BSc Graduation Thesis

Power flow control in a substation of a wind- and solar farm:

Design of an Optimisation Unit Considering a System's Physical Boundaries and Technical Constraints

Group members:
J. Bai
A.C. Neagu

Delft University of Technology
June 19, 2020

Power flow control in a substation of a wind- and solar farm:

Design of an Optimisation Unit Considering a System's Physical Boundaries and Technical Constraints

A Bachelor of Science Graduation Thesis

By

Alexandru Christian Neagu & Jinhan Bai

To be defended on Wednesday, July 1, 2020 at 13:00 PM

Student numbers:
4703723 (A.C. Neagu)
4708083 (J. Bai)

Supervisor:
Dr.ir. J.L. Rueda Torres, TU Delft

Thesis Committee:
Prof.Dr. A. Neto, TU Delft
Dr.ir. J.L. Rueda Torres, TU Delft
Dr.ir B. Gholizad, TU Delft
[redacted]

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
Delft University of Technology

Abstract

The goal of this graduation project was to design a state-of-the-art central farm controller. The designed central farm controller distinguishes itself from prevailing controllers by including a subsystem (the *optimisation unit*) that contributes to an improvement in power transmission efficiency, decrease in maintenance costs and an increase in system reliability/robustness. This thesis describes the design process of the *optimisation unit* and the verification of its feasibility. Originally, the plan was to test the design on a remote terminal unit (RTU), but unfortunately due to the current crisis, this was not possible. Therefore, the implementation and testing were carried out only in MATLAB. Tests were performed using a MATPOWER system model which has been derived from a real wind farm topology. The *optimisation unit* makes use of a meta-heuristic algorithm to solve an optimal reactive power flow dispatch optimisation problem. In this thesis, the feasibility of this *optimisation unit* is investigated. Furthermore, it is determined which devices should be controlled and of which the usage is optimised. This makes the treated optimisation problem a multiple objective optimisation. Lastly, the robustness is verified by extending the topology and testing the solutions of the *optimisation unit*.

Contents

Chapter 1

Introduction

1.1 Background

1.1.1 Power Flow Management in Substations

There are several ongoing wind farm (WF) projects in the Netherlands. There are for example a lot of small and old wind turbines that are being replaced with fewer and larger turbines. Wind parks generally consist of several wind turbine generator (WTG) strings. The power cables of the strings are gathered in the substation and transformers step up the voltage to deliver the power to the grid. Delivery is done at the point of common coupling (PCC). It is becoming a new case in the Netherlands that solar farms (SF) are also connected to these substations. The new farms have more generation capacity and thus substations should be scaled accordingly.

Power flow management is done as follows in such a substation: A central farm controller (CFC) receives setpoints from the transmission system operator (TSO). These setpoints can be converted into an amount of active and reactive power required at the PCC. It is usual that the required amount of active power is equal to the maximum active power generation possible at that moment. To achieve these setpoints, the CFC requests a certain amount of reactive power from each generating string. This is done by communicating these amounts to the local wind farm controllers (LWFC) and the local solar farm controllers (LSFC) by sending setpoints. Each local farm controller (LWFC or LSFC) distributes the active and reactive power production as desired between the available generation units on a string but ensures that the required setpoint by the CFC is met. While reaching the setpoint, the grid code must also be satisfied.

1.1.2 The System Topology in Question

The research is done on a real-life based case study of a WF. Due to confidentiality reasons, the exact case under study is not disclosed. However, a general system topology is presented in Fig. A.5. The system topology is shortly elaborated in this paragraph and then the optimisation problem is presented. The system in question consists of four 33 kV bus bars. To each bus bar, multiple WTG strings are connected. For reliability reasons, the bus bars can also be interconnected. Between the two bus bars in the middle, there is a shunt reactor which can absorb extra reactive power if needed. The bus bars are connected in pairs to the two main transformers. These transformers step up the voltage to 150 kV and have multiple tap positions. The tap positions alter the number of windings at the secondary side of the transformer. The switching of the taps is done by an on-load tap changer (OLTC). The transformers are connected to the switchgear at the PCC. After ensuring the grid requirements are met, power is delivered to the main grid.

1.2 Project Breakdown

As already mentioned in Section 1.1.1, requirements are given in the form of TSO setpoints and grid code. However, there are multiple ways in which these requirements can be reached. This is due to the fact that there are multiple controllable devices available in the system. These are for instance the individual reactive power outputs of the strings, the tap positions of the transformers and the connection of the shunt reactor. Each of these devices could receive a setpoint in order to reach an optimal system solution. This creates the possibility of achieving the same requirements using different combinations of setpoints, with some combinations to be more (economic) feasible than others. This means that some combinations allow for less active power losses and reduced system maintenance costs.

Currently, a CFC uses a rule-based approach for the distribution of reactive power but neglects the control of transformers and shunt devices. Furthermore, it takes grid compliance into account but neglects the possibility of more feasible setpoint combinations. Thus, current CFC's do not consider active power losses and system maintenance costs when performing park management. In this project, a CFC is proposed which distributes optimised setpoints that take the minimisation of active power losses and system maintenance costs into account. Furthermore, the proposed CFC also has the ability to control tap-changing transformers and the shunt reactor within a system. The design of the proposed CFC can be decomposed into three different sub-designs: the *control unit*, the *optimisation unit* and the *power flow model* of the farm. The interaction between the three sub-systems is displayed in Figure 1.1. In this thesis, the *optimisation unit* is designed and simulated. Below, the role of the *optimisation unit* within the CFC and its envisioned operation are discussed.

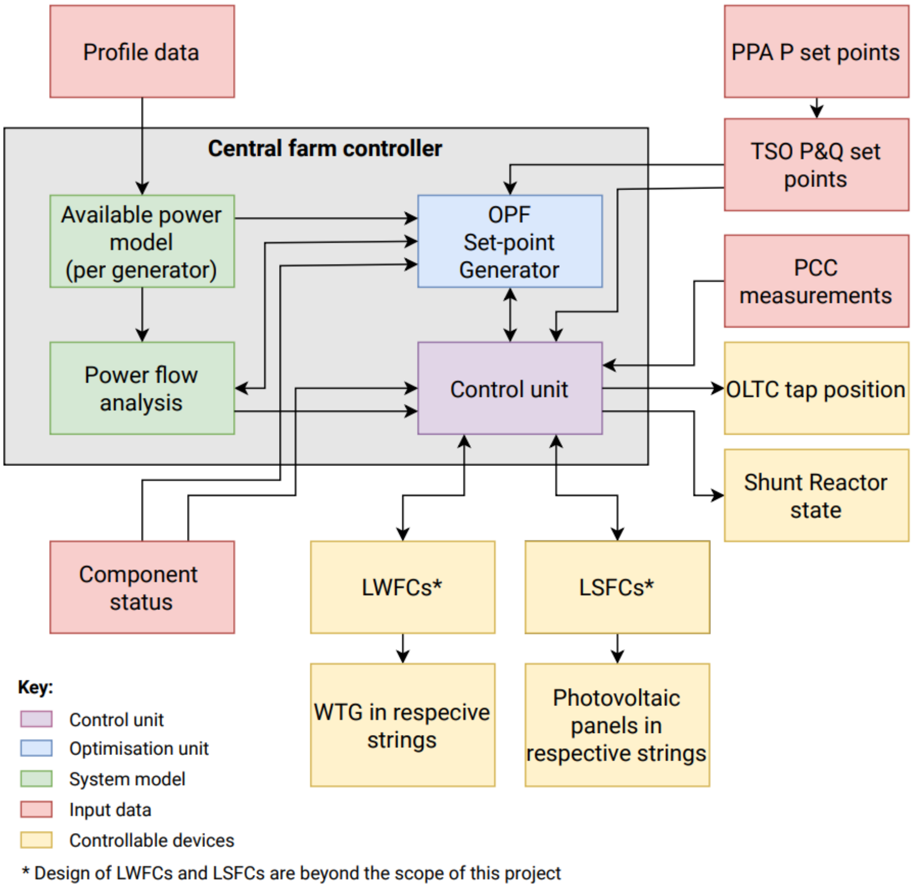


Figure 1.1: Decomposition of the Central Farm Controller

Optimisation Unit

In order to achieve an optimal system configuration, the CFC must be able to compute a combination of optimal setpoints using the system inputs presented in Figure 1.1. This is the responsibility of the *optimisation unit*. The purpose of the *optimisation unit* is to provide the *control unit* with a reference set of optimal setpoints which then ensures that those setpoints are achieved at the controllable devices. Using a certain wind/solar profile, the *optimisation unit* determines the physical boundaries of the control variables and subsequently determines the optimal setpoints such that the requests at the PCC are satisfied and the grid code is enforced. During the process of optimisation, the *optimisation unit* utilises the *power flow model* to determine the feasibility of a combination of setpoints. Therefore, the realistic feasibility of the optimal setpoints is bounded by the accuracy of the *power flow model*.

This project must realise the functioning of the *optimisation unit* as described above. The *optimisation unit* must be able to receive the wind speed, the solar irradiance, information about the system topology and the requested reactive power setpoint Q_{ref} by the TSO and must convert this information into optimal setpoints for the different controllable devices within a farm. These optimal setpoints should result in more economic feasibility in the long run and a more reliable system. Furthermore, the setpoints must lie within the physical boundaries of the system, i.e they must be physically possible and they must be computed every time the TSO gives new requests. The design process of the *optimisation unit* is elaborated in this thesis. For the exact decomposition of the requirements, please refer to Chapter 2.

1.3 Research Goal & State-of-the-Art Analysis

In order to realise the design of the *optimisation unit*, optimal power flow (OPF) problems are considered and especially optimal reactive power dispatch (ORPD) problems. The reactive power that circulates within a power system plays an important role in the real power transfer and voltage stability of that power system. The main objectives of ORPD include the minimisation of active power losses, maximisation of voltage stability and the minimisation of transmission costs [1]. Therefore, the research objective for this thesis concerns the proper formulation of an ORPD problem for the system in question and how to solve it. There are multiple and different types of control variables (the variables can be discrete, continuous or binary) which influence the solution of an ORPD problem. Such problems are researched extensively by the scientific community. In literature, OPF problems are investigated concerning the optimisation of controllable device settings within a WF such as in [2] and [3]. Furthermore, cases are considered in which optimisation is performed within a distribution network such as in [4], [5] and [6]. In the previously mentioned systems the goals of optimisation are for instance the minimisation of economic costs [4] or power losses within the network ([2], [6] and [7]). Other goals include the maximisation of the lifetime of components by minimising transformer tap switches [3] or deviations from system ratings such as bus voltage ratings [6]. Furthermore, the technical constraints of a system must be taken into account when reaching these optimisation goals. This makes the optimisation a high-dimensional, constrained, non-linear and mixed-integer programming problem.

Until now, research has been focusing on the proposal and implementation of various state-of-the-art algorithms to solve previously mentioned OPF problems. However, no investigation has been done about the feasibility of optimisation of controllable devices in a generator farm consisting of both wind and solar modules. Furthermore, no emphasis has been placed yet on the flexibility and extendability of the optimisation problems with variations within a farm's topology, i.e adding and removing generating units and controlling different devices. This goal of this thesis is to add knowledge to this grey area.

The previous mentioned optimal control improvements are beneficial to substation manufacturers who operate in renewable energy generation parks. As mentioned in Subsection 1.1.1, it is becoming more common to connect a SF to a(n), already existing, WF's. This requires more flexibility and robustness of the CFC and therefore also of the *optimisation unit*. The feasibility of the improvements is verified using the case study mentioned in Subsection 1.1.2.

1.4 Thesis Structure

The thesis is structured as follows: In Chapter 2, the program of requirements (PoR) of the *optimisation unit* is presented. Chapter 3 explains the taken design steps and how the design is implemented in MATLAB. In Chapter 4, the performance results of the design are presented and discussed. Finally, a conclusion is given in Chapter 5 together with recommendations for future work.

Chapter 2

Program of Requirements

The central farm controller (CFC) is in charge of maintaining grid compliance. That is, the controller needs to make sure that the requirements drafted by the transmission system operator (TSO) and governmental entities are satisfied. The controller ensures this by sending out setpoints to the local farm controllers (LFCs), transformers and shunt devices. The controller receives measurements at the PCC and LFC's as feedback, needed by the *control unit*. The TSO setpoint requests are described by the requested active (P_{ref}) and reactive power (Q_{ref}) at the PCC, which are refreshed every 15 minutes. The requested active power is usually equal to the maximum possible active power generation at a certain wind speed and solar irradiance. In case of a different active power request, the *control unit* provides the *optimisation unit* with a correct distribution of the active power outputs across the strings such that the active power request is satisfied. Furthermore, the final setpoints sent by the CFC must take the active losses and system maintenance costs into account. The minimisation of active power losses indicates that the controller is efficient and therefore increases its value. The minimisation of hardware maintenance occasions results in less long term maintenance costs. The implementation of this controller is thus beneficial for farm operators. Lastly, in order to guarantee robustness and usefulness of solutions, the requested setpoints should be as far away as possible from the devices' physical limits. From these functional requirements, the following mandatory and trade-off requirements are derived for the *optimisation unit* within the controller:

Mandatory requirements for the *optimisation unit*:

- The computation time of new optimal setpoints must not exceed 15 minutes.
- The computed optimal setpoints must be physically possible. That is, the setpoints must lie within the physical boundaries of the controllable system devices.
- The resulting reactive power (Q_{pcc}) and voltage (V_{pcc}) from the calculated setpoints must lie within the allowed grid boundaries respectively presented in Figures A.1 and A.2.
- The resulting reactive power (Q_{pcc}) from the calculated setpoints must lie within the maximum allowed deviation from $Qref$. This maximum allowed deviation equals 5% of the maximum reactive power capability of the farm [8].
- The computed setpoints must never result in violations of the system's technical constraints. These constraints include maximum bus voltages, branch currents and transformer power flows.

Trade-off requirements:

- The computed setpoints should result in minimal active power losses to increase active power which can be sold.
- The computed setpoints should result in minimal switching of transformer tap positions to minimise maintenance costs.
- The computed setpoints should result in minimal on and off switching of the shunt reactor to decrease system disturbances.
- The computed setpoints of the generator strings should be as far as possible from their physical boundaries to increase system robustness.

The *optimisation unit* consists of a configurable framework built around a meta-heuristic algorithm. This framework must satisfy the mandatory requirements and after configuration, it should meet the trade-off requirements as much as possible. In order to investigate the feasibility and robustness of the proposed *optimisation unit*, the *optimisation unit* must be benchmarked using different configurations and challenging operating conditions. Furthermore, it must be shown that the *optimisation unit* is able to operate when different sources of renewable energy generation are incorporated in the farm. Thus, the *optimisation unit* must satisfy the previously mentioned requirements when the topology in Fig. A.5 consists solely of WTG strings, i.e. the PV generator strings are removed. Thereafter, with minimal adjustments to the *optimisation unit*, the requirements should be fulfilled when the PV generator strings are connected in Fig. A.5.

Chapter 3

Design Methodology

In this chapter, the design methodology of the optimisation unit is presented. While designing the *optimisation unit*, the PoR (Ch. 2) is taken into account. Firstly, a formulation is given of the optimisation problem which the *optimisation unit* needs to solve (Section 3.1). Next, the choice of the optimisation algorithm is elaborated (Section 3.2). Then, it is discussed how the *optimisation unit* is implemented in MATLAB and how it is ensured that the mandatory requirements in Ch. 2 are satisfied (Section 3.3). Finally, an elaboration is given of the approach to arrive at the *optimisation unit's* final configuration (Section 3.4). This configuration is decisive for the satisfaction of the trade-off requirements in Ch. 2.

3.1 The Optimisation Problem

This section presents the optimisation problem which needs to be solved by the *optimisation unit*. This is done by firstly presenting the standard mathematical formulation of an optimisation problem. Thereafter, the mathematical optimisation formulation is applied to the system in question.

3.1.1 Mathematical Formulation

To begin with, the optimisation problem needs to be formulated. Knowing the type of problem allows for a more educated choice of an optimisation algorithm. In fact, the formulation of the optimisation problem is treated extensively in literature (see [2] and [3] for instance). Note that the mentioned literature sources are treating an optimisation problem at park level which is most closely related to the optimisation problem presented in Section 1.2. The mathematical formulation of an optimisation is as follows (from Eqs. 1-4 of [6]):

$$\underset{\mathbf{x}}{\text{minimize}} \quad OF(\mathbf{x}) \tag{3.1a}$$

$$\text{subject to} \quad g(\mathbf{x})_{\text{i}} \leq 0, \text{for } i = 1\ldots, m \tag{3.1b}$$

$$h(\mathbf{x})_{\text{j}} = 0, \text{for } j = 1, \ldots, p \tag{3.1c}$$

$$\mathbf{x} \in \mathbf{X} \tag{3.1d}$$

In Eq. 3.1a, the vector **x** represents the set of control variables. These control variables are bounded by **X** (Eq. 3.1d). **X** is the set containing all possible values of **x**. The $OF(\mathbf{x})$ is the objective function which represents the factors which need to be minimised. Furthermore, Eqs. 3.1b and 3.1c represent the constraints which the solution must meet.

