribution function for 10 minutes variations. Increasing the resolution to 1 min was then accomplished using linear interpolation. It was desirable to find the optimal environmental and economic performances of the study case. The methodology used in the study was to summarize the annual CO_2 emissions of the plant's lifetime and meausure the total cost to supply energy to the plant. The results from the study showed that the installation of an offshore wind farm was economically challenging, but it was possible to reduce the CO_2 emissions.

Another study that has used GA for optimization is the study done by Bilala *et al.* [43]. The study studies a stand-alone PV-wind-diesel-battery system, where the objective is to minimize the Levelized cost of energy and CO_2 emissions by using GA. Solar radiation, wind speed, and temperature data used in the model are extracted from the specific site Gandon, Senegal. The obtained results were used to present an optimal Pareto front.

Another example of GA optimization has been done by Dufo-López *et al*. [44]. The study has applied a strength Pareto evolutionary algorithm to the multi-objective optimization of a stand-alone PV-wind-diesel system with battery storage. The objective of the study is to minimize the Levelized cost of energy and the equivalent carbon dioxide life cycle emissions.

Katsigiannis *et al*. [41] have in a study used GA to optimize a hybrid system consisting of solar PV, wind turbines, batteries, and diesel generators where the objective function is to minimize the system's cost of energy. The chosen constraints for the study are total initial cost at the beginning of the system's lifetime, the annual unmet load, the yearly capacity shortage fraction, fuel availability, minimum renewable fraction, and the size of each system's component. For the case study, the project's lifetime is assumed to be 25 years, and the meteorological data used is from the Chania region in Crete, Greece. The method used in the study determines the optimum configuration and the system's cost of energy.

Hongxing *et al*. [45] have utilized GA to find the minimized annualized cost of the system and an optimal model for designing a hybrid system consisting of PV array, wind turbine, battery bank, inverter, controller, and other accessory devices and cables. The study has modeled the individual components in the hybrid system. It is used hourly meteorological data on an annual basis from the year 1989 in Hong Kong as the studied project is close to Hong Kong, China.

Particle swarm optimization is an optimization based on the method a swarm moves to find food in a specific area [37]. This is an efficient method for solving problems within optimization and scattering. An example of studies that have used this method for the optimization of hybrid systems is, for example, the study done by Hakimi *et al*. [46]. In the study, it is found the optimal sizing of a wind-fuel cell hybrid. The study is based on the Khanoum site in south-east Iran. The aim is to minimize the total cost of the system by optimizing the capital costs, replacement costs, operation, and maintenance cost of all the components of the system, as well as efficiency, lifetime of components, and a lifetime of the project.

Another study using PSO for the optimization of hybrid renewable energy systems is the study done by Amer *et al*. [47]. The study has used the PSO algorithm to reduce the Levelized Cost of energy while taking into consideration the losses between production and demand sides. The algorithm structure in this study is built by using the MATLAB software. The study focuses on the cost of the equipment, the total cost of the installed project (including fixed financing costs), and the Levelized cost of energy.

Boonbumroong *et al.* [48] have used PSO to minimize the total life-cost of a stand-alone hybrid power system consisting of PV, wind turbines, and diesel. For reference, an existing hybrid power system has been used from Chik Island, Thailand. The simulated hybrid system results from the study were compared with the reference system. The general constraint chosen in this study is that the hourly energy demand must be satisfied by the energy generation units. Other studies that have used PSO to minimize the system cost is the study done by Bashir *et al.* [49] and Ardakani *et al*. [50].

A study done by Mohamed *et al.* [51] have modeled, optimized, and simulated a hybrid renewable system integrated with the smart grid using both PSO and GA. The objective function is to minimize the total present cost of the entire system and maximize system reliability. The system consists of wind turbines, diesel generator, main load, dummy load, PV array, and a battery bank. The proposed algorithm in the study ensures that the total energy generated from the hybrid renewable energy system satisfies the load requirements. If not, it should increase the size of the wind energy system or/and PV system by a particular value, and opposite. The study has used a year of hourly wind speed data, solar radiation, and ambient temperature for a couple of different sites. It is also used ten different types of wind turbines (onshore) from other manufacturers. For meeting the load requirements of the different locations, the study has also used a 10 % penetration ratio (ratio of wind generation to the total renewable generation). By using the minimum Levelized energy cost, the study was able to determine the optimum penetration. The software used for optimization in this study is also MATLAB. The study was able to find an optimum solution of site, wind turbine, the most effective penetration ratio, and PV area that gave the lowest total present cost. The study concluded with finding the optimum solution was quicker and more precise when using PSO compared to GA.

Other studies have investigated the possibility of operating an offshore wind farm in parallel with gas turbines at an offshore O&G rig. One example of studies is a study published by Korpås *et al*. [52]. The study is based on a case study of an offshore platform in the North Sea. The electricity demand for the case-study oil platform is supplied by two gas turbines (with one in backup, which is not being considered in this study). The gas turbines both have a capacity of 23 MW and share the load equally. The power consumption for the oil rig varies between 20 MW and 35 MW over the year. For the case study, 4 x 5 MW wind turbines with 4500 utilization hours are integrated. For the combined wind-gas power system, a quasi-steady-state MATLAB model was employed. The chosen time step size for the simulation was set to 1 minute, and the system was simulated for one year (2009). The study investigates two different scenarios: default operating and fuel saving.

Another similar study has been done by Aardal *et al*. [53]. This study investigates a system consisting of five interconnected oil platforms connected to a 100 MW offshore wind farm. The platforms are assumed to have a constant load of 147 MW in total. For backup, each of the platforms should have two gas turbines working in parallel, but the wind turbines should always have the priority to supply the load. The study has done a one-year simulation and then compared it with a simulation without wind turbines to find the overall reduction in fuel consumption. Then the gas savings yield revenues were used to the break-even cost for the wind farm investment. Another study done by He *et al.* [54] has done a case study of integrating an offshore wind farm with an offshore O&G platform and with an onshore electrical grid. The study has studied three different cases compromising wind farms rated at 20 MW, 100 MW, and 1000 MW with a focus on reducing CO_2 and NO_x emissions, electrical grid stability, and technical implementation feasibility. This study has also used data for one year with real load data from the platform.

From studying previous studies, it is clear that optimizing a hybrid system with offshore wind and gas is challenging but a relevant topic. To perform the task, GA seems to be a useful optimization model as it seems to be easy to implement, is sufficient for solving nonlinear complex engineering optimization problems.

3 Method

This chapter will show the methods used for the thesis. The method describes each step of the progress of the work on the thesis. First, the case study is presented along with the data utilized for the model, and then the optimization method is presented.

3.1 Case study description

The case study is based on an existing platform field connected to a fictitious wind farm, see Figure 7, where it is desirable to minimize the total LCOE of the system. The platform field consists of three oil platforms, in which two of the platforms together have seven GE gas turbines of the type LM2500 PE. The objective is to gain the optimum LCOE while meeting the load demand and reducing CO_2 emissions, and therefore the gas turbines are used as a backup for the wind turbines when there is little or no wind. The model is based on a monthly interval for six years (2012-2018). The number of wind turbines varies to find the optimal solution.

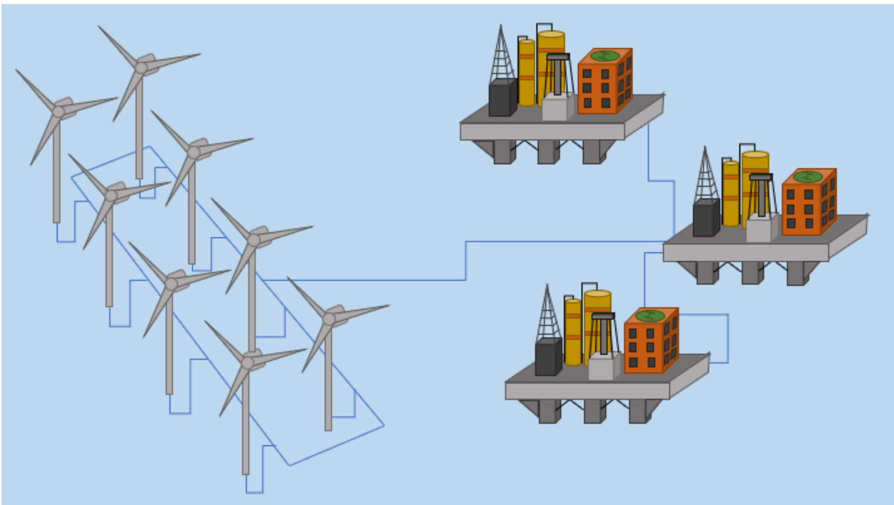

Figure 7: Illustration of the case study with a wind farm coupled with three O&G platforms

For the hybrid system, it is desirable to study three different scenarios. The scenarios are further explained in subsections 3.1.1, 3.1.2, and 3.1.3.

3.1.1 Scenario 1

The first scenario is to find the lowest LCOE and the number of wind turbines while meeting the load demand from the O&G platform.

3.1.2 Scenario 2

The second scenario is to find how many wind turbines it is needed to implement at the O&G platform if Norway wants to reduce their CO_2 emissions with 50 % within 2030 if gas turbines at O&G installations account for 86.4 % of the direct CO_2 emissions from platforms. For finding how many percent CO_2 emissions the O&G platforms need to reduce, the following calculation is applied:

$$(84.6\ \% \cdot x) + 15.4\ \% = 50\ \%$$

Where 84.6 % is the total amount of direct CO_2 emitted from O&G turbines, 15.4 % is the remaining CO_2 emissions, and 50 % is the desired CO_2 emissions reduction. Solving for x, the desired CO_2 emissions from the O&G platforms are found to be 34.6 %; this is a reduction of 59 %.

3.1.3 Scenario 3

In the third scenario, it is desired to find the minimum LCOE and number of offshore wind turbines, while meeting the load demand, assuming the LCOE for offshore wind will decrease with 30 % in 2030.

3.2 Wind and Load Data

The wind data used in this study is obtained from the Norwegian Meteorological Institute (MET) [55]. The Norwegian MET has historical data and real-time observations from stations both onshore and offshore. It is chosen to obtain historical data from the year 2012 to 2018. For this study, it is decided to use an 8 MW Vestas V164 wind turbine. The power curve for the Vestas V164 can be seen in Figure 8. The wind turbine has a cut-in speed at 4 m/s and a cut-out speed at 25 m/s. At 13 m/s, the wind turbine archives its rated power 8000 kW [56]. It is important to note that the power curve for the wind turbine is supplied by the manufacturer and is created under standard conditions and may not be 100 % representative for the chosen site. The data extracted is given with a time interval of 10 minutes. For simplicity, it is decided to look at a monthly time interval for the six years for the model. The mean wind speed per hour is therefore calculated and correlating power output from the power curve collected. This process is repeated for every sample MET has produced and summed up into days. For each day, the process completes, it sums up for a month. As it is only necessary to calculate this once, the data is saved as a separate CSV file.

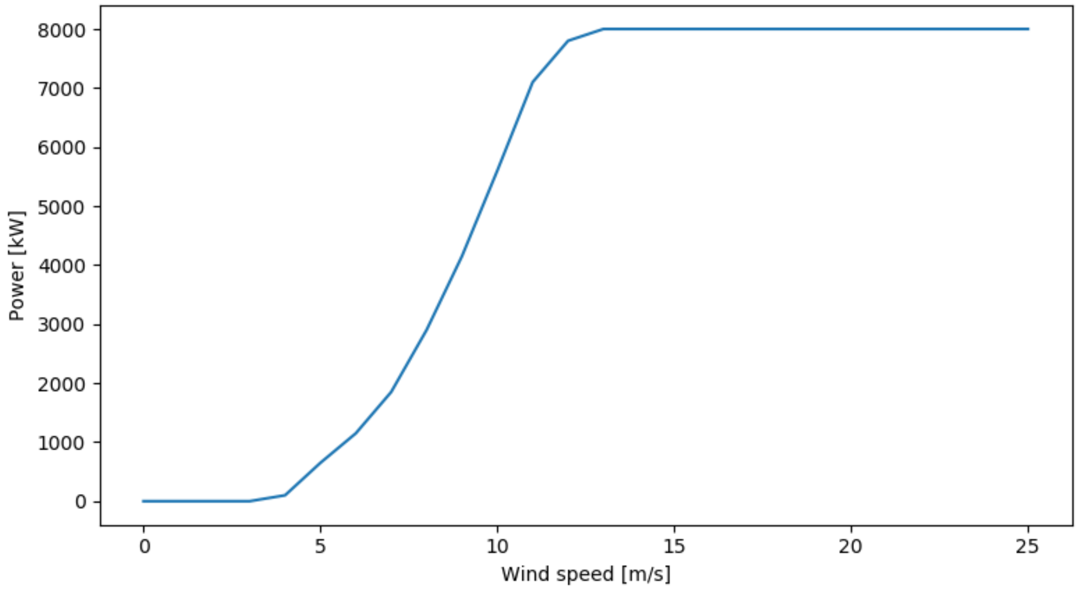

Figure 8: Power curve for Vestas V164-8.0 [56]

Assuming that there are no losses and downtime for the wind turbine(s), using the mean wind speed for each hour with the power curve for the Vestas wind turbine, the monthly theoretical wind power produced can be seen in Figure 10.

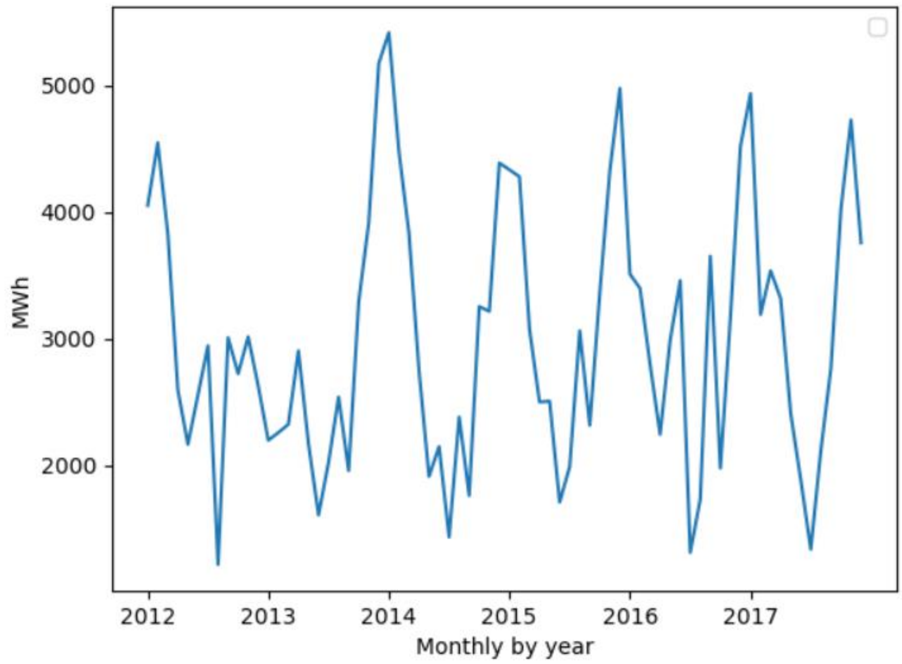

Figure 10: Monthly theoretical wind power produced from one 8MW Vestas V164 wind turbine

Due to the load demand profile for O&G platforms are confidential and hard to access, it is based on an average generation from seven LM2500 PE gas turbines and assumed that the load is correlated to the O&G production. The LM2500 PE gas turbines each have a total capacity of 23 MW, an operation time of 4000 hours, and an annual power generation of 56 000 MWh, which gives an average yearly generation of 392 000 MWh [57] [58]. The O&G production from the platform is obtained from the Norwegian Petroleum Directorate [59]. Figure 11 shows the monthly O&G production from 2012 to 2018.

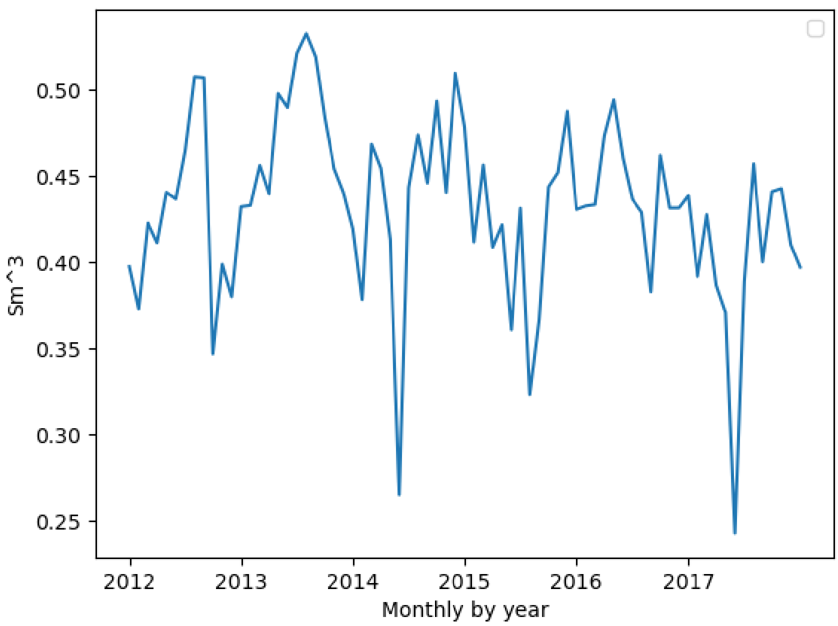

Figure 11: Monthly oil and gas production in Sm^3 from the platform from 2012 to 2018

If looking away from housing and other electricity need at the platforms, extracting oil and gas requires large amounts of energy, and it is assumed in this report that the load profile is correlated to the production of O&G. Eq. 4 shows the load factor for the platform.

$$Load\ factor = \frac{Average\ yearly\ generation}{produced\ oil\ and\ gas} \qquad \textit{Eq. 4}$$

Where the average yearly generation is in MWh, and the produced oil and gas is in millions (mill) standard cubic meters (Sm^3). It is used the average yearly O&G production from the platform from the year 2012-2018 to get a more accurate load factor. Multiplying the load factor with the monthly produced O&G then gives the average monthly load profile for the platform. Figure 12 shows the average monthly load profile for the platform from 2012 to 2018.

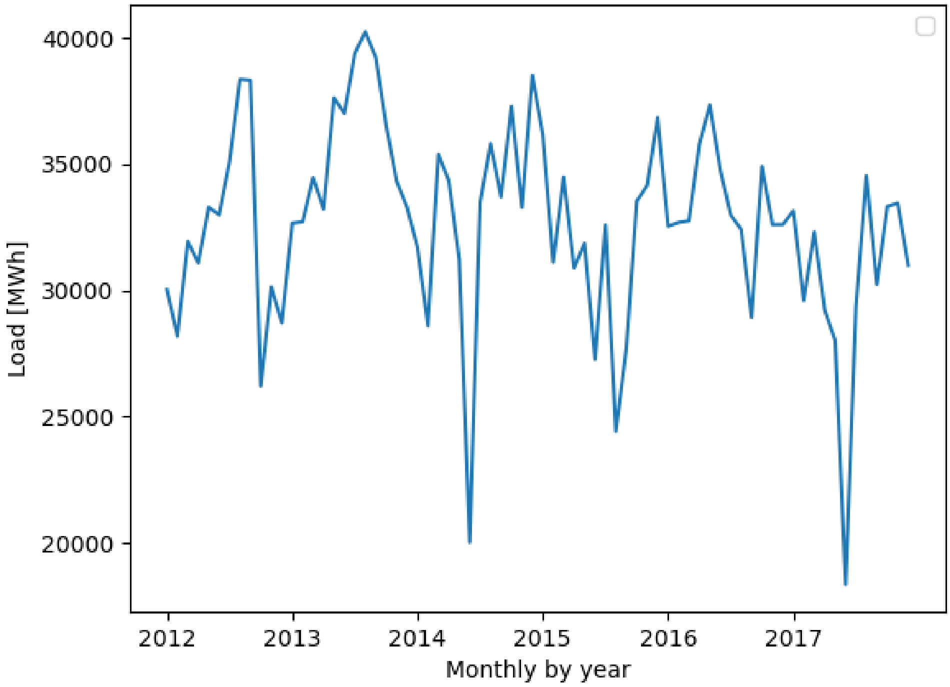

Figure 12: Load profile O&G platform from 2012-2018 assuming the oil and gas production correlates to the generation

The load profile data, as the wind power data, is saved as a separate CSV file to avoid running the script every time.

3.3 Cost of system components

The cost of the components is based on the average of numbers from newly published reports. For the floating offshore wind power, the LCOE is obtained from a previous study that has calculated the LCOE for floating offshore wind at the location Galicia (Spain)[27]. Using LCOE for floating offshore wind in Spain is a limitation as it will not be 100 % representative of this study with location off the coast in Norway, but it is assumed that the LCOE does not vary that much and probably be lower as for this case it is not needed grid connections to shore. LCOE is calculated with the assumption that all of the energy produced is used over its lifetime. When it is decided not to use this energy (exceeds load), the LCOE gets higher as the total cost of installation is the same, but the power used is lower. Therefore, the cost is the same but uses less power. That means the LCOE is raised when the excess power is not used. It is implemented a fix where it is also needed to pay for the excess MW/h; the LCOE stays the same, but the total cost increases as it also is needed to "pay" for the excess energy. According to IRENA, the LCOE for offshore wind is assumed to drop by around 30-60 % by the year 2030.

For the gas turbines, the LCOE is based on the average LCOE for gas-fired combined cycle power plant of the Norwegian market, which recent data shows the LCOE was 0.8206 NOK/kWh, which is with the exchange rate for 2018 equal to 0.10 USD/kWh. From the Norwegian petroleum directorate, the emission factor from gas turbines of used energy is 638 tons CO_2 emissions/GWh, and the price for EU ETS price and the Norwegian CO_2 taxes is 84.848 USD/tons CO_2. Using the emission factor and the taxes for emissions, it gives a cost of 54.133 USD/MWh for emissions from the gas turbines.

3.4 Energy cost optimization for offshore wind farm implemented to O&G platforms

For this research, it is desirable to find the optimum LCOE of the hybrid system. This section describes the implemented optimization algorithm.

3.4.1 Objective function

For the optimization process, it is essential to choose the appropriate objective function. In this case, the objective function is to minimize the total cost of the system based on the LCOE and the price for CO_2 emission taxes. The objective function is expressed as:

$$min\ Cost = \left(N_{wt} \cdot E_{WT}^{single}(t) \cdot C_{wt}\right) + \left(E_{gt}(t) \cdot C_{gt}\right) + \left(E_{gt}(t) \cdot C_{CO2}\right)$$

Where E_{wt} and E_{gt} is the monthly energy produced from one wind turbine and the gas turbines, respectively, in MWh. C_{wt} and C_{gt} is the LCOE for offshore (floating) wind turbines and gas turbines, respectively, in NOK/MWh. C_{CO2} is the cost for CO_2 emission taxes and N_{wt} is the number of wind turbines.

3.4.2 Decision variable

The decision variables are the number of wind turbines, N_{wt}.

$$x_1 = N_{wt}$$

And the known constants are the LCOE for wind turbines and gas turbines, C_{wt} and C_{gt} and the CO_2 emission taxes, C_{CO2}.

$$c_1 = C_{wt}$$

$$c_2 = C_{gt}$$
$$c_3 = C_{CO2}$$

3.4.3 Constraints

To make sure the system always meets the load demand, a constraint can be included in the calculation of the optimization problem. The sum of the power produced by the gas turbines and the wind turbines should not be smaller than the energy demand of the oil platform, $E_{load,op}(t)$, hence the constraint:

$$E_{load,op}(t) \leq E_{GT}(t) + E_{WT}(t) \cdot N_{wt}$$

Nor can the excessive wind power produced be used:

$$E_{WT}(t) = \begin{cases} E_{load,op}(t) if\ N_{wt} \cdot E_{wt}(t) > E_{load,op}(t) \\ N_{wt} \cdot E_{wt}(t) if\ N_{wt} \cdot E_{wt}(t) < E_{load,op}(t) \end{cases}$$

3.5 Genetic algorithm implementation

The programming and scripting language python were used to implement the model. External libraries were used in the model, pandas, and matplotlib [60] [61]. Pandas is used to import data from excel sheets, and matplotlib presents data visually. Wind data is extracted from METs excel with pandas and then sorted into three arrays. One array contains sample-date information while the second array holds corresponding wind speed values. The third array represents the Vestas power curve.