3.1.2 Application to the System in Question

The mathematical formulation consists of various components namely the control/optimisation variables in **x**, the set of possible values of those variables in **X**, the constraints of the system and the OF. The real system must be mapped to these components. Firstly, the control variables in **x** are determined. Please note that these control variables are changed at each optimisation instant (case) and therefore, the equations presented in the optimisation problem must hold for each **x**. The control variables in **x** are dependent on the system topology that is used and which devices of a topology are controlled. In this *optimisation unit*, the following controllable devices are considered: the reactive power output of the WTG and photo-voltaic generator (PVG) strings, the main transformers' tap positions and the shunt reactor. The setpoints for these controllable devices are found in the vector **x**. Firstly, only the reactive power output of the WTG strings is controlled. The controllable variables consist then of the reactive power outputs of the different WTG strings Q_{WTGi}. The vector **x** is then:

$$\mathbf{x} = [Q_{WTG1}, \ldots, Q_{WTG13}] \tag{3.2}$$

In order to minimise active power losses and to maximise the components' lifetime, the transformer tap positions and the shunt reactor should be controlled as well. In order to do this, Eq. 3.2 needs to be extended with extra variables. The two main transformers have multiple tap positions which are described by ($tapT_1$ and $tapT_2$). The shunt reactor can be connected or disconnected which is given by (R). The vector **x** is thus extended as:

$$\mathbf{x} = [Q_{WTG1}, \ldots, Q_{WTG13}, tapT_1, tapT_2, R] \tag{3.3}$$

In the last case, the system topology is changed by adding four photo-voltaic (PV) generator strings. Their reactive power output is represented by (Q_{PVi}). Adding these control variables yields the final vector **x**:

$$\mathbf{x} = [Q_{WTG1}, \ldots, Q_{WTG13}, Q_{PV1}, \ldots, Q_{PV4}, tapT_1, tapT_2, R] \tag{3.4}$$

These control variables are bounded by their physical limits. Each WTG has a limited reactive power range and the same holds for the inverter of the PV strings. This range is dependent on the wind speed and solar irradiation available at the evaluated time instance. Moreover, the transformers have a fixed number of tap positions. Lastly, the shunt reactor is either on (denoted by a 1) or off (denoted by a 0). All setpoints for the controllable devices given in **x** must lie within these limits since it is physically impossible to produce values outside this range. These limits are implemented in the optimisation problem as the bounds of the search space (Eq. 3.1d) and are given by the following set of equations:

$$Q_{WTGi}^{min} \leq Q_{WTGi} \leq Q_{WTGi}^{max} \tag{3.5a}$$

$$Q_{PVi}^{min} \leq Q_{PVi} \leq Q_{PVi}^{max} \tag{3.5b}$$

$$tapT_i^{min} \leq tapT_i \leq tapT_i^{max} \tag{3.5c}$$

$$R \in \{0, 1\} \tag{3.5d}$$

The system constraints include the grid code requirements at the PCC, the maximum allowed deviation from TSO reactive power requests and voltage/current limits within the system. These constraints make the distinction within feasible and infeasible solutions; solutions that result in violations of these constraints are considered infeasible. The main constraint is given by the TSO and concerns the maximum allowed deviation (ϵ) from the TSO request. Furthermore, the reactive power at the PCC ($\mathrm{Q_{PCC}}$) must lie within the grid limits shown in Fig. A.1. The next constraint is that the bus voltages should be within a minimum value and maximum value v_{min} and v_{max}. At the PCC, these limits are dependent on the amount of reactive power delivered to the grid (see Fig. A.2). In the system, the limits are given by the rated voltage of the system devices. If these devices operate at a different voltage, they can be damaged.

There are two other constraints which are due to the capacity limits of the system components: Firstly, the line currents should be below a maximum allowed limit i_{line}^{max}. Secondly, the power flowing through the lines and transformers should not exceed the long term ratings s_{line}^{max}. Violation of these two constraints can lead to system components being damaged or destroyed. Note that there are actually to other constraints which need to be met: nodal active and reactive power balance. However, MATPOWER already takes cares of these two constraints and they will therefore not be considered as constraints for the MATLAB implementation of the *optimisation unit*. The constraints are summarised as follows:

$$|Q_{PCC} - Q_{ref}| \leq \epsilon \tag{3.6a}$$

$$v_{min} \leq v_i \leq v_{max} \tag{3.6b}$$

$$i_{line,i} \leq i_{line,i}^{max} \tag{3.6c}$$

$$s_{line,i} \leq s_{line,i}^{max} \tag{3.6d}$$

Finally, the OF which needs to be minimised is formulated. With regard to the requirements in Ch. 2, the objective function needs to minimise the following parameters: power losses between the strings and the PCC (denoted by P_{loss}), the number of transformer tap switches (denoted by $OLTC_{\text{cost}}$), the usage of the shunt reactor (denoted by R_{cost}) and the distances to the origin of the reactive power outputs of the strings w.r.t. their maximum (denoted by Q_{distance}). These are the trade-off requirements. It makes sense to implement them in the objective function since it is not possible to have all of these requirements minimal. This has to do with the fact that the setpoints leading to the minimum of one objective probably do not lead to the minimum of the other objectives. The objective function has the following form:

$$OF = w_1 \cdot P_{\text{loss}}(\mathbf{x}_t) + w_2 \cdot OLTC_{\text{cost}}(\mathbf{x}_t) + w_3 \cdot R_{\text{cost}}(\mathbf{x}_t) + w_4 \cdot Q_{\text{distance}}(\mathbf{x}_t) \;\; \text{for} \;\; t = 1, \ldots, t_{max} \tag{3.7}$$

In Eq. 3.7, the w_i are the weights of the corresponding objectives and they are used to give more importance to a certain aspect of the multi-objective optimisation problem. The model used for the weights is the Weighted Sum Model (WSM). In this model, the weights of the different objectives add up to 1. This model is only valid under the additive utility assumption. In short, this means that the units of the different objectives need to be the same since otherwise, one would compare apples to oranges (as explained in [9]). The objectives in Eq. 3.7 have different units. To prevent a false comparison, all the values are normalised. For P_{loss}, it means that the p.u. value is taken. For the $OLTC_{\text{cost}}$, the taps are first normalised using the total number of tap changes possible per transformer and then divided by the total number of transformers. This way, when there are maximum tap position changes, the value is equal to 1. The R_{cost} is left as it is since it is a binary value so it is either 1 or 0. Q_{distance} is normalised by dividing the Q per string by the maximum possible value at that time instant Q_{max}. Moreover, the sum is divided by the total number of strings controlled. This results in a 1 when all strings are at their reactive power limit. Normalisation in this way results in the OF having a value between 0 and 1 at all times where a 0 is the best possible value.

In this last part, it is explained how the different objectives of Eq. 3.7 are computed. P_{loss} is given by the power flow calculation which is done using a MATPOWER model of the system in question. For more information about the model, please refer to [10]. The way in which the losses are calculated can be found in Section 9.2.4 of [11]. The total loss is then normalised to its p.u. value in order to have a value between 0 and 1. The $OLTC_{\text{cost}}$ is modelled as in [3] where a larger tap change results in a greater cost. This is due to the increased maintenance and operational costs when changing the tap positions. Moreover, the on-load tap changer (OLTC), which changes tap positions, is 30% of the time the reason of transformer failures (from Fig. 54 of [12]) so also from a reliability aspect, changing tap positions should occur as little as possible. The OLTC cost function is implemented as follows:

$$OLTC_{cost}(\mathbf{x}_t) = \frac{1}{N_T} \cdot \sum_{i=1}^{N_T} \frac{|tapT_{i,t} - tapT_{i,t-1}|}{tapT_{i,range}} \tag{3.8}$$

In Eq. 3.8, $tapT_{i,t}$ and $tapT_{i,t-1}$ are the current and previous tap position of a transformer respectively, N_T is the number of transformers (which for this topology equals 2) and $tapT_{i,range}$ is the maximum number of possible tap changes of a given transformer. This equation does not only consider if there is a tap switch between the current and the previous time instance, but it also implements how much the tap has changed. If the tap positions do not change, the costs are 0 as desired. Moreover, this equation is at most 1 which is the case when there is maximal tap change of all transformers. Minimising the switching of the shunt reactor is also desired. This is due to the fact that the considered shunt reactor has a fixed value. When this reactor is connected, a large amount of reactive power is suddenly injected in the system. This results in a temporary disturbance of the power flows in the system and therefore the on/off switching of the reactor should be minimised. The cost of the reactor is implemented in a similar way to the cost of the OLTC:

$$R_{cost}(\mathbf{x}_t) = \frac{1}{N_R} \cdot \sum_{i=1}^{N_R} |R_{i,t} - R_{i,t-1}| = |R_t - R_{t-1}| \text{ with } N_R = 1 \tag{3.9}$$

In Eq. 3.9, R_t and R_{t-1} represent the current and previous reactor status respectively and N_R is the total number of reactors (in this case this is 1). Lastly, it is desired to have the strings' reactive power output as far away as possible from their maximum reactive power output range at a given profile. In reality, there is turbulence. Turbulence, together with other factors such as wind shear, can impact a turbine's performance [13]. Giving setpoints near the reactive power boundary can result in this reactive power amount not being available due to changing atmospheric conditions. Giving a lower reactive power setpoint can prevent this; a low amount of reactive power is available for more wind speeds. In order to improve the robustness and usefulness of the setpoints, a setpoint is given which lies further away from the reactive power limit. Operating further away from the limits means that the distance is closer to 0 reactive power output. Implementing this requirement in this way is more suitable for the optimisation since the OF and thus its objectives are minimised. This 'extremeness' of the reactive power setpoints is determined as follows:

$$Q_{\text{distance}} = \frac{1}{N_{strings}} \cdot \sum_{i=1}^{N_{strings}} |\frac{Q_i}{Q_{i,max}}| \tag{3.10}$$

The average normalised distance to 0 is denoted by the term Q_{distance}. First, the requested reactive power output Q_i is divided by the maximum operating limit for a given profile $Q_{i,max}$. The absolute value is taken since in this implementation, it is assumed that the reactive power limits are symmetrical around 0. This is done for all strings (indicated by $N_{strings}$). The summation is divided by the number of strings in order to operate in a normalised range and s.t. Q_{distance} is 1 iff. all strings operate at their operating limit. One thing to note on the objective of Eq. 3.10: This objective minimises the distance of the string reactive power output w.r.t. 0. However, the LWFCs take care that these setpoints are reached at the bus bar. This means that it is possible that the LWFCs balance the reactive power output using extreme setpoints within the string. It is presumed that this situation does not occur. This is based on the assumption that it would not make sense to stress the turbines with extreme setpoints since this would result in a shorter lifetime of the components.

3.2 Optimisation Algorithm

In this section, the optimisation algorithm is presented and discussed. Firstly, the process of choosing an adequate optimisation algorithm is elaborated. Thereafter, a description of the algorithm mechanism is given. The types of algorithms used to solve an OPF can be distinguished into two types: classical and heuristic. Classical algorithms use mathematical techniques such as (non)linear programming or Newton's method to achieve the optimal solution [14] whereas heuristic algorithms often use nature-inspired techniques to achieve a (near-)optimal solution [2], [15]. Classical algorithms usually have reduced computational time and are guaranteed to find the best solution. However, they have difficulties when dealing with an optimisation problem type like the one mentioned in Sections 1.3 and 3.1. They require a simplification of the OPF problem formulation which is based on linearised equations of the system model. Therefore the use of classical algorithms for the previous mentioned OPF problem is undesired. On the other hand, heuristic algorithms overcome these difficulties but in return, they attempt to find a near-to-optimal solution (i.e not the best solution, but a sufficiently good solution). Disadvantages of heuristic algorithms are the possibility of local stagnation and premature convergence [3]. However, these could be overcome by adjusting the parameter settings of the algorithm. These features make heuristic algorithms worth investigating for the described OPF problem. The emphasis is placed on the comparison of different meta-heuristic algorithms i.e heuristic algorithms which treat optimisation problems as black-boxes and therefore allow for more flexibility when changes happen in the OPF problem. The choice of an algorithm is discussed and elaborated in Subsection 3.2.1. The working of the chosen algorithm is presented in Subsection 3.2.2.

3.2.1 Choice of Algorithm

The computation speed of the *optimisation unit* is mainly determined by the convergence time of the optimisation algorithm. This is the time needed for an algorithm to find a sufficiently good solution. Therefore, the convergence time of the algorithm must not exceed 15 minutes and is desired to be as short as possible in order to satisfy the requirements mentioned in Chapter 2. However, the convergence speed is a dubious performance measure when comparing the performance of algorithms on different types of OPF problems. This is because the convergence speed depends on factors such as problem characteristics (e.g number of constraints or solution dimension), computation hardware and algorithm parameters. Therefore it is chosen to focus on a meta-heuristic algorithm with a large set algorithm parameters allowing more flexibility and control in terms of solution quality and convergence speed.

It is chosen to implement an improved variant of mean-variance mapping optimisation (MVMO) namely MVMO-SHM as the optimisation algorithm in the *optimisation unit*. MVMO-SHM is a meta-heuristic optimisation algorithm that makes use of a memory archive consisting of the best solutions so far during one optimisation attempt. The algorithm then applies statistical techniques to this archive in order to improve the so far best solution. The procedure of MVMO-SHM is described in Subsection 3.2.2. The choice of MVMO-SHM is based on the fact that the predecessor of this algorithm is proven to be effective for similar OPF problems in terms of convergence characteristics. That is, MVMO is proven to be effective for optimal reactive power management in a near-shore wind farm which contains a similar amount of control variables [2]. Furthermore, the algorithm has a large pool of strategic parameters that influence different stages of the evolutionary process of the algorithm [16]. This allows for flexibility and control in terms of solution quality and convergence speed.

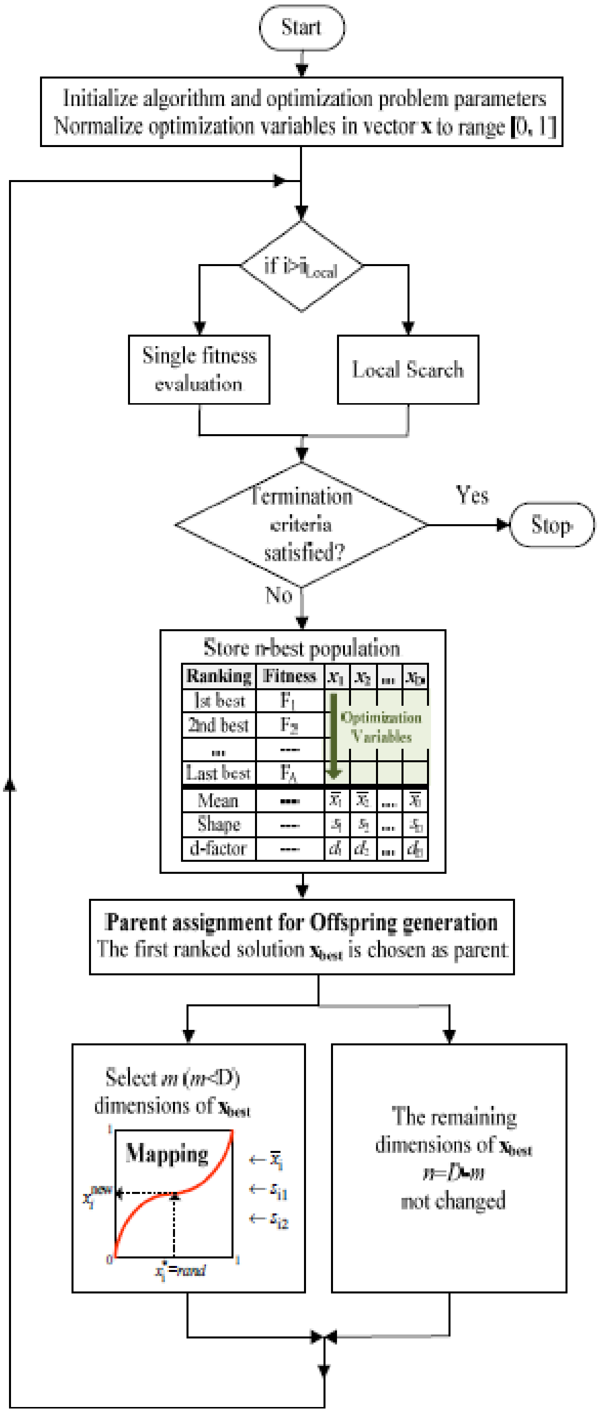
3.2.2 MVMO-SHM

MVMO-SHM is an evolutionary, multi-particle and meta-heuristic algorithm. Each particle of MVMO-SHM is moved throughout the possible solution space in order to find the region where the solution is feasible and more optimal. The aspect that makes MVMO-SHM unique is that it makes use of an on statistics based mapping function for improving the optimal solution throughout its iterative search process [16]. This mapping function uses the mean and variance of the so far N-best solutions in order to steer the general best solution towards a more feasible direction. N is the size of a particle's solution archive. In order to make use of this statistical mapping function, the different optimisation variables are normalised to the range [0,1]. The newly calculated solutions at every iteration are within the [0,1] range and consequently, the optimal solution always stays within the allowed physical boundaries. Thus, the implementation of MVMO-SHM allows for the satisfaction of the second mandatory requirement presented in Chapter 2. Furthermore, the algorithm easily switches between search exploration (i.e searching more globally) and exploitation (i.e searching more accurately in a specific region; usually around the best solutions so far). This decreases the possibility of the solution to be trapped in a local optimum. The algorithm also allows the enabling of an external local search mechanism. This mechanism is computationally expensive but allows the algorithm to generate more feasible solutions within a region.

The iterative procedure of the MVMO-SHM is depicted in Figure 3.1. To begin with, the algorithm and optimisation problem parameters must be initialised. Then, the optimisation variables are normalised using the specified physical min/max bounds. Thereafter, the iterative loop is initiated where the solution in the first algorithm iteration of every particle consists of a vector with random samples from the $U(0,1)$ distribution.

The termination criterion in Figure 3.1 is defined by a pre-specified number of allowed iterations. To clarify, a generation evaluation describes the process of an evaluation of **all** particles. Every particle evaluation is described as one iteration. If the local search mechanism is carried out, a particle evaluation can take up multiple iterations. Then, an intensive search around the first ranked solution of a particle's archive $\mathbf{x_{best}}^{\mathbf{k}}$ is carried out and the solution archive is updated accordingly. When no local search is carried out, the previously generated solution is evaluated on its fitness and the solution archive is updated accordingly.

Thereafter, a new solution vector of the k^{th} particle $\mathbf{x_{new}}^{\mathbf{k}}$ is generated using the hitherto best solution $\mathbf{x_{best}}^{\mathbf{k}}$ and a mapping function that uses the statistical properties of the solutions within this particle's solution archive. The mapping function is applied to *m* selected variables in the best solution vector ($\mathbf{x_{best}}^{\mathbf{k}}$) with $m \leq D$, where *D* is the number of optimisation variables. The mapping function uses the means and variances of these *m* variables (from the solution archive) to generate a new solution $\mathbf{x_{new}}^{\mathbf{k}}$. The mapping function is described and analysed extensively in [16]. Since understanding this mapping function allows for better understanding and implementation of the MVMO-SHM algorithm, a behavioural analysis of the mapping function is presented.


Start
Initialize algorithm and optimization problem parameters
Normalize optimization variables in vector x to range [0, 1]
if i>i_Local
Single fitness evaluation
Local Search
Termination criteria satisfied?
Yes
Stop
No
Store n-best population
Ranking
Fitness
x_1
x_2
...
x_D
1st best
F_1
2nd best
F_2
...

Last best
F_A
Optimization Variables
Mean
Shape
d-factor
Parent assignment for Offspring generation
The first ranked solution x_best is chosen as parent
Select m (m<D) dimensions of x_best
Mapping
x_i^new
x_i^*=rand
← x̄_i
← s_i1
← s_i2
0
1
The remaining dimensions of x_best
n=D-m
not changed

Figure 3.1: Procedure of the MVMO-SHM Algorithm (Figure 1 from [16]).

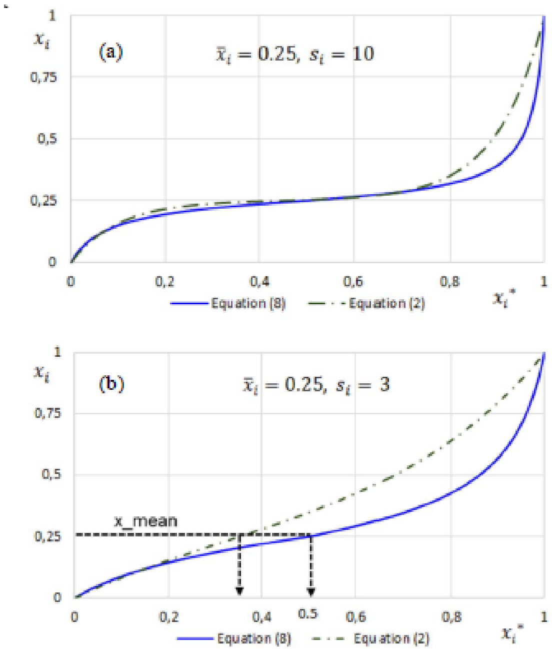


Figure 3.2: Mapping Function Behaviour (Figure 3 from [16])

Figure 3.2 describes how the mapping function behaves for different shape factors s_i. The shape factor is a controllable parameter within the meta-heuristic algorithm. Each solution **x** contains multiple variables; one variable is denoted as x_i. The newly generated variable x_i depends on random samples from the $U(0,1)$ distribution (x_i^*), the shape factor (s_i) and the mean ($\overline{x_i}$). For a bigger s_i, the newly generated variables are more likely to be mapped around the mean, creating new solutions that are close to the solutions stored in the solution archive. For a smaller s_i, the newly generated variables are more dependent on the random sample x_i^*, creating new solutions that are further away from the solutions in the solution archive. Thus for bigger s_i, the algorithm searches more accurately around the hitherto best solutions while for smaller s_i, the algorithm searches more globally. Equations 3.11a, 3.11b and 3.11c from [16] describe the algorithm parameters that can influence the solution search during the optimisation process.

$$s_\text{i} = -ln(v_\text{i}) \cdot f_\text{s} \tag{3.11a}$$

$$\text{with} \quad f_\text{s} = |f_\text{s0} \cdot [4 + 1.65(rand - 0.15)]| \tag{3.11b}$$

$$\text{and} \quad f_\text{s0} = f_\text{s\_ini} + k \cdot (f_\text{s\_final} - f_\text{s\_ini}) \tag{3.11c}$$

Equation 3.11a shows the expression of the shape factor, where v_i is the variance of the i^th variable and f_s is the scaling factor. The expressions corresponding to the scaling factor are given in Equations 3.11b and 3.11c in which *rand* is a random sample from the $U(0,1)$ distribution, *k* is the relative iteration count (i.e iteration count divided by the number of allowed iterations) and f_{s_ini} and f_{s_final} are tunable algorithm parameters that influence the initial and final values of f_s. Thus, the scaling parameter f_{s_final} allows s_i to be adjusted automatically throughout the optimisation process. At the beginning of an optimisation process, the algorithm is desired to search more globally such that more (feasible solution) regions are explored. At the later stages of the process, it is desired to search more accurately i.e to search the best solutions within the feasible solution regions. This can be realised by tuning f_{s_ini} and f_{s_final} such that s_i gradually increases throughout the optimisation process. Other parameters that influence the optimisation process and consequently the solution quality include the solution archive size, number of particles, maximum allowed iterations, number of variables to be selected for mutation and the local search mechanism threshold. [17] suggests that an archive size of 2-5 is sufficient. The local search mechanism threshold consists of a minimum amount of iterations and a probability of application after the minimum amount of iterations has been reached. The method for determining the best parameters is discussed in Subsection 3.4.2.

3.3 Implementation

In this section, the breakdown of the implementation of the *optimisation unit* is discussed. The *optimisation unit* is implemented and tested in MATLAB. A flowchart of the *optimisation unit* is found in Fig. A.6. The goal of the *optimisation unit* is to provide the *control unit* with a set of optimal setpoints every time the operating conditions are updated, that is every 15 minutes. To clarify: the 'setpoints' are named 'solutions' within the *optimisation unit*. This is due to the fact that these setpoints essentially are solutions to the formulated optimisation problem. This section mainly focuses on the working and interaction of the different subcomponents of the *optimisation unit* that are presented in Fig. A.6.

In Subsection 3.3.1, the initialisation of the *optimisation unit* before starting an optimisation process is discussed. Subsection 3.3.2 presents the interaction between the algorithm and the *optimisation unit*. In Subsection 3.3.3, the implementation of the optimisation problem is elaborated. Lastly, Subsection 3.3.4 explains the selection of the final solution.

3.3.1 Initialisation

Before the process of optimisation commences, the system configuration is initialised. The *optimisation unit* receives the operating conditions of the park at a certain time. This is denoted by the 'Measurements' input in Fig. A.6. These operating conditions consist of the wind/solar profile, requested reactive power amount by the TSO and an up-to-date farm topology. The latter is described by a MATPOWER case file which represents a model of the farm topology in matrix notation. The topology is updated when faults in the farm occur. Certain devices/components in the farm are then disconnected and the topology must be updated accordingly. Now and then, the TSO can request an amount of active power that is below the maximum available generation at a certain moment. In this case, the *optimisation unit* receives the desired active power dispatch of the generating strings from the *control unit*, such that the TSO demand is met. In the common case, the active power dispatch of the strings equals the maximum amount of active power that they can generate at that moment. Based on the given operation conditions, the active power generations of the strings are entered in the MATPOWER case file. This file is used by the fitness evaluation component to perform power flow runs. Furthermore, the boundaries of the optimisation variables are calculated. These boundaries correspond to the physical limitations of the system devices which are controlled and are used by the optimisation algorithm to generate solutions. The boundaries of reactive power outputs of strings depend on their active power outputs and should, therefore, be calculated for every update of operating conditions. While the possible reactive power outputs are continuous between their boundaries, the optimisation variables corresponding to the transformer taps and reactor status can only take discrete and binary values, respectively. The boundaries corresponding to these discrete values are fixed at all times, given that these devices are in service. The boundaries of the optimisation variables are stored in vectors that follow the same sequence as the vector presented in Equation 3.3.

3.3.2 Algorithm Interaction

The implemented algorithm is MVMO-SHM. This algorithm belongs to the superior class of meta-heuristic algorithms. This is backed up by the fact that MVMO-SHM won the world competition on computational expensive problems of the IEEE Computational Intelligence Society in 2018. The source code of this algorithm was given by this group's supervisor and contains the appropriate acknowledgements. It was chosen to use a source code since the fundamentals behind an algorithm are not part of the EE curriculum and it is possible to use more advanced features of the algorithm. However, note that the algorithm can be edited so desired adjustments can easily be made.

Once the algorithm is called, it generates a random initial solution. Then, it evaluates how good this solution is. For this, it calls the fitness evaluation function, a function which represents the objective function of the optimisation problem. A smaller fitness value corresponds to a better fitness. The fitness evaluation process is elaborated in Subsection 3.3.3. Throughout the evaluation of the objective function, the algorithm takes a solution, evaluates its fitness and mutates the solution vector towards the feasible regions. The best fitness value is saved and updated when a new solution yields a better fitness. After the maximum number of iterations, the algorithm stops and the best solution is returned. The algorithm is limited to a certain amount of iterations in order to reduce the computational impact of the algorithm. Note that due to the meta-heuristic nature of the algorithm, this solution is not necessarily the best possible solution but a sufficiently good solution. It is possible that another run yields a better solution. Therefore, it is recommended to evaluate a profile multiple times, i.e the algorithm is run multiple times for a given wind/solar profile. This is also possible since the computational speed of the algorithm allows for multiple runs within the allowed 15 minutes limit. This does not eliminate the possibility of the existence of a better solution but it decreases that probability. The selection of the best solution out of multiple runs is discussed in Subsection 3.3.4.

3.3.3 Fitness Evaluation

In order to determine what the feasible solutions are (i.e. solutions which do not violate constraints), a fitness evaluation function is needed. This function uses the currently proposed solution (in the form of Eq. 3.4) to evaluate how feasible this solution is regarding the constraints and objectives and it should differentiate between feasible and infeasible regions. A solution is considered infeasible when either:

a) The MATPOWER power flow computation does not converge.

b) One or more constraints of Eqs. 3.6a-3.6d are violated.

To determine this, the function first updates the controllable devices in the MATPOWER casefile with the proposed solution. One thing needs to be noted about this: The algorithm does not differentiate between continuous and discrete variables. Since some of the devices in the system are discrete, those variables are first rounded to their closest discrete value. For the transformers, these are the different taps or turn ratio's corresponding to the taps. For the reactor, this is either on or off. Rounding the variables allows for mixed-integer optimisation. After this step, it is possible to run the power flow computation with MATPOWER. If a solution does not yield convergence of the power flows, this means that the power flow equalities are not satisfied. A large penalty is then given to the fitness. If a solution yields convergence, it is checked whether constraints are violated or not. The constraints from Eqs. 3.6b and 3.6c are obtained from the system topology. The value of $s_{line,i}^{max}$ is taken from the long term rating of the lines. Even though temporarily it might be allowed to have a higher power flowing through a line, the *optimisation unit* must give long term solutions and not stress the system when possible. If at least one constraint is violated, the solution is considered to be infeasible. This is done to ensure that no physical limits are violated. The fitness value still receives a large penalty multiplied by the number of violations but the number is smaller than in the previous case. This is to help the algorithm to differentiate between more and less infeasible solutions and steer its solution to the feasible regions. If there are no violations, it means that the solution is feasible. In that case, the fitness equals the value of the OF. If a solution is better, it means that the value of the objectives is also smaller and this means that the fitness is also smaller. This allows for differentiation between more and less feasible solutions. The fitness function is thus implemented as:

$$F(\mathbf{x}) = \begin{cases} p_2 \cdot S(\mathbf{x}) & S(\mathbf{x}) \neq 0 \\ p_1 \cdot CV(\mathbf{x}) & S(\mathbf{x}) = 0 \text{ and } CV(\mathbf{x}) \neq 0 \\ OF(\mathbf{x}) & \text{otherwise} \end{cases} \tag{3.12}$$

In Eq. 3.12, $OF(\mathbf{x})$ is the objective function of Eq. 3.7, $CV(\mathbf{x})$ is a function which computes the constraint violation and $S(\mathbf{x})$ is a function which is used to penalise unsuccessful power flow computations. p_1 and p_2 are penalty factors with $1 \ll p_1 \ll p_2$ in order to differentiate between constraint violations and optimisation objectives. p_1 is set to 10^{20} and p_2 is set to 10^{50}. When there is no power flow convergence, $S(\mathbf{x})$ is set to 1 and otherwise, it is 0. $CV(\mathbf{x})$ implements the constraints of Eqs. 3.6a-3.6d. For the bus voltages (Eq. 3.6b) and branch limits (Eqs. 3.6c and 3.6d), it is looked at how many violations there are. For the accuracy of the amount of reactive power at the PCC (Eq. 3.6a), it is looked at the magnitude of the difference. If this difference is between the threshold ϵ, the magnitude of the difference is rounded to 0. This is done in order to have $CV(\mathbf{x})$ equal to 0 when no constraints are violated. The implementation of the fitness as in Eq. 3.12 has a major advantage: it allows for unconstrained optimisation. By implementing the constraints into the fitness directly, it is possible to evaluate and minimise a single function. It also allows for an intuitive interpretation of the fitness; the order of the fitness value is different for different kinds of violations. This approach allows for a distinction between setpoints that do and do not satisfy mandatory requirements 2-5 in Ch. 2. In case the optimal setpoints satisfy the requirements, the resulting fitness value $F(\mathbf{x_{optimal}})$ equals the value of the objective function $OF(\mathbf{x_{optimal}})$; otherwise, it equals a big penalty value.

3.3.4 Selection between Solutions

The *optimisation unit* stores the results from the different algorithm runs in a results buffer. This buffer is emptied every 15 minutes when the operational conditions are updated. It contains the calculated optimal solutions of the different runs and their corresponding fitness. Furthermore, other results of the power flows, such as the active power loss, are also stored in the buffer for each optimal solution. The best solution for a given set of operating conditions is solely defined by its fitness value. This is due to the fact that the fitness value already incorporates the different minimisation objectives. Thus, the solution which results in the lowest fitness is selected as the best solution. This solution is then sent to the *control unit* as the set of optimal setpoints.

3.4 Research Approach on the Trade-off Requirements

The implementation discussed in the previous section is mainly related to the mandatory requirements of Chapter 2. These requirements must always be satisfied, independent on the scope of optimisation. In this section, the approach for satisfying the trade-off requirements is discussed. These are the requirements which are satisfied as much as possible but a perfect combination of them does not exist since there are interdependencies.

The performance of the *optimisation unit* depends on multiple aspects: the performance of the algorithm, which control variables are optimised and the optimisation objectives. In this section, it is discussed how to arrive at the final configuration of the *optimisation unit*. The first task is to define a way to measure the performance of the *optimisation unit*. For this, the Key Performance Indicators (KPIs) are created. These are presented in Subsection 3.4.1. Hereafter, it is discussed what approach is used to determine the final configuration of the *optimisation unit* such that the trade-off requirements are also satisfied as much as possible. In order to do this, four steps are needed. In the first step, the algorithm parameters are tuned. In the next two steps, the optimal configuration of the *optimisation unit* is determined. In the last step, the performance of the unit is investigated under an extended topology. These steps are elaborated in Subsection 3.4.2. Lastly, a test profile is created. This is done to ensure that the *optimisation unit* is able to perform under different operating conditions. This benchmark is presented in Subsection 3.4.3.

3.4.1 Key Performance Indicators

In this section, the key performance indicators (KPIs) are discussed. These indicators show how well the *optimisation unit* performs. These indicators are used to discriminate between the different solutions and results obtained in the research on the trade-off requirements. To determine how well the *optimisation unit* works, different KPIs are used. These KPIs can be divided into two different parts: The first part is an indication of how well the unit as a whole performs. The second part is related to the solutions the *optimisation unit* yields. The KPIs are:

For the complete unit:

- The average runtime per run. Lower is better since this allows for faster results of the *optimisation unit*.
- The number of runs yielding a significant solution per case. Higher is better since this means that the *optimisation unit* is able to find solutions in more cases and is thus more robust.

For the solutions given by the unit:

- The best fitness value. Depending on the objectives of the OF, this value indicates how much the solution is in line with the objectives. As explained in Section 3.1, a lower OF value is better.
- The variability of the solutions for different runs of the same case. A small variability indicates that there is a higher chance to have solutions around the global optimum of the optimisation problem.
- The variability of the fitness value; lower is better. As with the variability of the solutions, a small variability ensures that all runs yield more or less the same results and thus, the solutions of the *optimisation unit* are robust.