3.5.1 Initial Population

The first step of the GA optimization is generating an initial population, see Figure 13. The parameters for a population are the size of the population (number of individuals), lower and upper boundaries for how many wind turbines an individual may have. This is accomplished with the initial_population (pop_size, wturbines_bounds) function. The function returns an array with individuals (a population) where the number of turbines is set using the external random [src random] library.

Initial_population() is the first function to be called in the genetic algorithm script. Parameters are population size equal to ten, minimum turbine amount is one, and the maximum number of turbines set to 40.

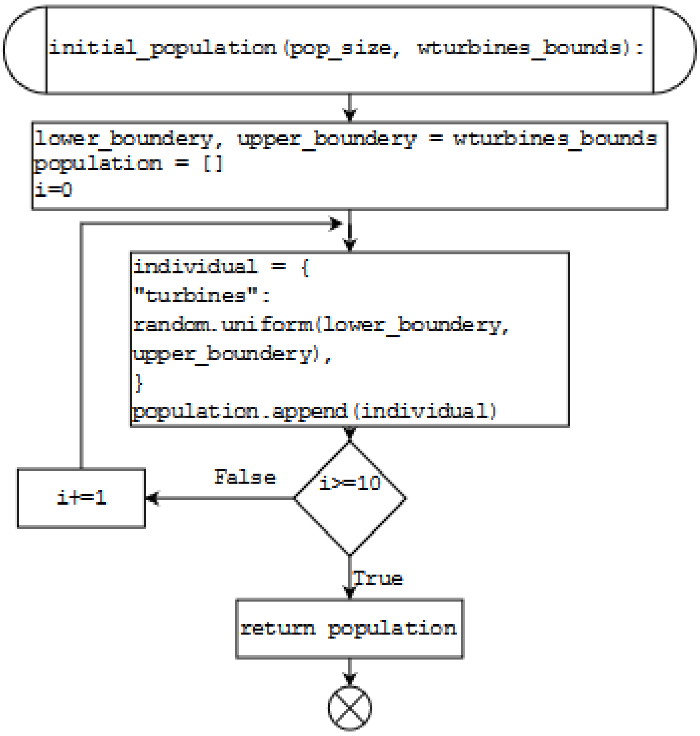

Figure 13: initial_population(pop_size, wturbine_bounds)

3.5.2 Fitness Function

The fitness function, see Figure 14, takes one parameter, a dictionary containing all the information about an individual: number of turbines and CO_2 waste reduction percentage. LCOE_wind function is called, which returns LCOE in relation to the number of MW installed. Before calculating fitness for the given individual, it checks whether an exact copy has been calculated before. If it has been calculated before, it will return that fitness instead of calculating it again.

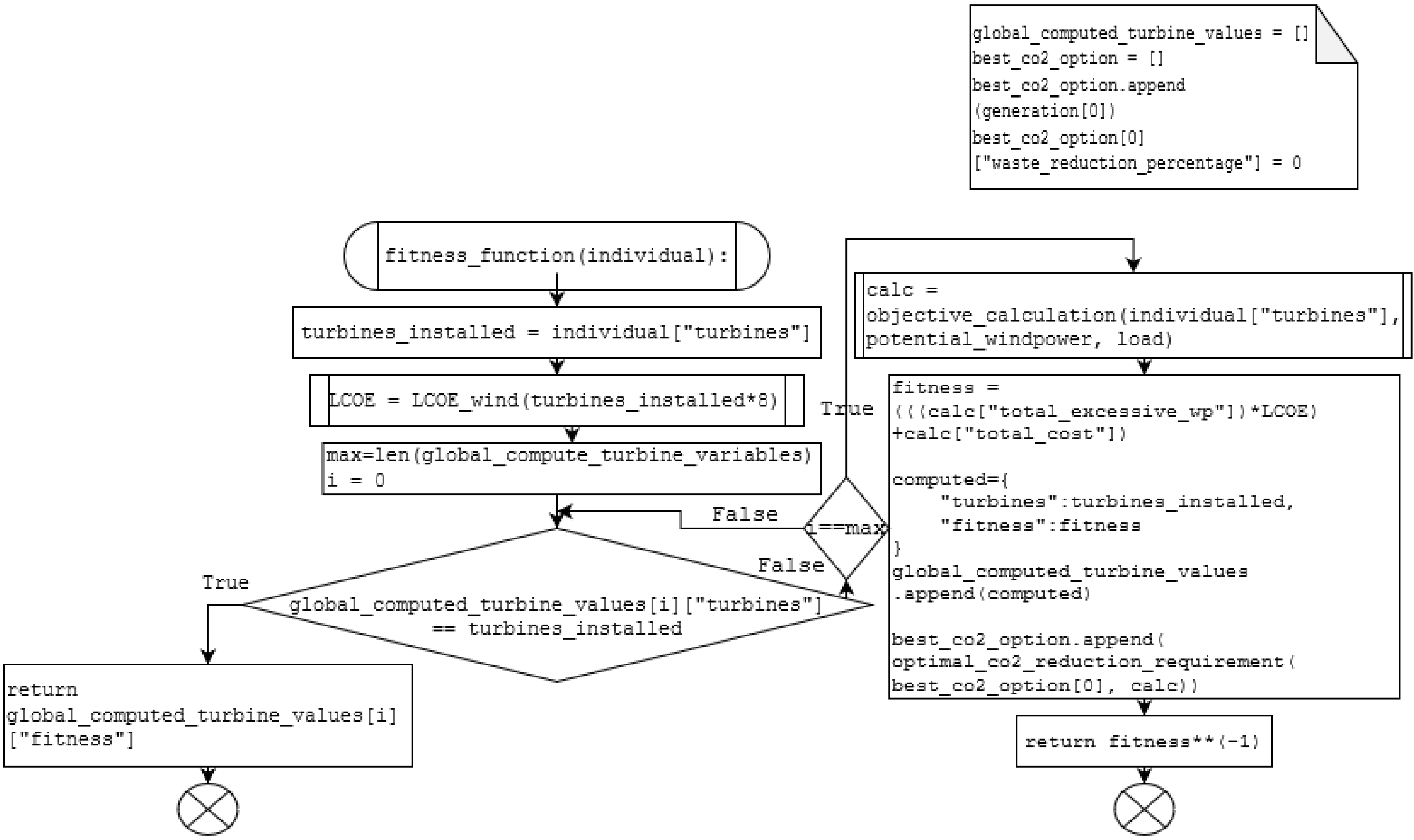

Figure 14: fitness_function(individual)

If the individual is unique, objective_calculation (num_of_winturbines, potential_windpower, load) is called. [obj_func expl], which returns how much excessive wind power there is and the cost for the current hybrid system, x number of wind turbines, and gas turbines. Fitness of the hybrid system is then calculated from a total cost perspective, wasted wind power included. This is because the LCOE is calculated with the assumption that all power is used. The fitness calculated for this individual is appended to an array, so it is not necessary to calculate for this exact number of turbines again. Finally, the fitness variable is elevated with minus one and returned. It is elevated with minus one, as it is desired to gain the best fitness to reflect the lowest cost.

3.5.3 Selection

Selection, see Figure 15, in this code, is structured so that individuals with better fitness have a higher probability of being selected for the mating process. A random float number between zero and one is generated using the random library [62]. Fitness is then calculated for the first individual in the generation, that individuals' probability for mating is determined by dividing fitness on the sum of all the individuals' fitness in the generation. If the selected individuals' probability is higher than the randomly generated floating number, the individual is selected for mating. Otherwise, the probability of that individual is added to an accumulated variable, and the same process is repeated for the next individual. In the next round, that individual's probability is summed with the previous individual(s) probability, increasing its chance for mating.

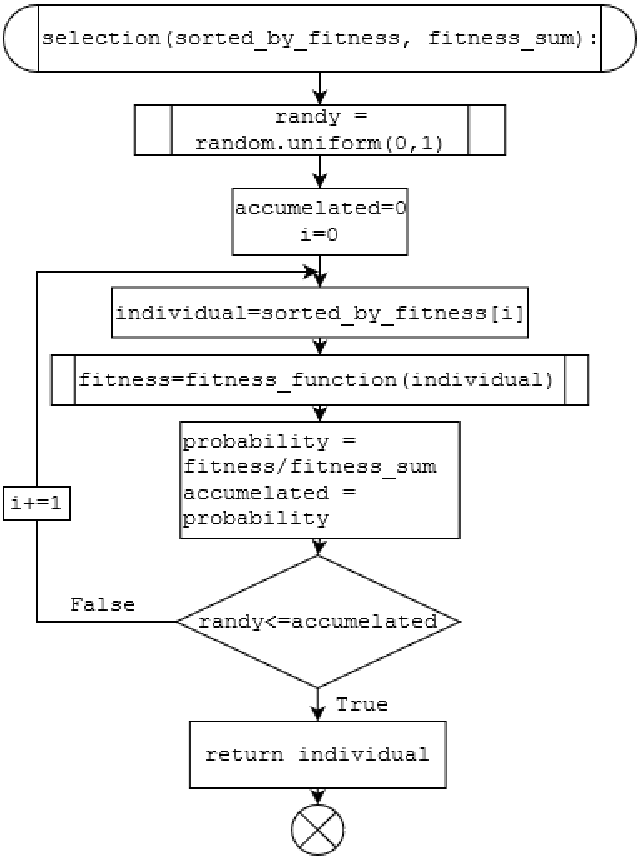

Figure 15: selection(sorted_by_fitness, fitness_sum)

3.5.4 Mating

The mating function, see Figure 16, is responsible for creating a new child from two parents' individuals. This is achieved with a simple yet effective solution. Return value for mating function is the mean of two "parent" individuals.

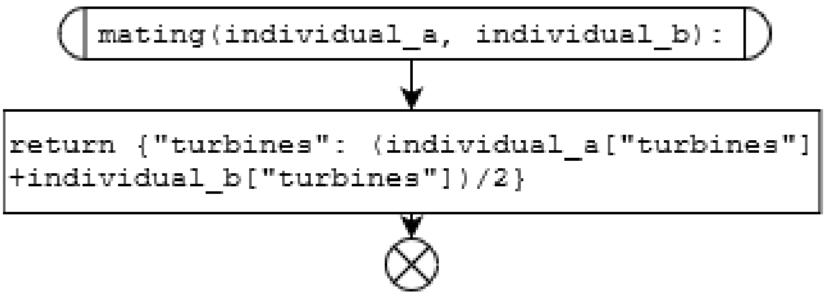

Figure 16: mating(individual_a, individual_b)

3.5.5 Mutation

The mutation function, see Figure 17, in this script takes two arguments, an individual to mutate and at how often an individual should mutate. A random float number between zero and one is generated, and if the randomly generated float number is equal to or less than mutation_rate, the mutation will commence. The mutation is usually not a huge variation of the original individual, but with a little experimentation and testing, the best results came from an entirely new generated individual. The individual generated has the same lower and upper boundaries as the first generation's individuals.