Two small remark needs to be made on the KPIs: Firstly, it is possible that the variability of the solutions is high. If the fitness value is still the same, this can indicate that the optimisation problem has no global optimum but only local optima with similar OF values. This indicates that there are multiple operation regions which yield the same results for the different objectives. On the other hand, high variability in the solutions may also indicate that the algorithm is not tuned properly and is easily trapped in local optima. To determine which of the two is the case, proper algorithm parameter determination needs to be performed. Secondly, the KPIs are mostly used to determine the performance and robustness of the *optimisation unit*. For steps 2 and 3 of Subsection 3.4.2, these indicators are not very significant. This has to do with the fact that in those cases, the absolute value of the objectives is important and not the deviations in the objectives. Nevertheless, the KPIs and thus the deviations in the objectives are considered when comparing the results.

3.4.2 Steps to Determine the Final Configuration

Now that the KPIs are known, the four steps are presented which will yield the final configuration of the *optimisation unit*:

Step 1: Algorithm Parameter Tuning

In this step, the optimal configuration of the algorithm is determined. This configuration is determined by means of a parametric sensitivity analysis. This means that certain algorithm parameters are swept individually while leaving the others fixed. In order to carry out the sensitivity analysis, the following assumption is made: The influences of different parameters on the results are independent. That is, when variations in the results (KPI) occur as a consequence of the parameter sweep, these variations are entirely due to the parameter that is swept and not due to other parameters. If this assumption is not made, the sensitivity analysis would be immensely time- and power-consuming. The parameters of the algorithm can be summarised as:

- Number of particles: $N_{particles}$
- Solution archive size: $N_{archive}$
- Maximum allowed iterations: i_{max}
- Number of mutated variables: initial (m_{ini}) and final (m_{final}) value
- Scaling parameter: initial (f_{s_ini}) and final (f_{s_final}) value
- Minimum amount iterations before enabling local search: i_{local}
- Local search probability: p_{local}

Empirical analysis showed that the local search mechanism is too time-consuming while yielding no significant improvements in the results. Therefore the use of the local search mechanism is excluded in this optimisation problem. This sensitivity analysis is desired to be carried out every time the optimisation problem dimension changes. However, due to the power/time-consuming nature of this sensitivity analysis, this is not done.

The parameter sweeps are carried out by optimising one challenging profile. The optimisation process is repeated 10 times for a total of 10 runs. The following operation conditions apply:

- Only the 13 WTG strings are controlled.
- v_{wind} = 7 m/s.
  At this wind speed, the strings allow for the biggest range of reactive power outputs. This corresponds with the biggest possible solution search space for the and consequently finding an optimum will be harder.
- $Q_{request}$ = -100 MVAr (-0.286 p.u).
  It is empirically found that for different operating conditions, the branches within the topology consistently result in a positive line charging injections; the injection is approximately 40 MVAr at the PCC. Consequently, negative TSO requests are harder to satisfy than positive requests. However, by giving this setpoint, it can be guessed where the optimal solution lies within the solution space: the optimal string outputs should be close to their min. boundaries.

This challenging profile requires for the algorithm to go to the boundary of the search space i.e. reactive power outputs close to the (negative) boundary of the WTG strings. Since the initial solution is chosen randomly, it is possible that the solution lies in the very positive end of the search space. If the algorithm of the *optimisation unit* is still able to find a feasible solution, it will also be possible to do this for all the other profiles as well since the search space is either smaller or less challenging.

Step 2: Choosing the Variables

In this step, it is decided which controllable devices are used in the optimisation. This is done by comparing the active power losses for different sets of control variables. There are several devices available in the system in question: the WTG and PVG strings, the on-load tap changer of the transformers and the shunt reactor. In practice, the CFC only controls the reactive power outputs of the strings. However, throughout literature, OLTCs and shunt reactors are also controlled while minimising the active power losses ([2] and [3]). The results were that not controlling these devices yielded an inferior solution. Probably, this is due to the fact that part of the search space is already bounded by an external factor which is not controllable. It is therefore desired to determine whether controlling these devices has a positive influence on minimisation of the active power losses.

Step 3: Determining the Weighting of the Objectives

In this step, the exact weighting of the different objectives is determined. The goal of this weighting is to satisfy the trade-off requirements in Chapter 2. In Eq. 3.7, each objective is weighted by a weight w_i. These weights determine the relative importance of the different objectives. Note that the objectives considered in this step are dependent on the controllable variables chosen in the previous step. For instance, if it is decided not to control the shunt reactor, it is impossible to minimise for reactor switching since the control is out of reach for the CFC. The main objective is a reduction in active power losses. Besides that, it is desired to reduce maintenance costs as well. Presumably, it is impossible to find a solution which minimises all these objectives. This has to do with the fact that the solution which minimises one objective completely is not the same as the solution which does this for another objective. Therefore, the interplay between the different objectives needs to be determined. For instance, it is possible that applying a higher weight to one objective (e.g. the changing of the transformer tap positions) results in only a minor increase of active power losses. In this case, it would be desirable to weight the first objective higher since a lot of maintenance could be saved in this way.

Sweeping the weights can quickly result in a lot of optimisation runs. Therefore, several choices and assumptions are made to reduce the computational effort: Firstly, the different cases are run only once. This is done since the *optimisation unit* already has a small variability in solutions and fitness (as can be seen from the standard deviation in Appendix B.1). It is therefore assumed that one run yields values comparable to those of multiple runs. The best solution is probably not found but in this section, the relative effect of the different objectives is of interest. Therefore, it is considered acceptable to use only one run.
Secondly, not all cases of the test profile are considered. For the first 15 cases, the improvement in active power losses is small when adding controllable devices. Controlling the transformer tap positions yields a large improvement in active power losses for cases 21-24 (Fig. 4.2). However, adding restrictions to the controllable devices could reduce this improvement. To check how the active power losses are reduced when reducing the other objectives as well, these cases with large improvement potential are considered.
Thirdly, the main objective is considered to be a reduction in costs. Reduction of reactor switching and the requested amount of reactive power from the strings are more important from a reliability and robustness point of view. An imbalance in power flows caused by switching the reactor is undesirable but is usually resolved pretty quickly. Moreover, requesting reactive power amounts near the WTG strings their boundaries can result in that range being unavailable by changing atmospheric conditions within the 15-minute interval. However, simulations demonstrated that the setpoints are usually far from the boundaries except when a negative setpoint is requested. Then, the park must generate large amounts of negative reactive power in order to compensate for the branch injections. Also, should the setpoint become unavailable due to e.g. turbulence, there are still other control mechanisms in the *controller unit* that are able to cope with these fluctuations. This demonstrates that the importance of the last two objectives is lower than that of the first two objectives. In order to reduce the computational effort, it is chosen to keep w_3 and w_4 fixed and to have a small value.

In order to compare the results of the different weight combinations, a total cost is computed:

$$Cost(\mathbf{x}_t) = c_1 \cdot t_{passed} \cdot P_{loss}(\mathbf{x}_t) + c_2 \cdot taps(\mathbf{x}_t). \tag{3.13}$$

In Eq. 3.13, c_1 and c_2 are the costs of the power loss and tap switching respectively, t_{passed} indicates how much time there is between two cases, $P_{loss}(\mathbf{x}_t)$ is the active power loss in MW and $taps(\mathbf{x}_t)$ is the number of tap switches. c_1 is taken to be around 80 € per MWh which is just above the cost price for producing a MWh using wind turbines. t_{passed} is needed to determine how much time has passed (in hours) and is used to determine how much MWh losses there is at time t. The exact value of c_2 is unknown but is chosen in the range of [2] and [3], namely 2 € per tap switch. Only the first two objectives are directly related to cost; minimising reactor switching and maximising the distance of the setpoints to their boundary is done in order to improve the stability and robustness of the *optimisation unit*. Therefore, the latter two objectives are omitted from Eq. 3.13.

Step 4: Determining the Flexibility and Robustness of the *Optimisation Unit*

In the previous two steps, the final balance between the trade-off requirements has been established. Whereas the previous steps focused on the performance of the *optimisation unit* for the WF topology, this step will focus on the flexibility and robustness of the unit when expanding the system. For this, the PVG strings in Fig. A.5 are connected and the optimisation process is run. The scope is to determine whether it is still possible to find feasible solutions. Moreover, it is desired to determine whether these solutions are good and whether tuning of the parameters yields an improvement of the solutions.

3.4.3 Test Profile

In order to ensure that the *optimisation unit* is able to operate under different profiles, a test profile is created. The goal of this test profile is to simulate a wide range of possible operating conditions. This is done in order to demonstrate the general feasibility and robustness of the *optimisation unit*. The test profile of Fig. 3.3 contains different reactive power setpoints at different wind speeds and solar irradiances. The operating conditions corresponding to the cases within the test profile are presented in Table A.1. The requested amount of reactive power is based on the usual setpoints given by the TSO. The step size between the different amounts of reactive power is 50 MVAr (from [8]). A setpoint of 0 MVAr is usually required, $\pm$ 50 MVAr ($\pm$ 0.143 p.u) occurs and in rare cases, $\pm$ 100 MVAr ($\pm$ 0.286 p.u) is requested. These setpoints are requested for the different environmental conditions. As can be seen in Fig. A.4, the reactive power capability range of the turbines changes significantly with different wind speeds. For the test profile, five different wind speeds are chosen namely 4.5, 5, 7, 12 and 15 m/s. Each wind speed is from a different reactive power capability region. The first two (4.5 and 5 m/s) are from the region after cut-in wind speed where the amount of reactive power produced increases cubically. At 7 m/s, the range of reactive power is at its maximum. At 12 m/s, the amount of reactive power is decreasing; this is due to the capability curves of the WTGs. At 15 m/s, the turbines operate at rated power and the reactive power range does not change anymore. It must be noted that, unlike the WTG strings, the reactive power capabilities of the PVG strings are proportional to the solar irradiance. Furthermore, the PVG strings do not allow for absorption of reactive power [18].

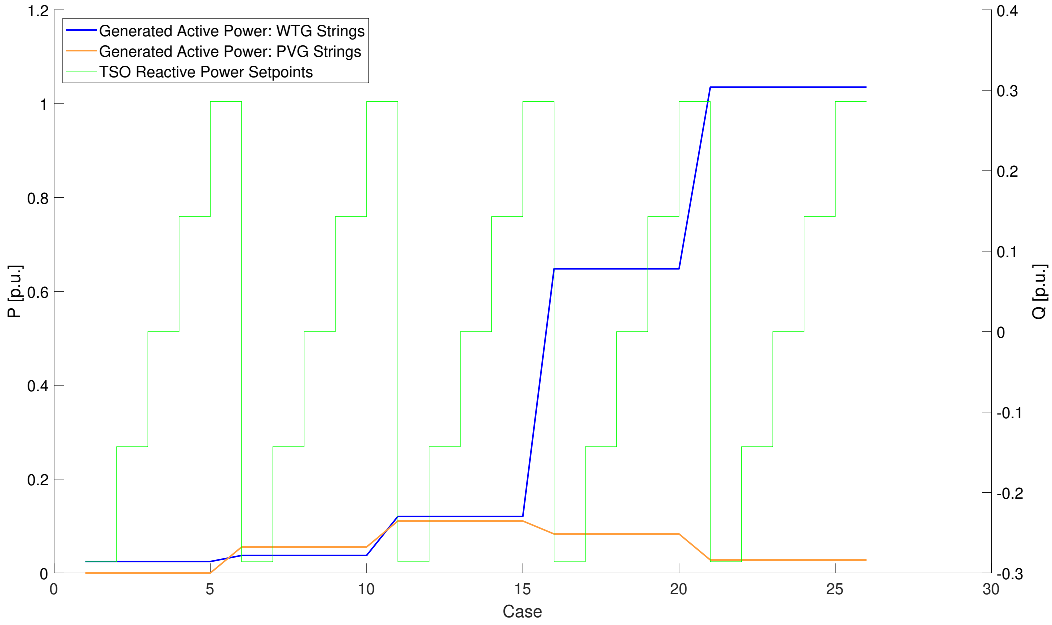


Figure 3.3: Test Profile for the *Optimisation Unit*

The chosen wind speeds represent a wide variety of ranges of the available amount of reactive power. Together with the different reactive power setpoints, a lot of challenging scenarios are covered. Some things need to be noted about the test profile: Firstly, at low wind speeds, the reactive power range is increasing but fairly small. This can result in the farm not having enough reactive power available to satisfy very negative setpoints. Another possibility is that the farm does have enough (negative) reactive power available, but due to branch injections of (positive) reactive power it is impossible to satisfy the negative setpoints. The second thing which should be noted is that a larger range increases the search space enormously. This has to do with the optimisation problem being multidimensional. The ability to find solutions in a large search space proves that the *optimisation unit* is robust enough to find a solution and is able to deal with the problem accordingly. The third thing to mention is that this test profile is only used in steps 2-4 of Subsection 3.4.2. This has to do with the fact that parameter tuning must be done in the hardest search space possible; this corresponds to a setpoint of -100 MVAr and a wind speed of 7 m/s. This results in the largest search space and if the *optimisation unit* is able to find a solution there, it is able to do so for all cases (provided a solution exists). The last thing which should be noted is that the test profile is fairly big. It has 25 cases and if each case is run multiple times, the total amount of runs needed soon exceeds 100. Therefore, in order to save computational time only certain regions of interest may be investigated. If this is the case, it will be mentioned together with a motivation.

Chapter 4

Results and Interpretation

In this chapter, the results that lead to the final configuration of the *optimisation unit* are presented and discussed. To begin with, it is described how the optimal algorithm parameters are found (Section 4.1). Secondly, the potential feasibility of the *optimisation unit* is discussed and it is investigated how this feasibility varies when different sets of system devices are controlled (Section 4.2). Thirdly, an elaboration is given of the selection process of the different objectives' weights (Section 4.3). Furthermore, a verification of the *optimisation unit's* flexibility and robustness is given (Section 4.4). This is accomplished by demonstrating the working of the *optimisation unit* when the system's topology is extended with controllable PVG strings. Finally, it is discussed how the *optimisation unit* would perform and operate in a physical prototype (Section 4.5).

4.1 Algorithm Parameters

In this section, the values of the algorithm parameters are determined. The following algorithm parameters are considered to have an impact on the performance of the *optimisation unit* and are therefore taken into account:

- Number of particles: $N_{particles}$
- Solution archive size: $N_{archive}$
- Maximum allowed iterations: i_{max}
- Number of mutated variables: initial (m_{ini}) and final (m_{final}) value
- Scaling parameter: initial (f_{s_ini}) and final (f_{s_final}) value

The following suggestions for the algorithm parameter values are given by this group's supervisor and is deemed to be most feasible:

- $N_{particles}$: 1-200 particles (default: 1)
- $N_{archive}$: 2-5 solutions (default: 4)
- i_{max}: 500·$\mathrm{N_{particles}}$ (default: 500·$\mathrm{N_{particles}}$)
- m_{ini}: 0.5D-0.75D (default: 1D) and m_{final}: <0.5D (default: 0.09D) where D is the total number of optimisation variables
- f_{s_ini}: 1 (default: 1) and f_{s_final}: 1-20 (default: 2)

The results of the parameter sweeps are presented in the tables in Appendix B.1. It is found that the following algorithm configuration yielded the best KPI results and therefore, is most feasible:

- $N_{particles}$: 35 particles
- $N_{archive}$: 3 solutions
- i_{max}: 17500 iterations
- m_{ini}: 0.54D and m_{final}: 0.31D where D is the total number of optimisation variables
- f_{s_ini}: 1 and f_{s_final}: 1

The population size sweep initially consisted of the following values: 1, 5, 20, 50, 100 and 200. However, it was found that the values of 20 and 50 were relatively more feasible in terms of the KPIs. Therefore it is decided to sweep more values between 20-50, namely 35, 40 and 45. The value of 35 is chosen because it yielded a significant smaller variability in fitness and solution values than a population size of 20, only at the cost of runtime. The latter is not a problem since the average runtime for 35 particles equals 86 seconds and thereby satisfying the first mandatory requirement in Ch. 2. Larger population sizes would not yield significant improvements but would require more time. In a similar fashion, the other parameters are determined.

It is noteworthy that (minor) differences within the average runtimes of the other sweeps are possibly caused by the operating conditions of the hardware on which the simulations are carried out. Therefore, the average runtime becomes an inaccurate performance measure when the average runtimes are in a similar range. Consequently, the remaining optimal parameters were selected based on their fitness values and the corresponding variabilities.

4.2 Feasibility

In this section, the feasibility of the optimisation is determined. Firstly, only reactive power setpoints are computed. If optimisation is feasible, it is investigated whether the feasibility of optimisation can be improved by controlling different devices.

4.2.1 Optimisation vs No Optimisation

Fig. 4.1 shows a comparison of the active power losses with and without optimisation for the different operating conditions within the test profile. This comparison is carried out when the control variables only consist of the reactive power outputs of the WTG strings. With 'No Optimisation' it is meant that the *optimisation unit* does not minimise the active power losses. This is achieved by setting w_1 equal to 0 and thereby removing the objective of minimal active power losses. When the optimisation mechanism of the *optimisation unit* is disabled, the fitness function equals the following:

$$F(\mathbf{x}) = \begin{cases} p_2 \cdot S(\mathbf{x}) & S(\mathbf{x}) \neq 0 \\ p_1 \cdot CV(\mathbf{x}) & S(\mathbf{x}) = 0 \text{ and } CV(\mathbf{x}) \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

By implementing this fitness function, the algorithm within the *optimisation unit* searches for solutions that satisfy system constraints but do not result in minimal active power losses. This approach is used as an approximation of setpoints that a prevailing CFC would give. This approximation is necessary to quantify the feasibility of the *optimisation unit* since MATLAB models of those CFC's are unavailable. Furthermore, it must be noted that the losses in cases 1,2 and 6 are equal to 0. It was found that in these cases, the TSO requests were outside the physical limitations of the farm (see also Figure A.4). That is, for those operating conditions the farm is unable to generate/absorb enough reactive power to satisfy those requests. It is assumed that in reality, the TSO does not request setpoints that are outside the capabilities of the farm.