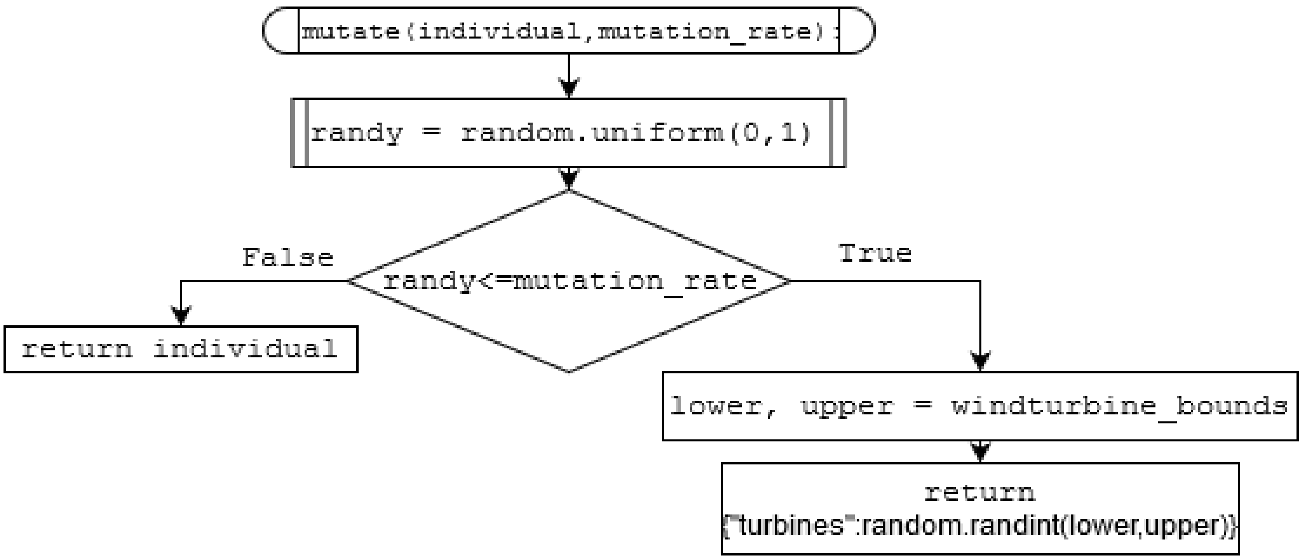

Figure 17: mutate(individual, mutation_rate)

3.5.6 Structure of the algorithm

Figure 18 is the loop performing the algorithm. It is a while True loop, which breaks if the number of generated generations is equal to the "generations" variable. It then enters a for loop, setting each individual equal to generation[index]. It performs the fitness_function for each individual and stores fitness value in temp_progress.

Furthermore, temp_progress is evaluated if it is better than the best_generated individual if it is the "new best individual" is stored as such. Every individual's cost/fitness is appended to an array alongside with the number of turbines. These values are used for plotting, making it easier to interpret the results, as the best generated individual is already stored.

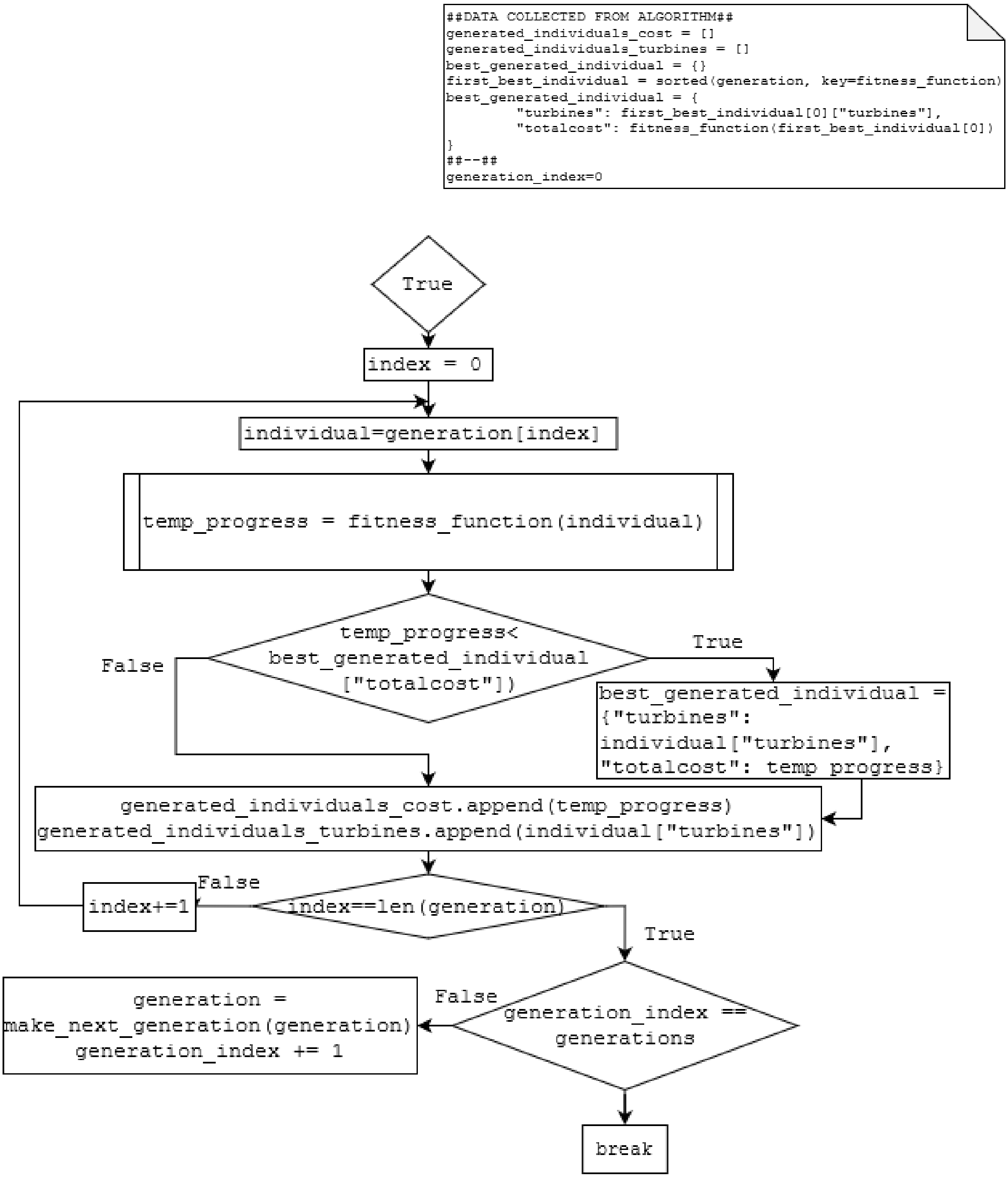

Figure 18: Structure of the algorithm

4 Results and discussion

The different scenarios are tested and recorded with the GA presented in section 3.5. In this chapter, the results from the different scenarios are provided and discussed.

4.1.1 Scenario 1

In scenario 1, the lower boundary for the number of the wind turbine is set to 1. The upper boundary is set to 50. It is believed pre-study that the optimum number of wind turbines for a hybrid system falls within this range. Population size is set to 20; this is also believed to be a sufficient number of individuals to reach a good result. Generations are set to 200, which should yield enough data to read. The mutation rate is set to 5 %. All numbers are generated in the floating-point spectrum, and no rounding is done for the calculations. This is to demonstrate the algorithm's ability to find a solution. Table 2 shows an overview of the algorithm settings. Rounded off results are included in section 4.1.4.

Table 2: Algorithm settings

Parameter	Configuration
Lower boundary	1
Upper boundary	50
Population size	20
Number of generations	200
Mutation rate	0.05

The blue dots in Figure 19 represent all individuals generated. The red dot represents the set of wind turbines, which results in the lowest cost. The black dot is the cost of operating without wind power for the six years period.

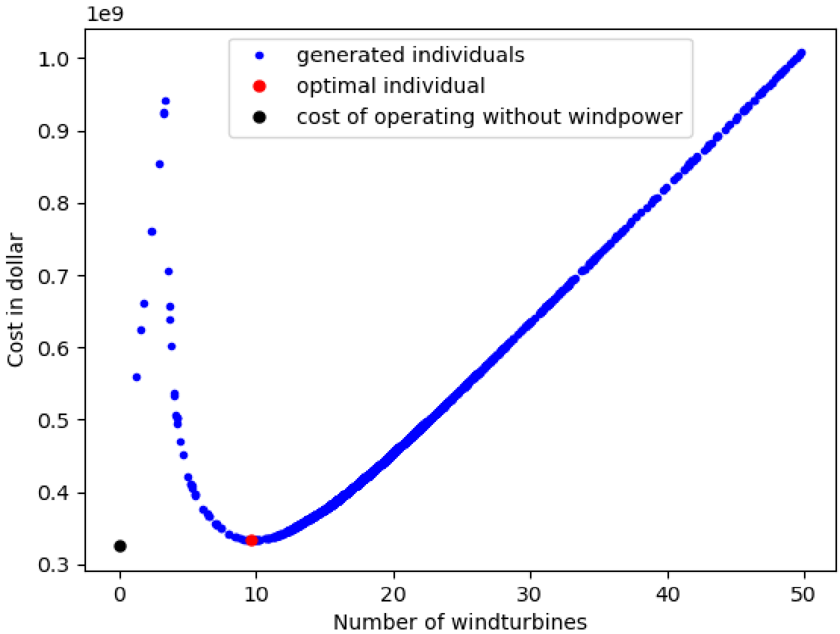

Figure 19: Total cost in relation to the number of installed wind turbines for scenario 1

In this case, the number of turbines that yields an optimum cost of operation is 9.62820762844231 wind turbines. Implementing this number of wind turbines gives a total cost of 333 936 050.28897613 USD. Table 3 shows the optimum number of wind turbines, excessive wind power, emissions, waste reduction, and the total cost for the scenario for the six years.

Table 3: Results from scenario 1

Number of turbines	Excessive wind power [MW/h]	Emission [Tons CO_2]	Waste reduction [%]	Cost [Mill USD]
9.62820762844231	190524.6843	297183.8735	81.1031	333.936

Table 4 shows the difference in total cost with and without wind turbines (red and black dot). For this scenario, it is 8 696 786 USD more expensive with wind turbines compared to without wind turbines.

Table 4: Cost of operation [USD] with and without wind turbines

Scenarios	Cost of operation [mill USD]
With wind turbines	333.936
Without wind turbines	325.239

4.1.2 Scenario 2

In this scenario, most of the algorithm settings are the same as in the first scenario, such as population size, generations, and boundaries (see Table 2). But in this case, it is calculated what the minimum number of wind turbines is needed if this platform is to reach the CO_2 reduction goal. From Figure 20, the green dotted line represents, at which point, the CO_2 reduction is reached. Everything to the right of the green line will suffice in reaching the climate goal.

The lowest number of turbines installed, which accomplishes this, is shown in Table 5. Table 5 also shows excessive wind power, emissions, waste reduction, and total cost for the six years. From Table 5, to gain a CO_2 reduction of at 59 %, the O&G platform needs to at least install 6. 4083680181856($\approx$ 7 wind turbines). Reducing the CO_2 emissions by 59 % results in an emission of 621333 tons CO_2.

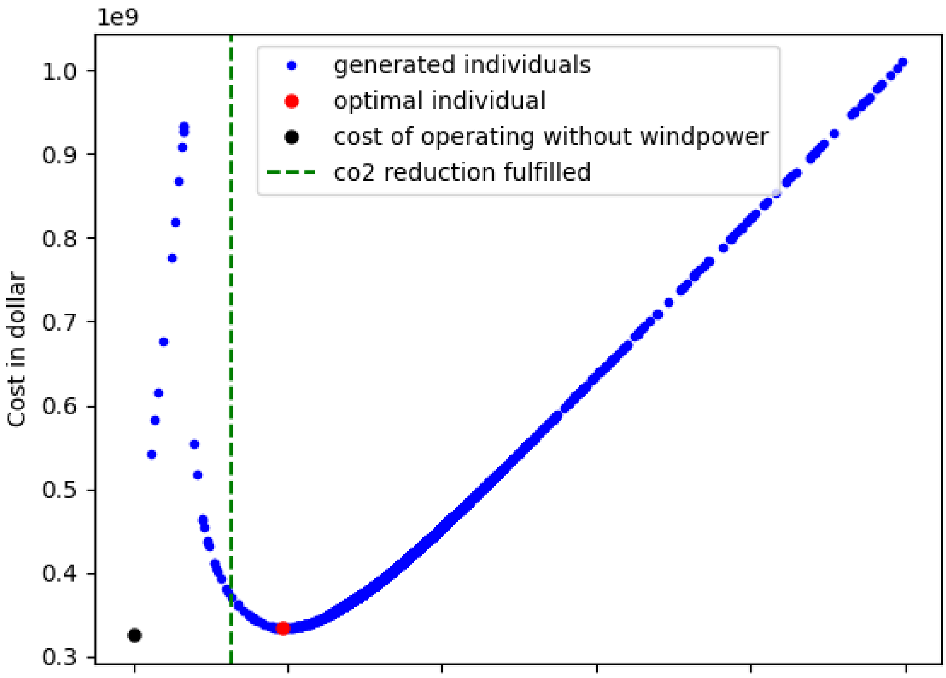

Figure 20: LCOE in relation to wind turbines and the CO_2 reduction goal

Table 5: Results from scenario 2

Number of turbines	Excessive wind power [MW/h]	Emissions [Tons CO_2]	Waste reduction [%]	Cost [Mill USD]
6.424286812196558	4334.8	619289	59.5	369.786

4.1.3 Scenario 3

In scenario 3, algorithm settings are the same as in the first and second scenarios, see Table 2. But in this scenario, it is assumed that in 2030 the LCOE for floating offshore wind turbines decreases by 30 %. It is important to note for this scenario, it would be desired to make a predicted weather forecast for the wind speed as it is a future scenario, but due to limited time, it is used the same wind speeds as for the other scenarios, which are based on historical data. From Figure 21, the blue dots represent all individuals generated, and the red dot the number of wind turbines that result in the lowest cost for the six years.

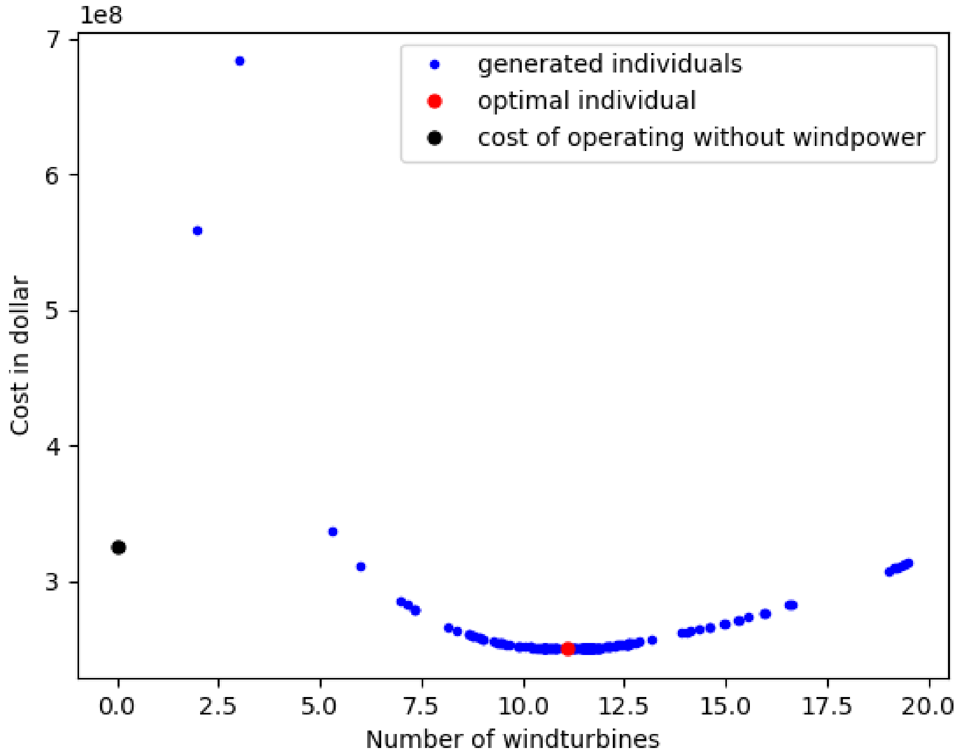

Figure 21: Total cost in relation to the wind turbines with LCOE for wind reduced by 30 %

In this scenario, the number of turbines that yields an optimum cost of operation is 11.11281897311095 wind turbines. This number of wind turbines gives a total cost of 249.783 mill USD. Table 6 shows the optimum number of wind turbines, excessive wind power, emissions, waste reduction, and the total cost of the scenario. Compared to the total cost without wind turbines (black dot), it is 75.456 mill USD less expensive with wind turbines (red dot).

Table 6: Results from scenario 3

Number of turbines	Excessive wind power [MW/h]	Emissions [Tons CO_2]	Waste reduction [%]	Cost [Mill USD]
11.11281897311095	368268.0204	208382.348	86.8643	249.783

A number of 11.11281897311095 wind turbines gives an excessive wind power of 368268.0204 MW/h, a CO_2 reduction of 208382.348 tons, which is equal to 86.8643 %.

4.1.4 Scenario 1,2 and 3 rounded numbers

This algorithm was calculated with different parameters. From previous results, the lower boundary for the wind turbine is set to 1 and upper to 20. All generated individuals have integer values, and the wind turbine number is rounded in the mating process. Figure 22 represents the results for scenarios 1 and 2.

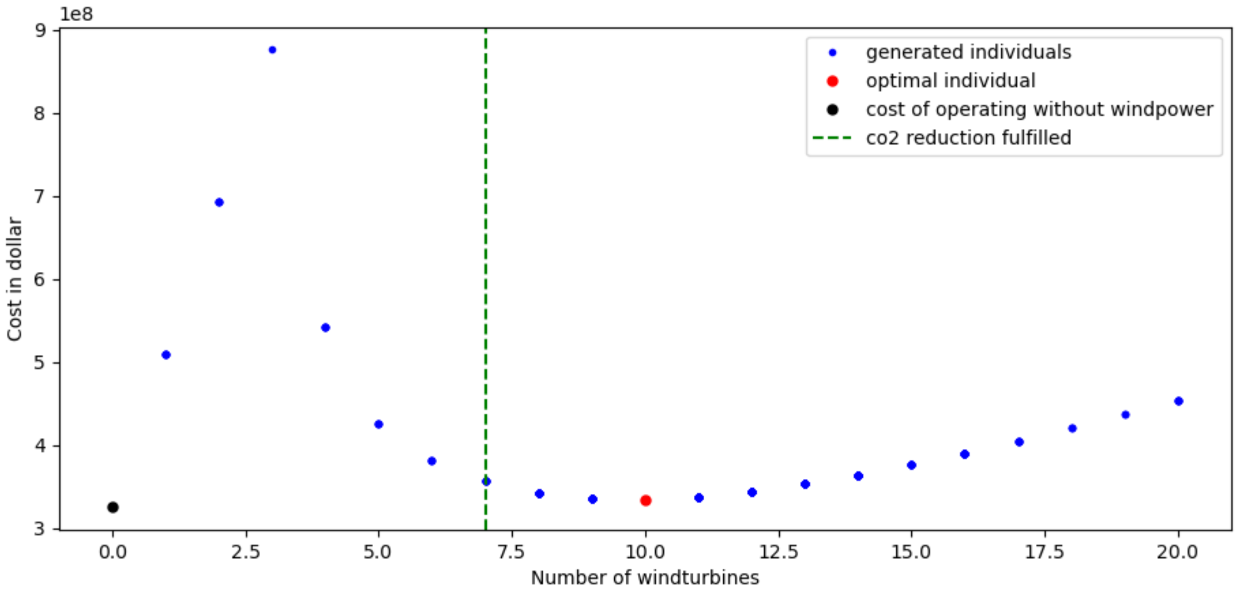

Figure 22: Total cost in relation to wind turbines and CO_2 reduction goal in integer values

Figure 23 represents the rounded number of wind turbines in relation to the total cost for scenario 3 for the six years. The following results from scenario 1,2, and 3 with a rounded number of wind turbines, and the total cost for the scenario without wind turbines (only gas turbines) are given in Table 7.

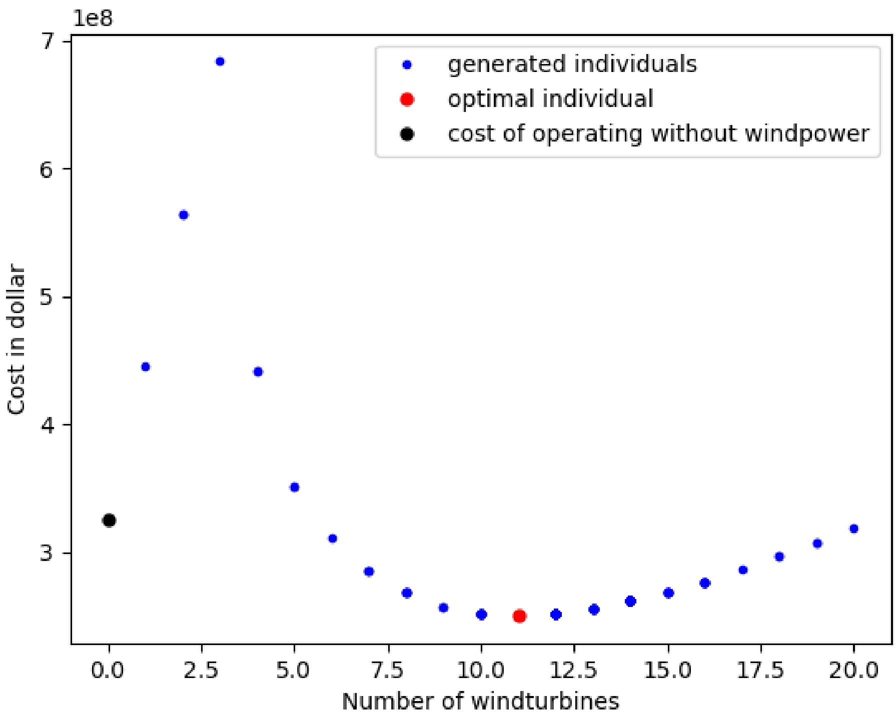

Figure 23: Total cost in relation to the number of wind turbines, with LCOE for wind turbines reduced by 30%

Table 7: Rounded results from scenario 1,2, and 3 and without wind turbines

Scenario	Number of Wind turbines	Excess wind power [MW/h]	Emissions [Tons CO_2]	Waste Reduction [%]	Cost [mill USD]
#1	10	231 262.1984	272011.7031	82.74676	334.046
#2	7	16 994.5291	548141.8039	64.2826	354.535
#3	11	355 434.890	213622.9224	86.5279	249.818
Without wind turbines	0	-	1500576.0	-	325.239

Figure 24 represents how efficient each wind turbine is as the number of wind turbines increases. The payoff reduces exponentially, as emission reduction percentage increases minimally, and the efficiency reduces drastically. Both from an economical and an environmental point of view, it does not seem reasonable to go beyond the black line.