It can be seen in Fig. 4.1 that lower active power losses are achieved when the optimisation mechanism is enabled. However, the amount of loss reduction varies substantially for different operating conditions. This verifies the contribution of the *optimisation unit* to an increase in transmission efficiency from WTG strings to the PCC. Furthermore, it is noticeable that the relative loss reduction (i.e loss reduction divided by the windfarm's total active power output) is the largest for cases 11-15; this follows from Figures 4.1 and A.3. These cases correspond with a wind speed of 7 m/s and according to Fig. A.4, this wind speed gives the largest possible reactive power capability of the farm. This implies that at 7 m/s, the WTG strings have the biggest possible range of reactive power outputs and consequently, the min/max bounds of the optimisation variables are also the largest possible. Thus, at 7 m/s the algorithm searches solutions in the largest possible search space. A possible explanation for the large relative loss reduction at 7 m/s is that a larger solution search space results in more feasible solution regions and consequently there are more solution regions that result in better solutions, i.e solutions that result in lower active power losses.

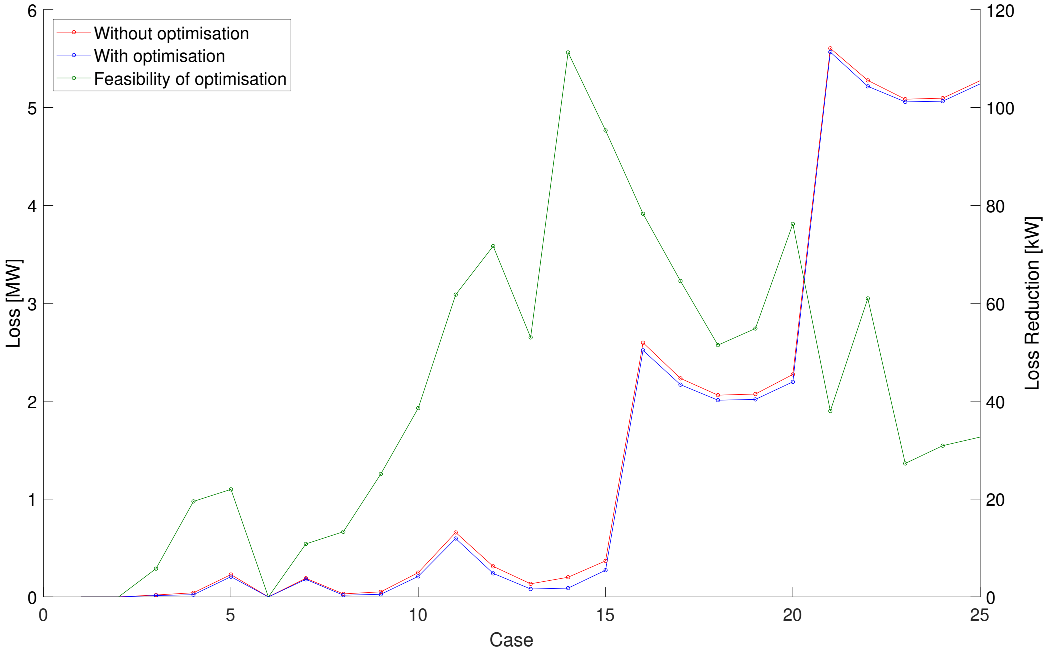


Figure 4.1: Active Power Loss Comparison: with and without Optimisation.

4.2.2 Choice of Control Variables

Fig. 4.2 shows the improvements of loss reductions for different sets of control variables. The improvements are calculated with respect to the loss reductions presented in Fig. 4.1. Thus, Fig. 4.2 represents the improvements when different control devices are added into the set of control variables. It can be seen that including transformer taps into the set of control variables improves the loss reduction significantly whereas only adding the reactor has no significant effect. However, the biggest improvement in loss reduction is achieved when both transformer taps and reactors are included in the set of control variables. This improvement can be explained by the fact that adding more control variables increases the search space of the algorithm and thereby the possibility of a better solution. Thus, the biggest active power loss reduction is achieved when the set of optimal setpoints consists of the Q's of the WTG strings, transformer tap positions and reactor status. Fig. 4.3 shows a loss comparison when the transformer taps and reactor status are included as well into the set of control variables.

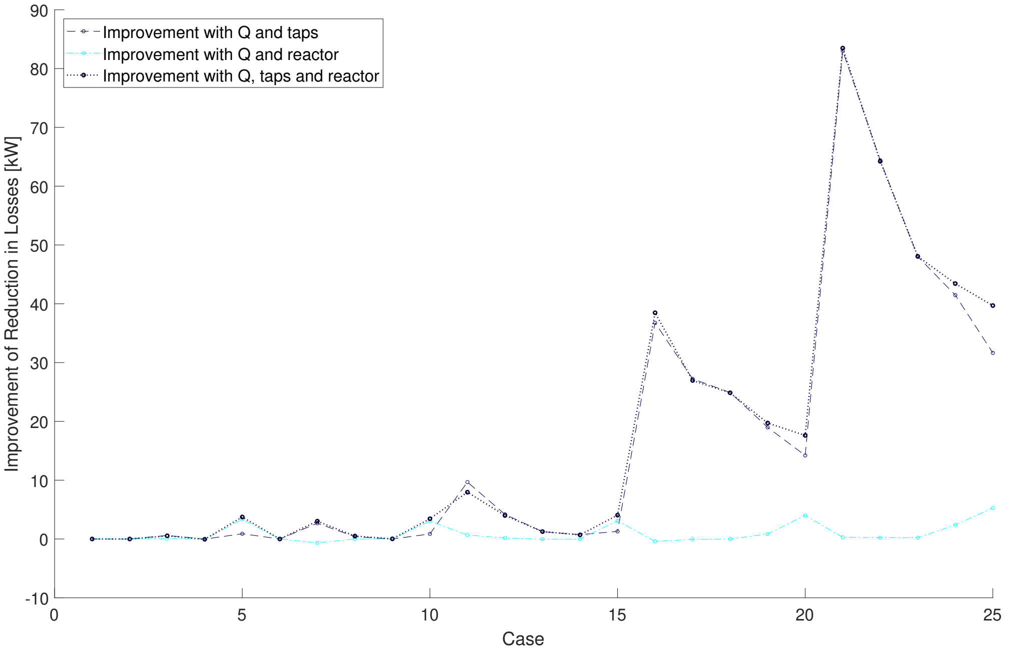


Figure 4.2: Loss Reduction Improvements with Different Sets of Control Variables.

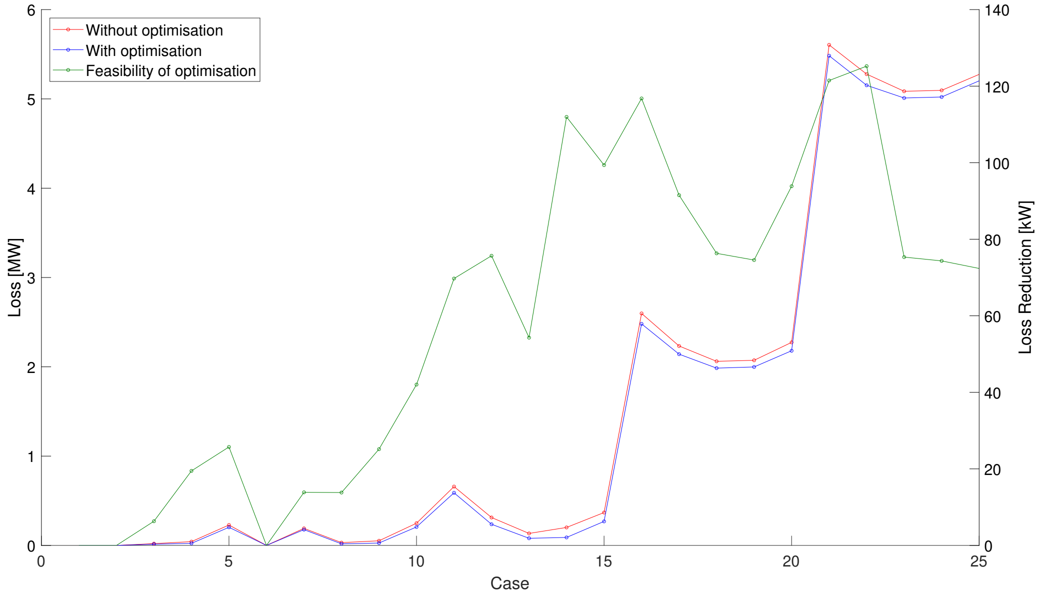


Figure 4.3: Active Power Loss Comparison: Loss Reduction Due to Most Feasible Set of Control Variables

4.3 Multiple Objective Optimisation

In this section, the optimisation is extended to all the objectives (to the full form of Eq. 3.7). All the objectives are considered since it is most feasible to regulate all controllable devices. The goal is to determine the weights of the different objectives. The main objective of the *optimisation unit* is to minimise the active power losses. To determine the effect of the different objectives, the weights are swept and it is looked at their effect on the cost from Eq. 3.13. Running the first parameters sweeps revealed something interesting: As soon as weight w_2 was non-zero, there were hardly any transformer switches and increasing the value

w_2 hardly had any effect on the amount of transformer tap switches. This is unexpected since increasing the weight should result in more emphasis lying on that objective. After sweeping w_2 for very small values, the following was found: Even though the weights suggested that the second objective was considered to be less important than the first, during optimisation the algorithm considered it to be the other way around. The problem lies in using the weighted-sum model in Eq. 3.7. This approach works well if the objectives are within the same range. This, however, is not the case. The reactor has a value in 10^0, the transformer taps in 10^{-2}, the losses in 10^{-5} and the distance of reactive power setpoints are in the range of 10^1 to 10^2 . Therefore, the actual order of importance is different from the intended order of importance. To solve this, the objectives must be transformed into the same unit. In this case, it would be costs in euro:

$$Cost(\mathbf{x}_t) = c_1 \cdot t_{passed} \cdot P_{loss}(\mathbf{x}_t) + c_2 \cdot taps(\mathbf{x}_t) + c_3 \cdot reactor(\mathbf{x}_t) + c_4 \cdot Q_{distance}(\mathbf{x}_t). \tag{4.2}$$

In Eq. 4.2, $c_1,\ c_2,\ c_3$ and c_4 are the costs associated to the different objectives, t_{passed} indicates how much time there is between two cases, $P_{loss}(\mathbf{x}_t)$ is the active power loss in MW, $taps(\mathbf{x}_t)$ and $reactor(\mathbf{x}_t)$ are the number of tap and reactor switches and $Q_{distance}(\mathbf{x}_t)$ is the average distance of the reactive power setpoints w.r.t. their boundary. To achieve this, Eqs. 3.8 and 3.9 are denormalised. In order to compare the different units, a cost is assigned to a change in each objective. The exact cost of energy is known. This is c_1 and is taken to be around 80 € per MWh which is just above the cost price for producing a MWh using wind turbines. It is assumed that an active power loss results in less power that can be sold. t_{passed} is needed to determine how much time has passed (in hours) and is used to determine how much MWh losses there is at time t. Since each case in the test profile describes a period of 15 minutes, t_{passed} should equal $\frac{1}{4}$ such that the total costs per case equal the costs induced after the 15-minute time span. The other objective costs are computed from c_1. To determine the other costs, it needs to be determined which changes are considered to yield an equal cost. For this, it is looked at the typical range of changes in the different objectives. To check the accuracy of these assumptions, an extensive reliability analysis of the system would be needed. Unfortunately, time and the available data do not allow this. Therefore, estimations are made which also illustrate the interplay between the different objectives based on simulations run so far.

First of all, one change in transformer taps is considered to be as costly as switching the reactor. Switching a couple of taps occurs often for different cases. Changing taps is considered to be worse than the switching of the reactor since it leads to maintenance costs whereas the other leads to temporary power flow disturbances. Also, the distance to the origin of the reactive power setpoints is of low importance in comparison with the other objectives. It is desired to have this value as close to the boundary as possible (in order to find solutions in challenging cases e.g. negative setpoints by the TSO) but not on the boundary itself. Therefore, an average distance of 70% of the boundaries is considered to be as costly as a transformer tap switch. Lastly, the difference in active power losses is in tens of kW (as shown in Fig. 4.2). Since the maximum improvement is sometimes just 10 kW, it is chosen to consider a small change in P_{loss} already significant. Therefore, a change of 5 kW in is considered to be as costly as 1 transformer tap change. This 5 kW corresponds to 1.25 kWh per 15 minutes which is also used to compute the price. Q$_{\text{distance}}$ has a value between 0 and 1 and indicates the average of the normalised distances to the origin of the different string setpoints. To compute the cost corresponding to Q$_{\text{distance}} = 1$, the price of 1.25 kWh is divided by 0.7. The costs of the different objectives are chosen such that they reflect this behaviour. The cost function of Eq. 4.2 becomes the OF which needs to be minimised by the algorithm. The costs are summarised in Tab. 4.1.

Table 4.1: Cost of Different Changes of the Controllable Devices

Change	5 kW or 1.25 kWh	1 tap switch	1 reactor switch	100% of Q_{range}
Associated *c_i*	c_1	c_2	c_3	c_4
Price [€]	0.10	0.10	0.10	0.14

To determine the weights, w_3 and w_4 are kept fixed and have values of 0.1 and 0.15. Then, w_1 and w_2 are swept. The result can be seen in Fig. B.3. The costs are around 419 € for $w_1 < w_2$. Then as $w_1 \geq w_2$, the costs suddenly drop. An investigation into the cause resulted in the following: When $w_1 < w_2$, there are no transformer tap switches. This is expected since the weight of tap switches is larger than the weight of power losses. The sudden drop in cost is due to the shift of importance between minimising active power losses and minimising transformer tap switches. As soon as $w_1 > w_2$, the active power loss decreases and the transformer tap switches increases. The cost stabilizes around 415 €. The small fluctuations are due to the deviations in the algorithm as was concluded during tuning of the algorithm parameters. The reduction in active power losses results in a larger cost reduction than the cost of the increase in transformer tap switches. Therefore, the *optimisation unit* concludes that the cost is lower by changing taps. This holds for different values of w_3 and w_4. Thus, it is concluded that using these costs, optimising on transformer tap switches does not increase savings (it decreases them). Consequently, w_2 is set to 0. To determine the values of w_3 and w_4, the small fluctuations of Fig. B.3 must be prevented. These are mainly due to deviations in the solutions between different runs. Therefore, 5 runs are performed per case while sweeping $w_3 = w_4$ from 0 to 0.2 while keeping w_1 constant. The results can be seen in Tab. B.6. For $w_3, w_4 = 0$, there is a larger cost. This is expected since these two objectives are not minimised and each causes a cost. After $w_3, w_4 = 0.10$, the cost starts increasing. This has to do with the fact that a reduction in active power losses results in a larger saving. Thus, the optimum of w_3 and w_4 is considered to be 0.05. To verify these parameters, the complete test profile of Fig. 3.3 was run. This is done for a CFC only controlling the reactive power setpoints of the WTG strings and for a CFC which also controls the taps and reactor. The latter is done to simulate a CFC of which the taps and reactor are in a more feasible position as is the case in reality. With no optimisation and only reactive power setpoints, the results with optimisation are always lower. For the second case, the costs with optimisation were larger in cases 3 and 7 of the test profile (see Fig. B.4). This was due to large changes in taps. The conclusions made before on minimising the tap switches were only for a small part of the test profile. That optimisation still yielded smaller costs in the end. To investigate whether there exists a combination of w_1 and w_2 which results in smaller losses for all cases, a final sweep is done on those weights with $w_1 > w_2$. The results can be seen in Figures 4.4 and B.5. These figures represent the cumulative costs for the test profile so the value after case 25 represents the total costs of the test profile.