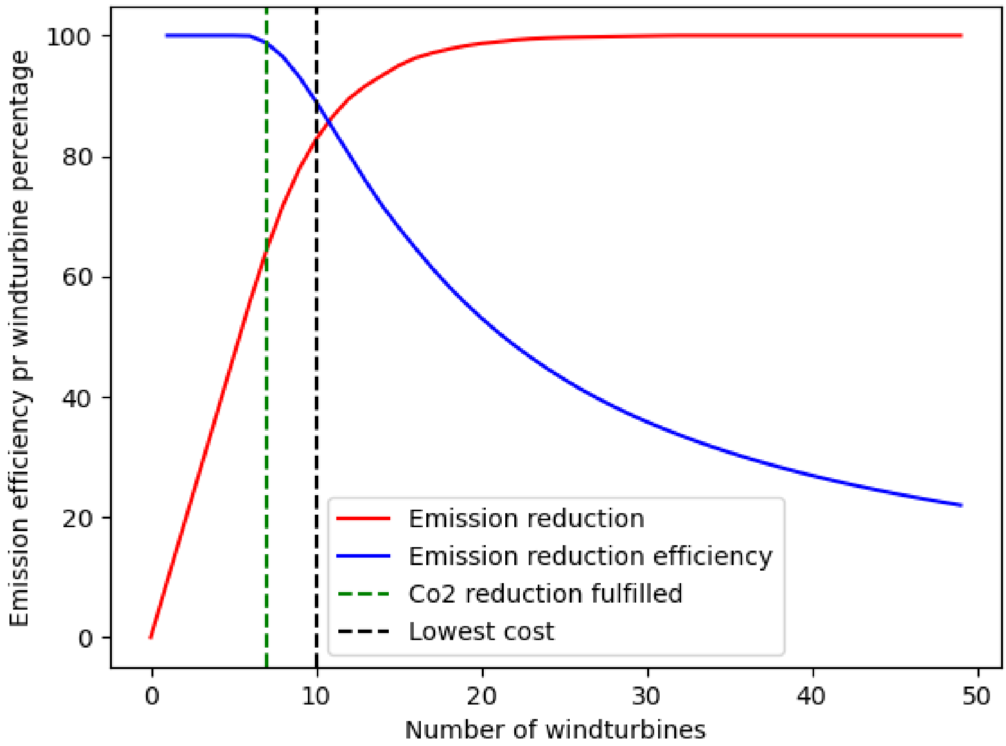

Figure 24: Emission reduction efficiency by turbine

When comparing the results from the different scenarios in Table 7, there is no doubt that it is that scenario 3 is most profitable as it has a reduced LCOE of 30 %, which represents a significantly lower total cost compared to the other scenarios. Scenario 3 is also the only scenario that has a lower total cost than without wind turbines. It is important to note that the total cost without wind turbines is based on the average LCOE for gas-fired combined cycle power plant of the Norwegian market. Scenario 2 is one of the three with less excess wind power, but has a higher emission of CO_2, compared to the other. The higher amount of CO_2 emissions for scenario 2 can be explained by the fact that scenario 1 and 3 has a higher number of wind turbines and therefore provides a greater production of wind power. This again gives scenarios 1 and 3 a much higher excess wind power than scenario 2. Nevertheless, all of the scenarios have a significant amount of excess wind power. Energy storage such as batteries could store this excess power and be dispatched when periods with low wind speeds occur. Comparing the results from scenario 1 and 2, the total cost for scenario 2 is higher than for scenario 1. This is because LCOE decreases as more windturbines are installed, but excessive also increases as more windturbines are installed, and that needs to be taken into account. Either by directly increasing the LCOE as the xcessive energy is not used, or "pay" for the excessive energy which yields the same result.

A lot was learned about programming during this research, and Python programming especially. In retrospect, the GA is utilized on an optimization problem, which in reality could have been tested for an individual by individual, increasing wind turbines by one and selecting the one with the lowest cost. But the experience gained from setting up a GA, constructing it, and watching it work was excellent. The GA would have been more fit for a problem with more chromosomes. This could be further expanded upon quite feasibly with the existing code for this thesis. That would take full advantage of the algorithm's possibilities. For example, there could have been a second chromosome with solar panels. More inputs showcase the use case of the GA. Even so, the results conducted during the research are intriguing.

5 Conclusion

The main objective of this thesis is to minimize the total cost for a hybrid renewable and gas system, while always meeting the load demand. The hybrid system consists of seven gas turbines located at an O&G installation implemented with a varied number of 8 MW offshore wind turbines. The optimization method used for this thesis is GA. For all of the scenarios, the algorithm settings that gained the optimum solution were; the population size should be 20; the number of generations should be 200; the mutation rate should be 5 %; the lower boundary set to 1 and the upper to 50. The optimal solution was found for three different scenarios for the hybrid system. The first scenario was to find how many wind turbines should be implemented to gain the minimum total cost while meeting the load demand. The number of wind turbines that gave the optimal solution for this scenario was 10 wind turbines. 10 wind turbines give the total cost of 334.046 mill USD. For scenario 2 it was desirable to find how many wind turbines should be implemented to the hybrid system to reduce the direct CO_2 emissions by 59 %, the algorithm found that the optimal solution was 7 wind turbines. As LCOE increases by the number of installed wind turbines, the total cost of implementing 7 wind turbines gives a total cost of 354.535 mill USD. For scenario 3, the LCOE for wind turbines were decreased by 30 %. The algorithm found that the optimal number of wind turbines that gave the lowest cost, while meeting the load demand for this scenario was 11 wind turbines. The total cost for this number of wind turbines with a decreased LCOE for wind turbines of 30% was 249.818 mill USD. The total cost for scenario 3 is lower than the total cost without wind turbines, which were found to be 325.239 mill USD. Scenario 3 is the only scenario that had a total cost lower than without wind turbines.

From the obtained results from the algorithm, it is clear that implementing floating offshore wind turbines still is an expensive technology to implement, but if the cost of the technology starts to decrease as the technology starts to mature as it has shown to do for bottom fixed wind turbines for the last years and the years to come. As shown for scenario 3 in this study, the total cost can even be cheaper than the cost without the wind turbines, making floating offshore wind an attractive energy source to replace some of the fossil energy and reduce GHG emissions. But as wind energy is intermittent as it is dependent on the right wind resources, it could never replace 100 % of the fossil fuels by itself. Implementing other energy sources like PV and energy storage, such as batteries, can help to reduce the use of fossil fuels.

6 Recommendations

Due to lack of time, there is further work to be done in this study to get more realistic results. In this study, assumptions were used for the load profile of the O&G platform, as this was confidential material. For future work, retrieving correct data for the load profile of the O&G platform is recommended. For future scenarios in this study, it is used historical wind data, for further work, it is proposed to implement weather forecasting to predict wind for future scenarios.

This study case is also based on a monthly time period for six years due to the limited data for the load. For further work, it would be more accurate to have data for an hourly- or daily time scale. Using a daily time scale and do an intraday analysis would make it possible to address the high variability of the wind resources and its impact on always meeting the load demand.

In this study, LCOE is based on previous work and average figures; for future work, it is recommended to do a study on all costs, including installation cost, operation, and maintenance, decommission, etc. for floating offshore wind and create an LCOE based on the findings. In this study, the possibility of implementing batteries has not been considered; for future work, this may be an essential factor for storing excess wind energy for days without wind.

For future work, the GA should further implement more chromosomes, and not only include wind turbines, but implement other chromosomes with, for example, solar panels and batteries. Implementing multiple chromosomes in the algorithm makes the algorithm more fit, and the algorithm's full potential could be achieved. For further work, other optimization methods, such as PSO or SA, should also be tested for comparing the results.

7 List of References

[1] Regjeringen, “Klimaendringer og norsk klimapolitikk.” [Online]. Available: https://www.regjeringen.no/no/tema/klima-og-miljo/innsiktsartikler-klima-miljo/klimaendringer-og-norsk-klimapolitikk/id2636812/. [Accessed: 14-Mar-2020].

[2] Energi og klima, “Norges utslipp.” .

[3] W. He *et al.*, “The potential of integrating wind power with offshore oil and gas platforms,” *Wind Eng.*, vol. 34, no. 2, pp. 125–137, 2010, doi: 10.1260/0309-524X.34.2.125.

[4] Norsk Petroleum, “Fields.” [Online]. Available: https://www.norskpetroleum.no/en/facts/field/. [Accessed: 20-May-2020].

[5] Regjeringen, “Oil and Gas.” [Online]. Available: https://www.regjeringen.no/en/topics/energy/oil-and-gas/id1003/. [Accessed: 15-May-2020].

[6] Norwegian Petroleum Directorate, “Employment in the petroleum industry.”

[7] F. O. G. Utviklingstrekk, “Olje- og gassindustriens miljøarbeid fakta og utviklingstrekk,” 2019.

[8] IRENA, *Global energy transformation: A roadmap to 2050 (2019 edition)*. 2019.

[9] “Rapport Elektrifiseringsvurderinger for midtre nordsjø.”

[10] D. Vyncke-Wilson, “Advantages of aero-derivative gas turbines in industrial applications: technical, operational and financial considerations on equipment selection.,” *20th Symp. Ind. Appl. Gas Turbines Comm.*, no. 13-IAGT-206, pp. 1–15, 2013.

[11] General Electric (GE), “LM/TM2500 SAC TO DLE.” [Online]. Available: https://www.ge.com/power/services/gas-turbines/upgrades/lm2500-sac-to-dle. [Accessed: 15-May-2020].

[12] GE power, “Lm2500 power plants 21-34,” vol. 60, p. 34340, 2019.

[13] E. F. Norway, “TAXES AND EMISSIONS TRADING,” 2020. [Online]. Available: http://energifaktanorge.no/en/et-baerekraftig-og-sikkert-energisystem/avgifter-og-kvoteplikt/.

[14] M. J. Pickl, “The renewable energy strategies of oil majors – From oil to energy?,” *Energy Strateg. Rev.*, vol. 26, no. April, p. 100370, 2019, doi: 10.1016/j.esr.2019.100370.

[15] Norwegian Petroleum Directorate, “Emissions to air.” [Online]. Available: https://www.norskpetroleum.no/en/environment-and-technology/emissions-to-air/. [Accessed: 25-May-2020].

[16] International Energy Agency (IEA), “Offshore Wind Outlook 2019,” 2019.

[17] International Energy Agency (IEA), “Tracking Power,” 2019. [Online]. Available: https://www.iea.org/reports/tracking-power-2019/offshore-wind.

[18] I. International and R. Energy, *Future of Wind - Technology , Grid Integration and Socio-Economic Aspects*. .

[19] J. Bråten Måssing, “Utbygging av Hywind Tampen godkjent,” 2020. [Online]. Available: https://www.nrk.no/vestland/utbygging-av-hywind-tampen-er-godkjent-1.14977920. [Accessed: 08-Apr-2020].

[20] P. Jain, “Wind energy engineering.” McGraw-Hill, New York, 2011.

[21] IRENA International Renewable Energy Agency, *Renewable Power Generation Costs in 2017*. 2018.

[22] E. M. Module, “Cost and Performance Characteristics of New Generating Technologies , Annual Energy Outlook 2020,” vol. 2020, no. January, pp. 1–4, 2020.
[23] U. S. E. I. Administration, “Levelized Cost and Levelized Avoided Cost of New Generation Resources in the Annual Energy Outlook 2016,” *Us Eia Lcoe*, no. February, pp. 1–20, 2020.
[24] Thomas F. Stacy; and G. S. Taylor, “The Levelised Cost of Electricity from Existing Generation Resources,” *Inst. Energy Res.*, no. June, p. 60, 2016.
[25] Centre for Sustainable Energy Studies, *Conditions for Growth in the Norwegian Offshore Wind Industry*. 2019.
[26] A. Myhr, C. Bjerkseter, A. Ågotnes, and T. A. Nygaard, “Levelised cost of energy for offshore fl oating wind turbines in a life cycle perspective,” *Renew. Energy*, vol. 66, pp. 714–728, 2014, doi: 10.1016/j.renene.2014.01.017.
[27] L. Castro-santos, A. Filgueira-vizoso, C. Álvarez-feal, and L. Carral, “Influence of Size on the Economic Feasibility of Floating Offshore Wind Farms,” pp. 1–13, 2018, doi: 10.3390/su10124484.
[28] The Norwegian Water Resources and Energgy Directorate (NVE), “Kostnader i energisektoren,” 2020. [Online]. Available: https://www.nve.no/energiforsyning/energiforsyningsdata/kostnader-i-energisektoren.
[29] J. Mletzko, S. Ehlers, and A. Kather, “Comparison of natural gas combined cycle power plants with post combustion and oxyfuel technology at different CO2 capture rates,” *Energy Procedia*, vol. 86, pp. 2–11, 2016, doi: 10.1016/j.egypro.2016.01.001.
[30] P. L. Spath and M. K. Mann, “Life Cycle Assessment of a Natural Gas Combined-Cycle Power Generation System - National Renewable Energy Laboratory - NREL/TP-570-27715,” 2000.
[31] M. J. Mazzetti, “OTC-24034-MS Offshore Energy Efficiency Technologies.”
[32] E.Øyslebø and M.Korpås, “Electrification of offshore petroleum installations with offshore wind integration.”
[33] GE power, “Hybrid Solutions.”
[34] X.-S. Yang, “Metaheuristic Optimization,” 2011. [Online]. Available: http://www.scholarpedia.org/article/Metaheuristic_Optimization. [Accessed: 26-May-2020].
[35] J. Sadeghi, S. Sadeghi, and S. T. A. Niaki, “Optimizing a hybrid vendor-managed inventory and transportation problem with fuzzy demand: An improved particle swarm optimization algorithm,” *Inf. Sci. (Ny).*, vol. 272, pp. 126–144, 2014, doi: 10.1016/j.ins.2014.02.075.
[36] M. Tabassum, “a Genetic Algorithm Analysis Towards Optimization Solutions,” *Int. J. Digit. Inf. Wirel. Commun.*, vol. 4, no. 1, pp. 124–142, 2014, doi: 10.17781/p001091.
[37] M. Ghofrani and N. N. Hosseini, “Optimizing Hybrid Renewable Energy Systems: A Review,” *Intech*, vol. i, no. Sustainable Energy-Technological Issues, Applications and case studies, p. 13, 2012, doi: 10.1016/j.colsurfa.2011.12.014.
[38] S. Bandaru and K. Deb, “Metaheuristic techniques,” *Decis. Sci. Theory Pract.*, pp. 693–749, 2016, doi: 10.1201/9781315183176.
[39] A. Saif, K. G. Elrab, H. H. Zeineldin, S. Kennedy, and J. L. Kirtley, “Multi-objective capacity planning of a PV-wind-diesel-battery hybrid power system,” *2010 IEEE Int. Energy Conf. Exhib. EnergyCon 2010*, no. December 2010, pp. 217–222, 2010, doi: 10.1109/ENERGYCON.2010.5771679.