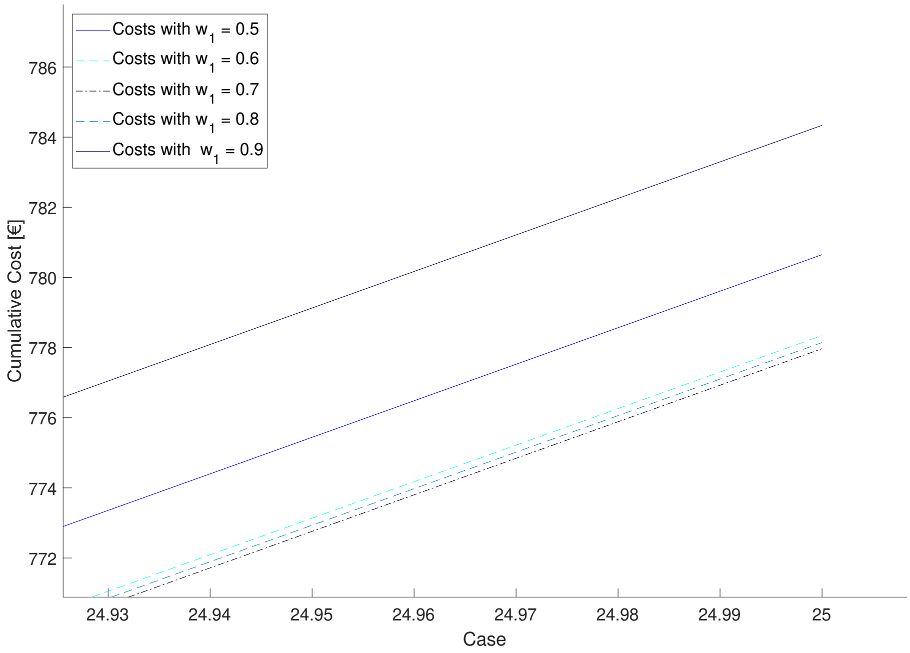


Figure 4.4: Final Cumulative Costs for the Test Profile of Subsection 3.4.3

Fig. 4.4 is zoomed in at the end of Fig. B.5. This is done in order to see the exact difference between the different weights. From Figure 4.4, it can be concluded that w_1 = 0.70 result in the smallest cumulative costs for the test profile and this value is chosen. The difference, however, is small and could also be caused by variations due to the algorithm. w_2 is computed by 1-w_1-w_3-w_4 and thus is 0.20. Please note that the costs without optimisation are higher and cannot be seen in this figure (but it is visible in Fig. B.5). In Fig. B.4, it is visible that the proposed final weights result in costs that are smaller than the costs without optimisation for case 3 and 7. This also holds for all cases as can be seen in Fig. B.6. The final weights are summarised in Tab. 4.2. With this step, the trade-off requirements of Chapter 2 are satisfied as much as possible. It is impossible to satisfy them all maximally since they have an influence on each other. Based on the costs of Tab. 4.1, the optimal combination is found i.e. the combination that yields the smallest costs.

Table 4.2: Value of the Final Weights

Weight w_i	w_1	w_2	w_3	w_4
Value	0.70	0.20	0.05	0.05

4.4 Flexibility and Robustness

This section discusses the flexibility and robustness of the *optimisation unit* when the 4 PVG strings are added into the system topology. Firstly, it is elaborated how the configuration of the *optimisation unit* changes when the extra strings are taken into account. Then, the feasibility after the addition of the extra strings is discussed. Finally, another algorithm parameter tuning for the final *optimisation unit* configuration is performed. This additional parameter tuning has the objective of additional improvement of the *optimisation unit's* performance and robustness.

When the PVG strings are added into the farm's topology, changes to the MATPOWER topology case file must be made. The locations of the extra strings were known beforehand and therefore the generator, branch and bus components describing the extra strings were already declared in the case file. However, the status of those generator, branch and bus components were initialised as 'disconnected'. This has the same effect as using a topology case file that describes the wind farm only. Now that the PVG strings must be taken into account, the status of the components describing the strings must be changed to 'connected' during the initialisation process of the *optimisation unit*. Furthermore, the number of optimisation variables of the optimisation problem changes from 16 (13 WTG strings, 2 transformers and 1 reactor) to 20 (13 WTG strings, 4 PVG strings, 2 transformers and 1 reactor). This increase in problem dimension requires specification of extra min/max boundaries of the new variables. These boundaries correspond to the PVG strings' reactive power capability; this depends on the momentary operating conditions, i.e solar irradiance. Therefore, solar irradiance must become an additional input of the *optimisation unit*.

Fig. 4.5 presents the feasibility of the *optimisation unit* when the PVG strings are added in addition to the set of control variables determined in Subsection 4.2.2. In this instance, 'No Optimisation' means that in addition to the WTG strings, the PVG strings are also controlled. This is based on the assumption that prevailing CFC's are also capable of controlling PVG strings since they require similar setpoints as WTG strings. Furthermore, 'Optimisation' means optimising for the optimal combination of weighted objectives determined in Section 4.3. Therefore, the feasibility is determined in terms of monetary costs (in €) rather than only the active power losses. The results presented in Fig. 4.5 state that the *optimisation unit* is able to find feasible solutions after the addition of the PVG strings. Thereby the *optimisation unit's* flexibility and robustness for varying system topologies are verified.

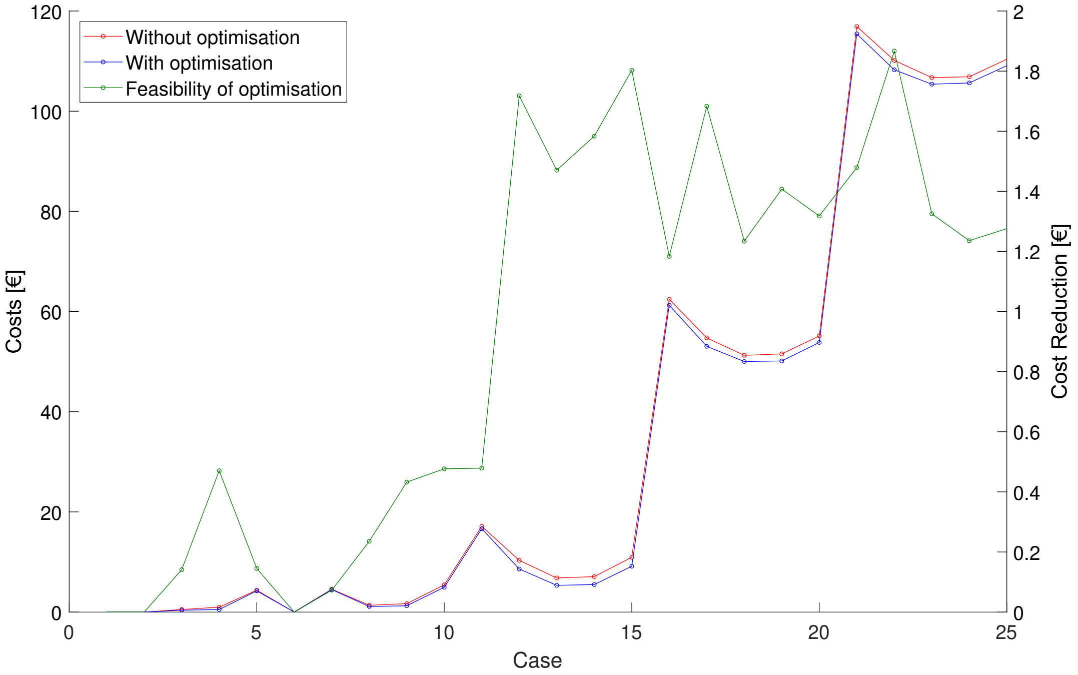


Figure 4.5: Cost Comparison: No Optimisation vs Optimisation

To establish how flexible and robust the *optimisation unit* is, another tuning of the parameters is done to determine whether choosing different parameters yields a significant improvement. The tuning is done with the extended topology using the approach of Section 4.1. The KPIs for the different parameters are presented in Appendix B.4. The following values for the algorithm parameters yielded the best KPIs and are chosen as parameters for the extended topology:

- $N_{particles}$: 50 particles
- $N_{archive}$: 2 solutions
- i_{max}: 25000 iterations
- m_{ini}: 0.80D and m_{final}: 0.25D where D is the total number of optimisation variables
- f_{s_ini}: 1 and f_{s_final}: 1

As can be seen, the optimal parameters have changed. This was expected since the dimension of the optimisation problem has also changed by adding PVG strings. Running the *optimisation unit* without tuned parameters still yields good solutions but tuning the parameters could ensure that smaller costs are obtained. In Fig. 4.6, the test profile is run for the tuned parameters. The results are compared to the performance of the *optimisation unit* without tuning of the parameters. The total costs are almost identical. When plotting the difference, it can be seen that in some cases tuning improves the costs and in others, it increases the costs. It is remarkable that for the case on which tuning is performed, the old parameters perform better than the new ones. Investigation revealed that the best fitness is indeed lower for the tuned parameters but the total costs are higher. This is due to the fact that some objectives (with different costs) are weighted higher in the fitness. This means that the relation between the fitness and the total costs is not linearly (as it was assumed) and therefore considering the fitness in the KPIs is not a good indication of the final objective (i.e. minimising costs). This means that no conclusion can be drawn from the tuning and therefore the old parameters are kept. If the parameters are chosen based on the total cost, it is possible that tuning improves the optimisation results for the changed topology provided that the improvement of tuning is significantly larger in comparison with the variation in solutions due to the non-deterministic nature of the algorithm.

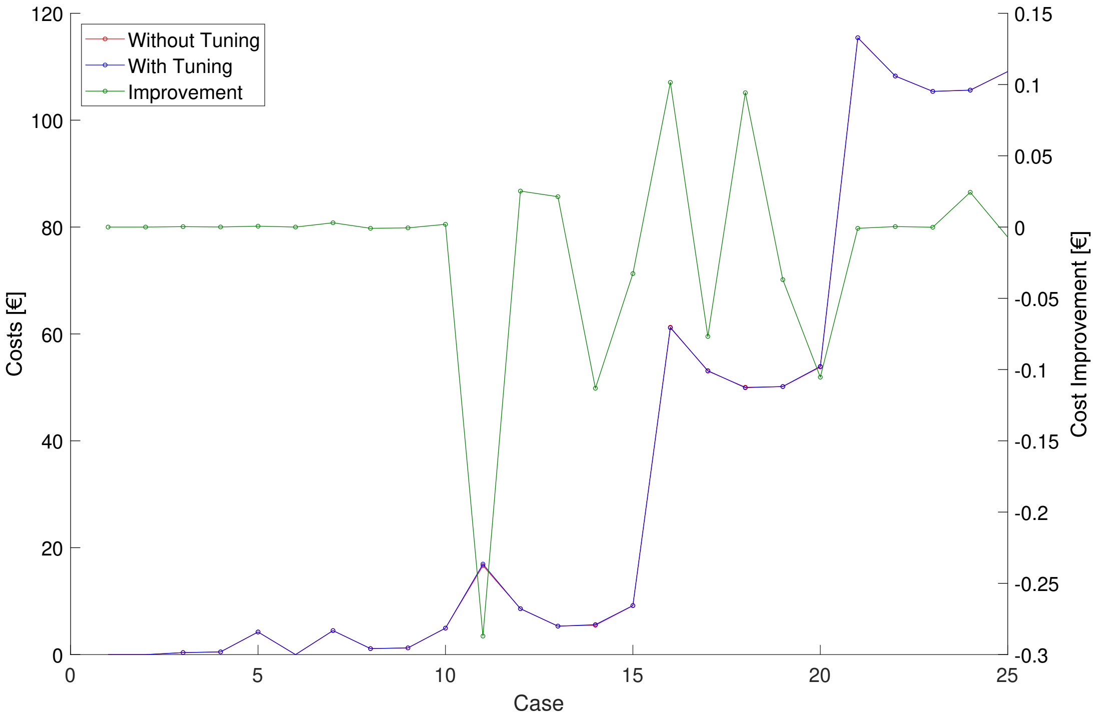


Figure 4.6: Cost Comparison: Before and After Algorithm Tuning

4.5 Prototype Integration

This section discusses two possible physical implementations of the *optimisation unit* within the designed CFC. Furthermore, a demonstration of the working principle of the *optimisation unit* is given based on the proposed implementation.

The proposed implementation involves calculating setpoints for the next time interval. This means that at $t = x$ minutes until $t = x + 15$ minutes, the *optimisation unit* calculates the optimal setpoints for the operating conditions at $t = x+15$ minutes using a 15-minute-ahead weather prediction. It is assumed that 15-minute-ahead weather predictions are fairly accurate in reality. Furthermore, it is assumed that the requested active power output usually equals the momentary available active power and that the requested reactive power equals zero [8]. When there is a request of active power curtailment, it is assumed that this information is known 15 minutes in advance. If the requested reactive power amount is nonzero as a result of reactive power imbalances in the grid, it is known that the possible requests take up a set discrete of values. The *optimisation unit* is then able to calculate setpoints for all possible reactive power requests using the 15-minute-ahead predictions. Therefore, this way of implementing the *optimisation unit* is expected to be feasible in reality. The other possible implementation is to perform real-time optimisation i.e calculate optimal setpoints at $t = x$ minutes using the measured operating conditions at $t = x$ minutes. However, this implementation is deemed to be less feasible due to the execution times of the *optimisation unit*: It takes around 5 - 10 minutes to complete 5 runs leaving only a few minutes for the actual implementation of the optimal setpoints.

For the demonstration of the proposed CFC, all the subgroups make use of the same test profile. This self-devised profile is one hour long and it consists of different operating conditions. The relevant profile information for the *optimisation unit* is presented in Tab. 4.3. It must be noted that the wind speeds and solar irradiances are not the actual values at the given times but they should be interpreted as predictions. The actual wind speed and solar irradiance values corresponding to the test profile are particularly relevant for the *control unit* and can be found in [19]. To clarify, the optimal setpoints corresponding to the first case are calculated at T = 0 min, the setpoints corresponding to the second case at T = 15 min and so forth. The calculated setpoints are presented in Tab. 4.4. The corresponding strings can be read from Fig. A.5.

These optimal setpoints are then passed on to the *model unit* for power flow analysis and to the *control unit* for enforcement of these setpoints and real-time control of the system. Those results are discussed in [10] and [19] respectively.

Table 4.3: Test Profile for the Prototype

Case	**Wind Speed [m/s]**	**Solar Irradiance [W/m^2]**	**$P_{requested}$[% of $P_{available}$]**	**$Q_{setpoint}$ [MVAr]**
1) T = 15 min	7.5	600	80	-50
2) T = 30 min	8.5	800	90	0
3) T = 45 min	10	450	100	50
4) T = 60 min	10	450	100	-50

Table 4.4: Optimal Setpoints for the Test Profile of the Prototype

Case [min]	**Q_{WTG1} [MVAr]**	**Q_{WTG2} [MVAr]**	**Q_{WTG3} [MVAr]**	**Q_{WTG4} [MVAr]**	**Q_{WTG5} [MVAr]**	**Q_{WTG6} [MVAr]**	**Q_{WTG7} [MVAr]**	**Q_{WTG8} [MVAr]**	**Q_{WTG9} [MVAr]**	**Q_{WTG10} [MVAr]**
T = 15	-14.049	-9.079	-14.739	-5.230	-5.814	-8.252	-1.912	-4.488	-6.758	-5.504
T = 30	-6.493	-3.734	-5.479	-3.333	-2.775	-3.272	-2.535	-2.769	-3.476	-3.575
T = 45	0.147	0.127	0.680	-0.662	-1.225	-1.376	-1.149	-0.854	0.101	-0.691
T = 60	0.092	-0.473	1.727	1.693	-1.154	-1.180	-1.309	-0.680	0.410	-0.043

Case [min]	**Q_{WTG11} [MVAr]**	**Q_{WTG12} [MVAr]**	**Q_{WTG13} [MVAr]**	**Q_{PVG1} [MVAr]**	**Q_{PVG2} [MVAr]**	**Q_{PVG3} [MVAr]**	**Q_{PVG4} [MVAr]**	**$tapT_1$ [position]**	**$tapT_2$ [position]**	**R_1 [status]**
T = 15	-4.274	-4.146	-9.449	2.023	1.823	1.945	0.604	9	9	ON
T = 30	-2.750	-2.835	-3.043	1.026	0.663	1.750	1.534	9	9	ON
T = 45	-1.545	-1.317	-0.142	1.918	0.984	2.425	3.022	9	9	ON
T = 60	-1.063	-1.282	-0.836	1.477	1.581	1.721	1.188	9	9	ON

Chapter 5

Concluding Remarks and Recommendations

This project concerns the design of the *optimisation unit* for a state-of-the-art central farm controller. The working principle of the *optimisation unit* is verified using the farm topology presented in Fig. A.5 which is based on a case study. The designed *optimisation unit* should be able to provide the *control unit* with a set of optimal setpoints for various operating conditions and within a time limit of 15 minutes. The calculated setpoints must be within the physical boundaries of the controllable devices and they must satisfy the TSO requests (Q_{ref}), the grid code (Figures A.1 and A.2) and all the technical constraints (Eqs. 3.6a-3.6d). Moreover, the calculated setpoints must take the following into account:

- Minimal active power losses
- Minimal long-term maintenance costs
- Minimal system disturbances
- Maximal system robustness/reliability

As presented in Chapter 4, the design of such a unit was successful. The optimisation results in cost reduction which is beneficiary in the long term; this is achieved while complying to all the mandatory requirements of Chapter 2. To achieve this, it is desired to control all the controllable devices in the farm. Moreover, the trade-off requirements are also satisfied as much as possible and it is demonstrated that this *optimisation unit* is able to perform under changing topologies. Also, it was found that based on the results obtained from the tuning of the algorithm parameters, no conclusion could be drawn about an improvement in performance. Lastly, the prototype is presented. It is also shown that this prototype works as desired and fulfils the requirements drafted in Chapter 2.

Recommendations for Further Research:
The final design of the *optimisation unit* was satisfactory. Looking back at the design process, some points of improvement are found. Moreover, some further research is needed in order to make such a unit ready for real-time implementation in a CFC. In order to have an even better *optimisation unit*, these points are presented below as recommendations for future research:

First of all, the parameter tuning is discussed. Ideally, this tuning is done after changing the optimisation problem. In this thesis, this was omitted from some steps in order to save computational time. If parameter tuning is done after each change of the optimisation problem, the best operation is ensured. Moreover, this tuning is ideally done in a controlled measurement environment. With this, it is meant that a computer with a clean install of Windows is used, which is also the device on which the *optimisation unit* is implemented. For logistic reasons, this was not possible for this group. Doing parameter tuning in this way allows for valid comparisons between run times and gives a better comparison between the results. Another thing needs to be noted on tuning: The costs of the objectives used in this optimisation problem (Tab. 4.1) are relatively low and thus different parameters result in a small change in total costs which has no meaning in real life. If the costs become larger, a small change in one of the algorithm parameters can result in large savings. Then, using different parameters will have a significant effect on the final savings. It is therefore recommended to check what the influence of tuning is when changing the optimisation problem. Moreover, it needs to be determined whether the change is caused by the change of the algorithm parameter or due to the variability in solutions.