[40] A. C. Nagabhushana, R. Jyoti, and A. B. Raju, “Economic analysis and comparison of proposed HRES for stand-alone applications at various places in Karnataka state,” *2011 IEEE PES Int. Conf. Innov. Smart Grid Technol. ISGT India 2011*, pp. 380–385, 2011, doi: 10.1109/ISET-India.2011.6145346.
[41] S. Ashok, “Optimised model for community-based hybrid energy system,” *Renew. Energy*, vol. 32, no. 7, pp. 1155–1164, 2007, doi: 10.1016/j.renene.2006.04.008.
[42] L. Riboldi and L. O. Nord, “Offshore power plants integrating a wind farm: Design optimisation and techno-economic assessment based on surrogate modelling,” *Processes*, vol. 6, no. 12, 2018, doi: 10.3390/pr6120249.
[43] B. O. Bilal, V. Sambou, C. M. F. Kébé, P. A. Ndiaye, and M. Ndongo, “Methodology to size an optimal stand-alone PV/wind/diesel/battery system minimizing the levelized cost of energy and the CO 2 emissions,” *Energy Procedia*, vol. 14, no. 2011, pp. 1636–1647, 2012, doi: 10.1016/j.egypro.2011.12.1145.
[44] R. Dufo-López *et al.*, “Multi-objective optimization minimizing cost and life cycle emissions of stand-alone PV-wind-diesel systems with batteries storage,” *Appl. Energy*, vol. 88, no. 11, pp. 4033–4041, 2011, doi: 10.1016/j.apenergy.2011.04.019.
[45] H. Yang, Z. Wei, and L. Chengzhi, “Optimal design and techno-economic analysis of a hybrid solar-wind power generation system,” *Appl. Energy*, vol. 86, no. 2, pp. 163–169, 2009, doi: 10.1016/j.apenergy.2008.03.008.
[46] S. M. Hakimi and S. M. Moghaddas-Tafreshi, “Optimal sizing of a stand-alone hybrid power system via particle swarm optimization for Kahnouj area in south-east of Iran,” *Renew. Energy*, vol. 34, no. 7, pp. 1855–1862, 2009, doi: 10.1016/j.renene.2008.11.022.
[47] M. Amer, A. Namaane, and N. K. M’Sirdi, “Optimization of hybrid renewable energy systems (HRES) using PSO for cost reduction,” *Energy Procedia*, vol. 42, pp. 318–327, 2013, doi: 10.1016/j.egypro.2013.11.032.
[48] U. Boonbumroong, N. Pratinthong, S. Thepa, C. Jivacate, and W. Pridasawas, “Particle swarm optimization for AC-coupling stand alone hybrid power systems,” *Sol. Energy*, vol. 85, no. 3, pp. 560–569, 2011, doi: 10.1016/j.solener.2010.12.027.
[49] M. Bashir and J. Sadeh, “Optimal sizing of hybrid wind/photovoltaic/battery considering the uncertainty of wind and photovoltaic power using Monte Carlo,” *2012 11th Int. Conf. Environ. Electr. Eng. EEEIC 2012 - Conf. Proc.*, pp. 1081–1086, 2012, doi: 10.1109/EEEIC.2012.6221541.
[50] F. J. Ardakani, G. Riahy, and M. Abedi, “Design of an optimum hybrid renewable energy system considering reliability indices,” *Proc. - 2010 18th Iran. Conf. Electr. Eng. ICEE 2010*, vol. 1000, pp. 842–847, 2010, doi: 10.1109/IRANIANCEE.2010.5506958.
[51] M. Abdelaziz, M. Ali, and M. Eltamaly, *Studies in Systems, Decision and Control 121 Modeling and Simulation of Smart Grid Integrated with Hybrid Renewable Energy Systems*. .
[52] M. Korpås, L. Warland, W. He, and J. O. G. Tande, “A case-study on offshore wind power supply to oil and gas rigs,” *Energy Procedia*, vol. 24, no. 1876, pp. 18–26, 2012, doi: 10.1016/j.egypro.2012.06.082.
[53] A. R. Aardal, J. I. Marvik, H. Svendsen, and J. O. Giaever Tande, “Offshore wind as power supply to oil and gas platforms,” *Offshore Technol. Conf. Proc.*, vol. 2, no. May, pp. 1624–1631, 2012, doi: 10.4043/23245-ms.
[54] W. He, K. Uhlen, M. Hadiya, Z. Chen, G. Shi, and E. del Rio, “Case Study of Integrating an Offshore Wind Farm with Offshore Oil and Gas Platforms and with

an Onshore Electrical Grid," *J. Renew. Energy*, vol. 2013, no. April, pp. 1–10, 2013, doi: 10.1155/2013/607165.
[55] Meteorologisk institutt, "eKlima." [Online]. Available: http://sharki.oslo.dnmi.no/portal/page?_pageid=73,39035,73_39049&_dad=portal&_schema=PORTAL.
[56] Wind-turbine-models.com, "Vestas V164-8.0." [Online]. Available: https://en.wind-turbine-models.com/turbines/318-vestas-v164-8.0. [Accessed: 02-Feb-2020].
[57] Oljedirektoratet, "Oversikt over motorer og turbiner," 2013. [Online]. Available: https://www.npd.no/publikasjoner/rapporter/utslipp-av-nox-fra-petroleumsvirksomheten-pa-norsk-sokkel/. [Accessed: 20-Feb-2020].
[58] Å. Tveiten, "Faculty of Science and Technology MASTER ' S THESIS," 2018.
[59] Oljedirektoratet, "Faktasider." [Online]. Available: https://factpages.npd.no/nb-no/field/pageview/all/43686. [Accessed: 14-Feb-2020].
[60] the pandas development team, "Pandas documentation," 2020. [Online]. Available: IwAR3fJy043zWbkL3UuMAoIT6SDxI1GTXFkgHcdQQXcZjF3zzdZSKKtFVykA0. [Accessed: 05-May-2020].
[61] The Matplotlib development team, "matplotlib overview," 2020. [Online]. Available: https://matplotlib.org/contents.html?fbclid=IwAR2mLuLSI82LGhqwMgO1DoFRyl80Hyjj4bIdo_qu3yrJSu8dm4TAR_fG9_E. [Accessed: 05-May-2020].
[62] Python Software Foundation, "random — Generate pseudo-random numbers." [Online]. Available: https://docs.python.org/3/library/random.html.

Appendices

Appendix A

Python Scripts

optimizationAlgorithm.py is the main script with the genetic algorithm. It loads the CSV files in the beginning, which are created in create_csv_files.py. Load.py calculates the load profile used in this thesis, and which is referenced as load throughout the thesis. Create_csv_files.py creates both the load.csv and potential_windpower.csv. It also calculates potential wind power using the wind data from MET and power curve from vestas. Power_output _wt.py contains most of the core functions and calculations used in the optimization algorithm, and the efficiency_of_windturbines.py. Efficiency_of_windturbines is used to calculate and plot Figure 23, which represents how efficient each wind turbine is at reducing CO_2 emission.

Credit for the LCOE_wind.py script to my supervisor, Joao Leal.

optimizationAlgorithm.py

```
import random
from LCOE_wind import LCOE_wind
from power_output_wt import objective_calculation,
optimal_co2_reduction_requirement,cost_of_wind_and_gas
import numpy as np
import matplotlib.pyplot as plt

##GET DATA##
load = np.loadtxt('our_load.csv', delimiter=',')
potential_windpower =  np.loadtxt('power_from_one_turbine.csv', delimiter=',')
##--##
##MENU##
windturbine_bounds = (1,50)
pop_size = 10
mutation_rate = 0.05
generations = 10
##--##

def initial_population(pop_size, wturbines_bounds):
    lower_boundery, upper_boundery = wturbines_bounds

    population = []
    for i in range(pop_size):
        individual = {
            "turbines": random.uniform(lower_boundery, upper_boundery),
        }
        population.append(individual)

    return population
generation = initial_population(pop_size, windturbine_bounds)
```

```python
##fitness function data##
global_computed_turbine_values = []
best_co2_option = []
best_co2_option.append(generation[0])
best_co2_option[0]["emission_reduction_percentage"] = 0
##--##
def fitness_function(individual):
    turbines_installed = individual["turbines"]
    LCOE = LCOE_wind(turbines_installed*8)

    for i in range(len(global_computed_turbine_values)):
        if(global_computed_turbine_values[i]["turbines"] == turbines_installed):
            return global_computed_turbine_values[i]["fitness"]

    calc = objective_calculation(individual["turbines"], potential_windpower, load)
    fitness = ((((calc["total_excessive_wp"])*LCOE)+calc["total_cost"])**(-1))

    computed={
        "turbines":turbines_installed,
        "fitness":fitness
    }
    global_computed_turbine_values.append(computed)
    best_co2_option.append(optimal_co2_reduction_requirement(best_co2_option[-1], calc))
    return fitness

def selection(sorted_by_fitness, fitness_sum):

    randy = random.uniform(0,1)
    accumulated = 0
    for individual in sorted_by_fitness:
        fitness = fitness_function(individual)
        probability = fitness/fitness_sum
        accumulated += probability
        if randy <= accumulated:
            return individual

def mating(individual_a, individual_b):
    return {"turbines": (individual_a["turbines"]+individual_b["turbines"])/2}
```

```
def mutate(individual,mutation_rate):
    randy = random.uniform(0,1)

    if(randy<=mutation_rate):
        lower, upper = windturbine_bounds
        return {
            "turbines":random.randint(lower, upper)
        }
    else:
        return individual

def make_next_generation(previous_generation):
    next_generation =[]
    sorted_by_fitness_population = sorted(previous_generation, key=fitness_function)
    fitness_sum = sum(fitness_function(individual) for individual in generation)

    for i in range(len(previous_generation)):
        first_choice = selection(sorted_by_fitness_population, fitness_sum)
        second_choice = selection(sorted_by_fitness_population, fitness_sum)

        individual = mating(first_choice, second_choice)
        individual = mutate(individual,mutation_rate)
        next_generation.append(individual)

    return next_generation
```

```python
##DATA COLLECTED FROM ALGORITHM##
generated_individuals_cost = []
generated_individuals_turbines = []
best_generated_individual = {}
first_best_individual = sorted(generation, key=fitness_function)
best_generated_individual = {
        "turbines": first_best_individual[0]["turbines"],
        "totalcost": (fitness_function(first_best_individual[0])**-1)
}
##--##
generation_index=0
while True:
    for individual in generation:
        fitness = fitness_function(individual)
        temp_progress = fitness**-1
        if(temp_progress<best_generated_individual["totalcost"]):

            best_generated_individual = {
            "turbines": individual["turbines"],
            "totalcost": temp_progress
            }
        generated_individuals_cost.append(temp_progress)
        generated_individuals_turbines.append(individual["turbines"])

    if generation_index == generations:
        break

    generation = make_next_generation(generation)
    generation_index += 1
#plotting#
total_cost_for_system_without_windturbines = 0
for i in range(len(load)):
    total_cost_for_system_without_windturbines+=cost_of_wind_and_gas(0,0,load[i])

sorted_co2_options = sorted(best_co2_option, key=lambda i:i["turbines"])
```

```python
#plotting#
plt.ylabel("Cost in dollar")
plt.xlabel("Number of windturbines")
s = [0.5]*len(generated_individuals_turbines)
plt.plot(generated_individuals_turbines, generated_individuals_cost, 'bo', label="generated
plt.plot(best_generated_individual["turbines"], best_generated_individual["totalcost"], 'ro
plt.plot(0, total_cost_for_system_without_windturbines, 'ko', label="cost of operating with

plt.plot()
plt.legend()
plt.show()
```