Secondly, more accurate costs can be used for the multiple objective optimisation. In this thesis, this was based on the real costs of wind energy and equivalent importance was determined for the different objectives. In reality, it is possible that e.g. the transformers hardly ever fail. For this, a probabilistic study can be carried out to determine what the chance is for a transformer to fail. Together with the repair cost it can be used to compute a more realistic cost per tap switch. The same holds for the other objectives as well. In the end, the different costs can result in a different distribution of the weights for the objectives. This results in an optimisation closer to reality.

Thirdly, a more accurate model can be used. The model used is based on a real-life case study but has some simplifications. This is partly due to the amount of data available. In the end, a more accurate model results inherently in a better optimisation which is closer to reality.

Moreover, the functionality of the *optimisation unit* can be extended. For instance, it is possible to consider past data to generate better setpoints more quickly. Also, it is possible to implement a different decision system for the final solution. Currently, this is done based on the best costs. It is also possible to choose solutions based on different criteria such as setpoint distances w.r.t. past setpoints. Moreover, research can be carried out to investigate the possibility to compute the optimal active power setpoints as well. This would result in optimal setpoints for both active and reactive power which increases the efficiency of the CFC.

Lastly, the possible implementation of the *optimisation unit* in a real CFC is discussed. Essentially, there are two methods in which this is possible. The first one is predictive optimisation and the second one is real-time optimisation. In this thesis, predictive optimisation is used. This method relies on a prediction of the operating conditions of the given time intervals. The major advantage of this method is that there is more time available to find optimal setpoints. This allows for longer and more optimisation runs which decrease the variability due to the meta-heuristic nature of the algorithm. There is one problem with this approach: It is not known what the TSO request is beforehand and sudden setpoints requests are unaccounted for. The first issue can be solved by running multiple optimisations in parallel for the possible setpoint requests of the TSO. When the time slot in question arrives, the final setpoint is known and the corresponding solution is chosen. The latter problem can be solved by switching to real-time optimisation. In this type of optimisation, speed and small variability are crucial. The idea here is to run a quick optimisation run. Control is firstly done by the *control unit* since it is needed to have grid compliance quickly after the request. When the optimal setpoints are known, the *control unit* will move towards the optimal setpoints thus ensuring more efficient operation of the farm. This real-time optimisation can be investigated in more detail and algorithm parameters should be chosen such that runtimes and variability in solutions are as small as possible. Also, this mode of operation should be tested extensively in order to ensure correct operation under different conditions.

The proposed optimisation unit can be implemented with non-expensive computer hardware. A fast, multiple-core processor with a large amount of memory is recommended in order to speed up MATLAB simulations. The other components are fairly basic. An estimated price would probably lie in the 1000-2000 € range. This would yield optimisation run times under a minute (probably around 30 seconds) which is therefore also suited for real-time optimisation.

Acknowledgements:
We would like to thank our supervisor Jose Rueda Torres for his efforts in guiding us and giving us advice during the realisation of this thesis. We would also like to thank him for providing us with exemplary materials and his efforts in organising everything for us.

We would like to thank Vinay Sewdien, Babak Gholizad, Jianning Dong and Ugur Cicek for their participation in the green light assessment and their useful feedback.

[redacted]

We would like to thank I.E. Lager for his great organisation of the graduation project and for helping us during various stages of this project.

We would also like to thank J. Spaans, V. Scholten, P. López Cantero and T. Coggins who created and guided us through the Ethics and Business Plan part of this course.

Last but not least, we would like to thank our group members and friends Dennis, Farley, Laurens and Marouane for their support and understanding during this project. We enjoyed working with them very much and we will always look back to this period with a smile.

But most importantly, we would like to express our deepest appreciation and gratitude to our parents for their unconditional support since the moment we were born.

Bibliography

[1] M. S. S. et al., "Solution to optimal reactive power dispatch in transmission system using meta-heuristic techniques—status and technological review," *Electric Power Systems Research*, vol. 178, January 2020.

[2] J. R. Torres, A. Theologi, M. Ndreko, I. Erlich, and P. Palensky, "Metaheuristic approach for online optimal reactive power management in near-shore wind power plants," *2017 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America)*, 2017.

[3] A.-M. Theologi, "Metaheuristics-based optimal management of reactive power sources in offshore wind farms," Master's thesis, Delft University of Technology, 2016.

[4] P. Biswas, G. Amaratunga, and P. N. Suganthan, "Optimal power flow solutions incorporating stochastic wind and solar power," *Energy Conversion and Management*, vol. 147, 2017. DOI: 10.1016/j.enconman.2017.06.071.

[5] T. Zhou and W. Sun, "Optimization of wind-pv hybrid power system based on interactive multi-objective optimization algorithm," *Proceedings of 2012 International Conference on Measurement, Information and Control*, 2012. DOI: 10.1109/MIC.2012.6273421.

[6] Z. Ullah, S. Wang, J. Radosavjevic, and J. Lai, "A solution to the optimal power flow problem considering wt and pv generation," *IEEE Access*, vol. 7, 2019. DOI: 10.1109/ACCESS.2019.2909561.

[7] A. Naeem, N. U. Hassan, and C. Yuen, "Power loss minimization in power distribution systems using wind and solar complementarity," *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, vol. 5, 2018. DOI: 10.1109/ISGT-Asia.2018.8467939.

[8] [redacted]

[9] E. Triantaphyllou, *Multi-Criteria Decision Making Methods: A Comparative Study*. Dordrecht: SPRINGER-SCIENCE + BUSINESS MEDIA B.V., 2000.

[10] M. Mastouri and F. Rimon, "Power flow control in a substation of a wind- and solar farm," bsc graduation thesis, Delft University of Technology, July 2020.

[11] R. Zimmerman and M.-S. C.E., *Matpower User's Manual*. Power Systems Engineering Research Center (PS$_{\text{ERC}}$), 7 ed., June 2019.

[12] S. T. (convener of working group A2.37) et al., "Transformer reliability survey," tech. rep., CIGRE, 2015.

[13] J. Lundquist and A. Clifton, "How turbulence can impact power performance," *North American Windpower*, September 2012.

[14] J. Momoh, R. Adapa, and M. El-Hawary, "A review of selected optimal power flow literature to 1993. i. nonlinear and quadratic programming approaches," *IEEE Transactions on Power Systems*, vol. 14, 1999. DOI: 10.1109/59.744492.

[15] S. Behera, S. Sahoo, and B. Pati, "A review on optimization algorithms and application to wind energy integration to grid," *Renewable and Sustainable Energy Reviews*, vol. 48, 2015.

[16] J. L. Rueda and I. Erlich, "Hybrid single parent-offspring mvmo for solving cec2018 computationally expensive problems," *2018 IEEE Congress on Evolutionary Computation (CEC)*, 2018. 10.1109/CEC.2018.8477807.

[17] I. Erlich, "Mean variance mapping optimization."

[18] F. L. Albuquerque, A. Moraes, G. Caixeta, S. M. R. Sanhueza, and A. R. Vaz, "Photovoltaic solar system connected to the electric power grid operating as active power generator and reactive power compensator," *Solar Energy*, vol. 84, July 2010. 10.1016/j.solener.2010.04.011.

[19] D. Groenenberg and L. Beijnen, “Active- and reactive power control in a wind- and solar park,” bsc graduation thesis, Delft University of Technology, July 2020.

[20] [redacted]

Appendix A

Code, Tables & Figures

This appendix contains a reference to the source code. Moreover, this appendix has supplementary tables and figures. These tables and figures help with the clarification of certain statements/conclusions but are not considered to be essential enough to be placed in the main body of this thesis. These tables and figures are related to the design of the *optimisation unit*. The tables and figures related to the results can be found in Appendix B.

A.1 Source Code

The source code is published on GitHub at the following link:
`www.github.com/acneaguu/CFC-Optimisation`.

A.2 Grid Code

This section contains multiple figures which illustrate the required reactive power and voltage at the PCC.

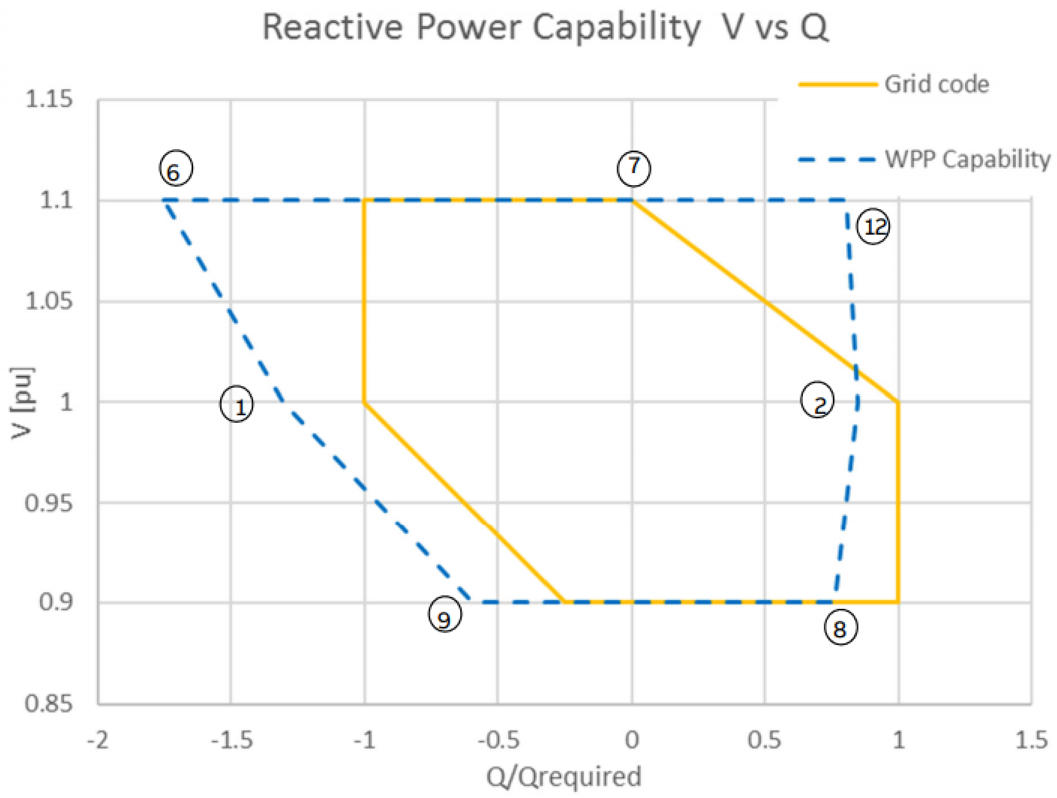


Figure A.1: Allowed Active and Reactive Power at the PCC (Figure 3-1 from [20])

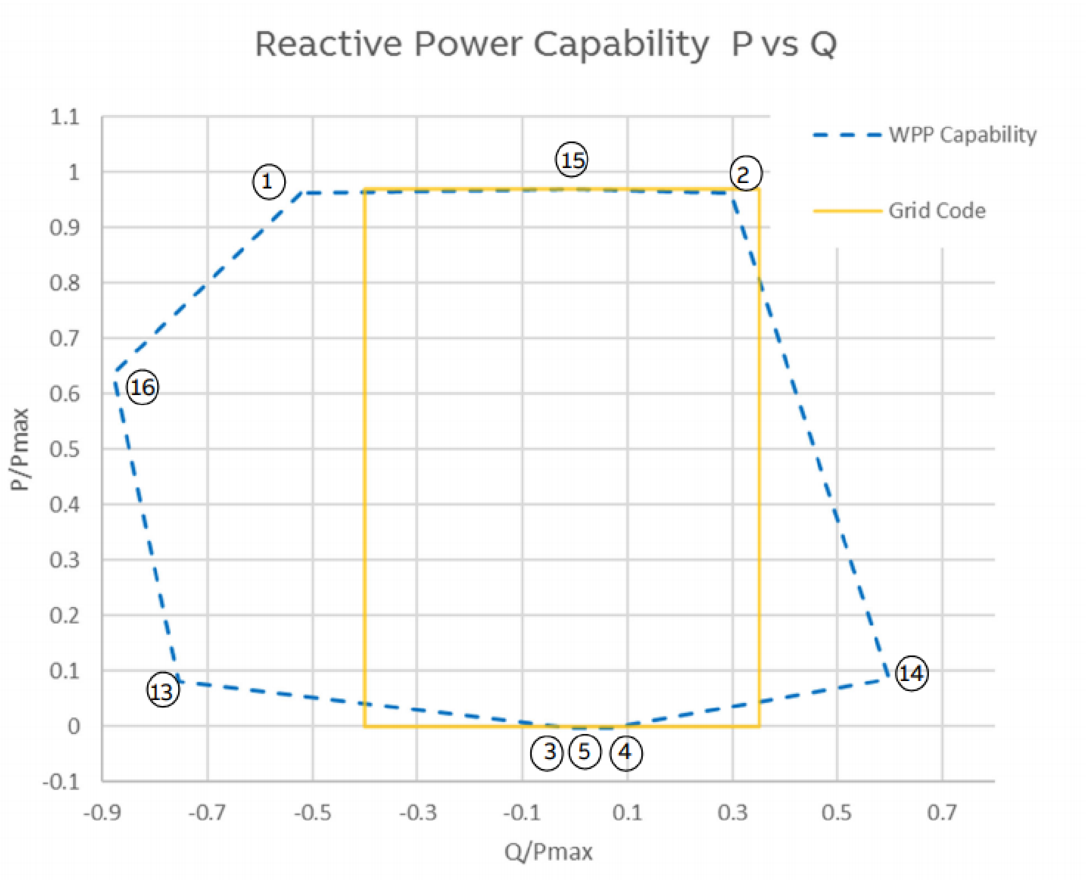


Figure A.2: Allowed Voltage Range as Function of the Reactive Power at the PCC (Figure 3-2 from [20])

A.3 Power Capabilities of the System

A.3.1 Active Power Capability

In Fig. A.3, the active power capability of the WF is presented. This capability is approximated using an aggregated model of all the WTG strings' active power capabilities .

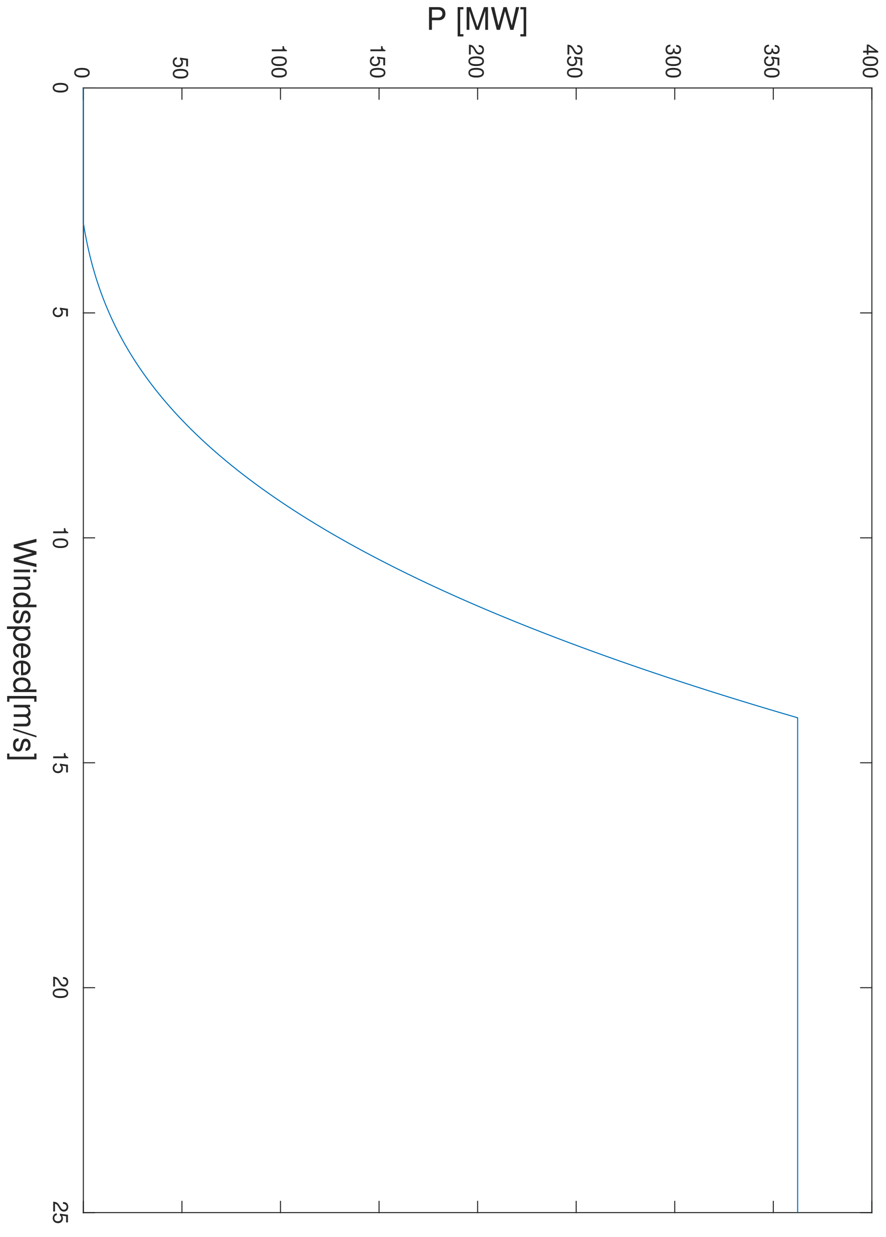


Figure A.3: Approximation of the Active Power Capability of the WF

A.3.2 Reactive Power Capability

In Fig. A.4, the reactive power capability of the WF and hybrid wind- and solar farm is presented. The blue lines indicate the smallest amount of reactive power the farm can produce and the red lines indicate the largest amount of reactive power. For the other lines, refer to the key. Please note that this figure is a rough approximation; reactive power losses are unaccounted for. This results in the actual injected reactive power by the branches having a lower effective value. The reactive power capability is therefore smaller than represented by the solid blue and red lines. This means that the reactive power capability is therefore between the dashed and solid blue and red lines. Nevertheless, this figure gives useful insight in what kind of setpoints can be reached, what kinds could be reached and which setpoints are impossible to satisfy at a given wind speed.