load.py

```python
import pandas
import matplotlib.pyplot as plt
import numpy as np
##READ DATA##
df = pandas.read_excel(r'production_oil_and_gas_gullfaks.xlsx')
brutto_oil_column = df['Brutto oil'].values
brutto_gas_column = df['Brutto gas'].values
month_column = df['Month'].values
year_column = df['year'].values
total_column =df['Total'].values
##--##

def loadProfile():
    total_annual_defined_years = [0]*6
    current_year = 2012
    year_element = 0
    element = 0
    for total in brutto_oil_column:
        total_annual_defined_years[year_element] += total_column[element]
        element += 1
        if(year_column[element] == 2018):
            break
        if(year_column[element] != current_year):
            current_year += 1
            year_element += 1
    total = sum(total_annual_defined_years)
    average_total = total / 6

    ratio_production_consumption = 392000/average_total #MWh/mill sm3
    element=0
    load=[0]*72
    for myIndex in range(0,72):
        load[myIndex] = total_column[element] * ratio_production_consumption
        element+=1
    return load
```

create_cvs_files.py

```
from load import loadProfile
import numpy as np
import pandas
import math
import matplotlib.pyplot as plt
import settings as s
s.init()

##VINDDATA FROM GULLFAKS##
winddata = pandas.read_excel(r'gullfaks_vinddata_2007_til_2017.xlsx')
year_column = winddata['Year'].values
month_column = winddata['Month'].values
day_column = winddata['Day'].values
hour_column = winddata['Hour'].values
windspeed_column = winddata['Windspeed'].values
##--##
##POWER CURVE FROM VESTAS##
df = pandas.read_excel(r'PowerCurveVestas.xlsx')
windspeed_power_column = df['Windspeed'].values
power_column = df['Power'].values
##--##
def total_power_output():
    windspeed_data = {
        "nexthour": 0,
        "windspeed": 0,
    }
    completed_months = []

    while True:
        if(s.index_of_excel_file >= len(year_column)-1):
            break
        current_month = month_column[s.index_of_excel_file]
        this_months_data = {
            "month": current_month,
            "power": 0,
        }
        while True:
            if(s.index_of_excel_file >= len(year_column)-1):
                completed_months.append(this_months_data)
                break
            else:
                sampled_month = month_column[s.index_of_excel_file]
            if(current_month!=sampled_month):
                completed_months.append(this_months_data)
```

```
                break
            else:
                windspeed_data = average_windspeed_per_hour(windspeed_data)
                this_months_data["power"] += power_in_kw(windspeed_data)
    return completed_months

def avg(minute_count, total_windspeed, sampled_hour):
    if(minute_count < 1):
        minute_count = 1
    average_windspeed = total_windspeed/minute_count
    data =  {
        "windspeed":average_windspeed,
        "nexthour":sampled_hour,
    }
    return data

def average_windspeed_per_hour(windspeed_data):
    minute_count = 0
    total_windspeed = 0
    data={}
    while True:
        if(s.index_of_excel_file >= len(hour_column)):
            data = avg(minute_count, total_windspeed, sampled_hour)
            break

        sampled_hour = hour_column[s.index_of_excel_file]
        if (windspeed_data["nexthour"] != sampled_hour):
            data = avg(minute_count, total_windspeed, sampled_hour)
            break
        else:
            total_windspeed += windspeed_column[s.index_of_excel_file]
            minute_count += 1
            s.index_of_excel_file+=1
    return data

def power_in_kw(wind_data):
    frac, whole = math.modf(wind_data["windspeed"])
    if(whole>13):
        return 8
    if(whole<4):
        return 0
    initial_power = power_column[int(whole)]
    factor = 1+frac
    estimated_power = factor*initial_power
    return estimated_power/1000
```

```
##########CRATING CSV FILES##########
total_power = total_power_output()
power_array = []

for i in range(len(total_power)):
    power_array.append(total_power[i]["power"])

x = np.arange(0, len(power_array))
plt.plot(x, power_array)
plt.show()

np.savetxt("power_from_one_turbine.csv", power_array, delimiter=",")

load = loadProfile()
np.savetxt("our_load.csv", load, delimiter=",")
##--##
```

power_output_wt.py

```
import pandas
import math
import matplotlib.pyplot as plt
import numpy as np
from LCOE_wind import LCOE_wind

def Egt(load, energy_wind_power):
    return load - energy_wind_power

def cost_of_wind_and_gas(Lcoe, Ewt, Egt):
    Cgt = 100.87#gascost
    Cco2 = 37.412#emission taxes
    return(Ewt*Lcoe)+(Egt*Cgt)+(Egt*Cco2)

def total_cost(num_of_windturbines, potential_windpower, load):
    data = objective_calculation(num_of_winturbines, potential_windpower, load)
    return data["total_cost"]

def total_excessive(num_of_winturbines, potential_windpower, load):
    data = objective_calculation(num_of_winturbines, potential_windpower, load)
    return data["total_excessive_wp"]

def co2_constraint(load, gas_requirement):
    ##CURRENT CO2 emission = 84.6%
    ##GOAL, CUT IT BY 35.46
    co2_reduction_goal = 59
    co2_factor = 0.638 #ton per MWh
    current_emission = load*co2_factor
    emission = gas_requirement*co2_factor
    if(emission <0):
        emission_reduction_percentage = 100
    else:
        emission_reduction_percentage = (((emission-current_emission)/current_emission)*-10(

    data = {
    "emission" : emission,
    "emission_reduction_percentage": emission_reduction_percentage,
    }
    return data
def optimal_co2_reduction_requirement(current_optimal, contender):
    if(contender["emission_reduction_percentage"]>=59):
            return contender
    return current_optimal
```

```python
def objective_calculation(num_of_winturbines, potential_windpower, load):
    LCOE = LCOE_wind(num_of_winturbines*8)#8MW
    power={}
    data = {
        "total_cost":0,
        "total_excessive_wp":0,
        "turbines": num_of_winturbines,
        "emission": 0,
        "emission_reduction_percentage":0
    }
    total_percentage_reduced = []
    for i in range(len(potential_windpower)):
        power = power_requirement(load[i] , potential_windpower[i] , num_of_winturbines)
        data["total_cost"] += cost_of_wind_and_gas(LCOE , power["consumed_windpower"] , pow
        data["total_excessive_wp"]+=power["excessive_windpower"]
        co2_data = co2_constraint(load[i], power["gas_requirement"])
        data["emission"]+=co2_data["emission"]
        total_percentage_reduced.append(co2_data["emission_reduction_percentage"])

    data["emission_reduction_percentage"] = np.mean(total_percentage_reduced)

    return data

def power_requirement(load, windpower, number_of_turbines):
    total_wp = windpower*number_of_turbines

    if(windpower*number_of_turbines<load):
         return {
            "consumed_windpower"  :total_wp,
            "excessive_windpower":0,
            "gas_requirement": Egt(load, total_wp)
        }
    else:
        return {
            "consumed_windpower":load,
            "excessive_windpower": total_wp-load,
            "gas_requirement": 0
        }
```

power_output_wt.py

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

def LCOE_wind(Pwind):
    X = np.array([100, 200, 300, 400, 500, 600])
    Y = np.array([117.05, 100.31, 96.29, 93.7, 92.69, 91.74])

    #30%REDUCTION#
    #for i in range(Y):
    #    Y = Y[i]*0.7
    n=len(Y)

    def model(X, a, b):
        'Nested function for the model'
        y=a*X/(X-b)
        return y

    guesses = [90, 1]

    popt, pcov = curve_fit(model, X, Y, guesses)
    n_popt=len(popt)

    y_fit=model(X,popt[0],popt[1])
    SS_mean=np.sum(np.square(Y-np.mean(Y)))

    SS_fit=np.sum(np.square(Y-y_fit))
    R2=1-SS_fit/SS_mean

    LCOE_val=model(Pwind,popt[0],popt[1])
    if Pwind<27.2:
        LCOE_val=model(27.2,popt[0],popt[1])

    x_fit=np.arange(8,600,1)
    f_fit=model(x_fit,popt[0],popt[1])
    for i in range(0,len(f_fit)):
        if x_fit[i]<27.2:
            f_fit[i]=model(27.2,popt[0],popt[1])

    return LCOE_val
```

LCOE_wind.py

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

def LCOE_wind(Pwind):
    X = np.array([100, 200, 300, 400, 500, 600])

    Y = np.array([117.05, 100.31, 96.29, 93.7, 92.69, 91.74])
    #30%REDUCTION#
    #for i in range(Y):
    #    Y = Y[i]*0.7

    def model(X, a, b):
        'Nested function for the model'
        y=a*X/(X-b)
        return y
    guesses = [90, 1]

    popt, pcov = curve_fit(model, X, Y, guesses) # executes the regression with curve_fit

    LCOE_val=model(Pwind,popt[0],popt[1])
    if Pwind<27.2:
        LCOE_val=model(27.2,popt[0],popt[1])

    # COMPUTING THE VALUES FROM REGRESSION IN THE GRID CREATED
    x_fit=np.arange(8,600,1)
    f_fit=model(x_fit,popt[0],popt[1])
    for i in range(0,len(f_fit)):
        if x_fit[i]<27.2:
            f_fit[i]=model(27.2,popt[0],popt[1])
    return LCOE_val
```

efficiency_of_windturbines.py

```
import matplotlib.pyplot as plt
import numpy as np
from power_output_wt import objective_calculation
##GET DATA##
load = np.loadtxt('our_load.csv', delimiter=',')
potential_windpower =  np.loadtxt('power_from_one_turbine.csv', delimiter=',')
##--##
##VARIABLES##
efficency_rate = 100/9.293095354351655
turbines = [] #turbines
emission_reduction_percentage = []
emission_reduction_efficiency = []
max_turbine = 50
##--##
for turbine in range(0, max_turbine):
    calc = objective_calculation(turbine, potential_windpower, load)
    if(turbine!=0):
        result = ((calc["emission_reduction_percentage"]/calc["turbines"])*efficency_rate)
        emission_reduction_efficiency.append(result)
    else:
        emission_reduction_efficiency.append(None)
    turbines.append(calc["turbines"])
    emission_reduction_percentage.append(calc["emission_reduction_percentage"])

##PLOTTING##
plt.ylabel("Emission efficiency pr windturbine percentage")
plt.xlabel("Number of windturbines")

plt.plot(turbines, emission_reduction_percentage, color='r', label='Emission reduction')
plt.plot(turbines, emission_reduction_efficiency, color='b', label='Emission reduction effi
plt.axvline(7, color="g", linestyle='--',label="Co2 reduction fulfilled")
plt.axvline(10, color="k", linestyle='--',label="Lowest cost")

plt.legend()
plt.show()
```

settings.py

```
def init():
    global index_of_excel_file
    index_of_excel_file = 0
```