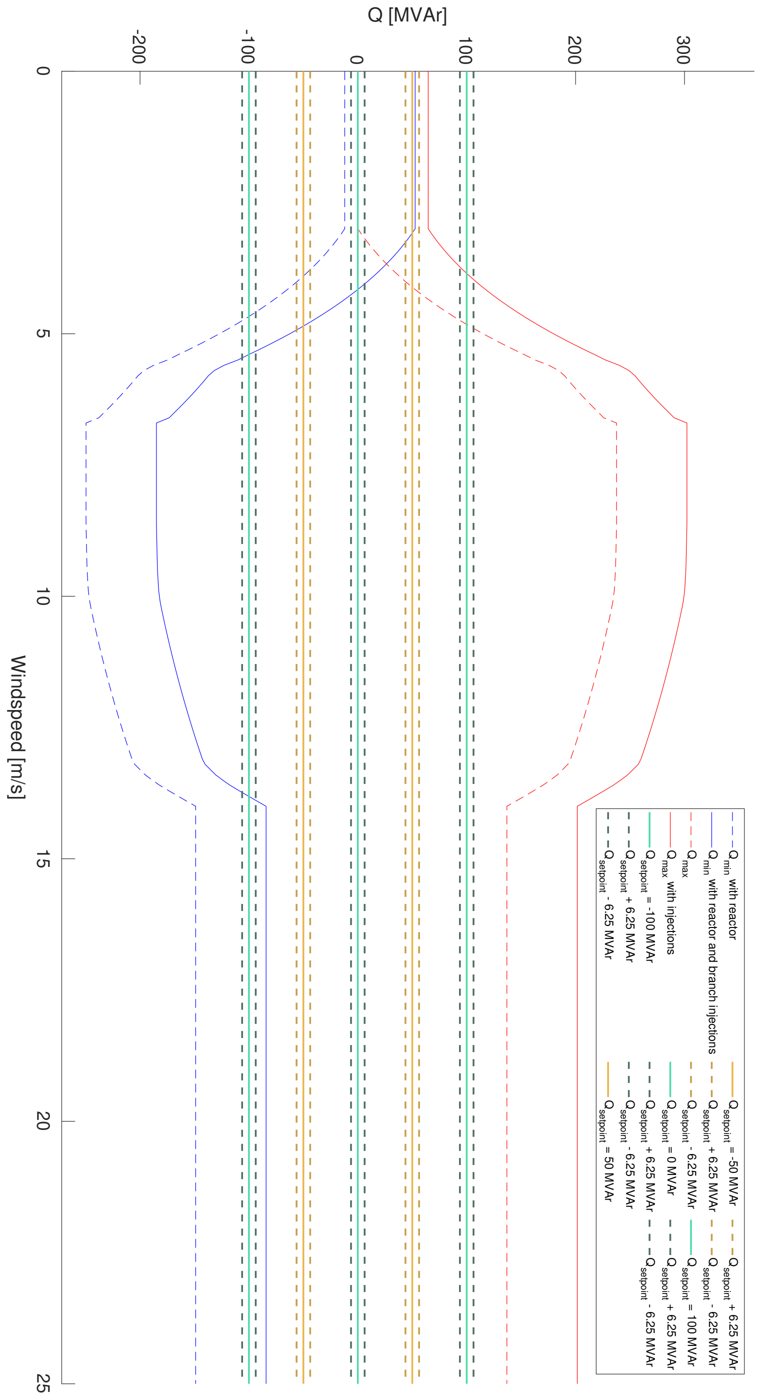
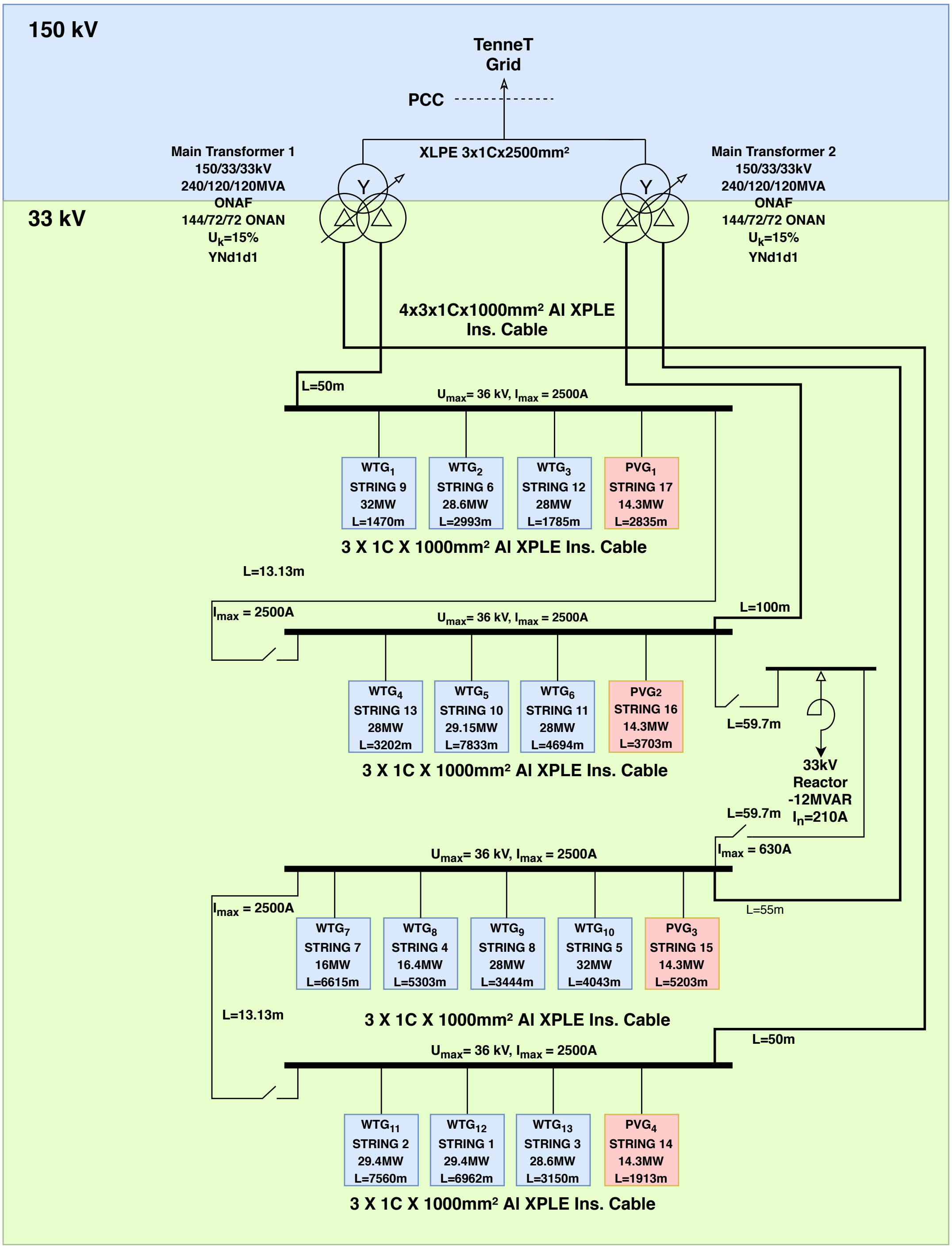


Figure A.4: Approximation of the Reactive Power Capability of the WF

A.4 The System in Question

This section presents the system topology used for the research in this paper. The system topology is seen in Fig. A.5.



150 kV
TenneT Grid
PCC
XLPE 3x1Cx2500mm²
Main Transformer 1
150/33/33kV
240/120/120MVA
ONAF
144/72/72 ONAN
U_k=15%
YNd1d1
Main Transformer 2
150/33/33kV
240/120/120MVA
ONAF
144/72/72 ONAN
U_k=15%
YNd1d1
Y
Y
33 kV
4x3x1Cx1000mm² Al XPLE Ins. Cable
L=50m
U_{max}= 36 kV, I_{max} = 2500A
WTG_1 STRING 9 32MW L=1470m
WTG_2 STRING 6 28.6MW L=2993m
WTG_3 STRING 12 28MW L=1785m
PVG_1 STRING 17 14.3MW L=2835m
3 X 1C X 1000mm² Al XPLE Ins. Cable
L=13.13m
I_{max} = 2500A
U_{max}= 36 kV, I_{max} = 2500A
L=100m
WTG_4 STRING 13 28MW L=3202m
WTG_5 STRING 10 29.15MW L=7833m
WTG_6 STRING 11 28MW L=4694m
PVG2 STRING 16 14.3MW L=3703m
3 X 1C X 1000mm² Al XPLE Ins. Cable
L=59.7m
33kV Reactor -12MVAR I_n=210A
L=59.7m
I_{max} = 630A
U_{max}= 36 kV, I_{max} = 2500A
I_{max} = 2500A
L=55m
WTG_7 STRING 7 16MW L=6615m
WTG_8 STRING 4 16.4MW L=5303m
WTG_9 STRING 8 28MW L=3444m
WTG_{10} STRING 5 32MW L=4043m
PVG_3 STRING 15 14.3MW L=5203m
L=13.13m
3 X 1C X 1000mm² Al XPLE Ins. Cable
L=50m
U_{max}= 36 kV, I_{max} = 2500A
WTG_{11} STRING 2 29.4MW L=7560m
WTG_{12} STRING 1 29.4MW L=6962m
WTG_{13} STRING 3 28.6MW L=3150m
PVG_4 STRING 14 14.3MW L=1913m
3 X 1C X 1000mm² Al XPLE Ins. Cable

Figure A.5: System Topology

A.5 Flowchart of the *Optimisation Unit*

In this section, the flowchart of the *optimisation unit* is presented. Please note that the algorithm is not elaborated in this flowchart. For the exact working of the algorithm, please refer to Fig. 3.1.

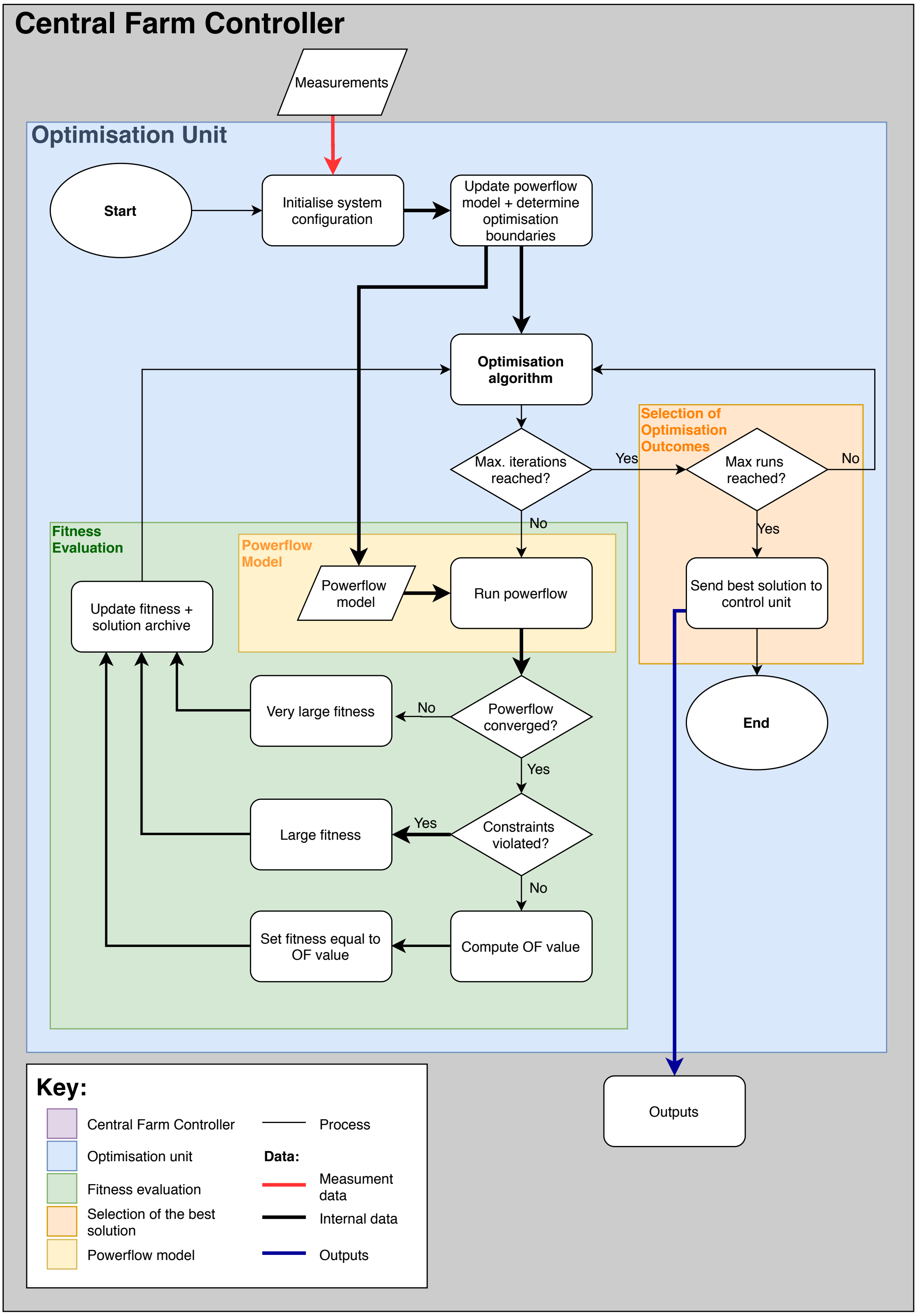


Figure A.6: Flowchart *Optimisation Unit*

A.6 Test Profile

This section presents the operating conditions corresponding to the cases of the test profile in Fig. 3.3.

Table A.1: Cases within the Test Profile.

Case	Wind Speed [m/s]	Solar Irradiance [W/m^2]	$P_{available}$ [p.u]	$Q_{setpoint}$ [p.u]
1	4.5	0	0.024434	-0.286
2	4.5	0	0.024434	-0.143
3	4.5	0	0.024434	0
4	4.5	0	0.024434	0.143
5	4.5	0	0.024434	0.286
6	5	340	0.092738	-0.286
7	5	340	0.092738	-0.143
8	5	340	0.092738	0
9	5	340	0.092738	0.143
10	5	340	0.092738	0.286
11	7	680	0.231201	-0.286
12	7	680	0.231201	-0.143
13	7	680	0.231201	0
14	7	680	0.231201	0.143
15	7	680	0.231201	0.286
16	12	510	0.731243	-0.286
17	12	510	0.731243	-0.143
18	12	510	0.731243	0
19	12	510	0.731243	0.143
20	12	510	0.731243	0.286
21	15	170	1.062984	-0.286
22	15	170	1.062984	-0.143
23	15	170	1.062984	0
24	15	170	1.062984	0.143
25	15	170	1.062984	0.286

Appendix B

Results

This appendix contains tables and figures which give more insight on the results obtained in Chapter 4. These tables and figures are omitted since they are either very extensive or not important enough to place in the main body.

B.1 Algorithm Parameter Tuning using only Reactive Power Setpoints of WTGs

Table B.1: KPI Results for Various Particle Population Sizes

Population Size	1	5	10	20	**35**
Best Fitness	1.6176E-03	1.5522E-03	1.5316E-03	1.5355E-03	1.5317E-03
Avg Fitness	1.9554E+18	1.5745E-03	1.5611E-03	1.5473E-03	1.5376E-03
Std Fitness	6.1835E+18	1.8141E-05	2.7258E-05	9.9401E-06	3.9681E-06
Avg Runtime [s]	2.50E+00	1.20E+01	2.28E+01	4.65E+01	8.64E+01
Times Converged	9	10	10	10	10
Avg Std Q's [MVAr]	7.9036	3.6335	3.1936	2.3555	1.5945

Population Size	40	45	50	100	200
Best Fitness	1.5280E-03	1.5313E-03	1.5293E-03	1.5301E-03	1.5268E-03
Avg Fitness	1.5362E-03	1.5371E-03	1.5349E-03	1.5314E-03	1.5295E-03
Std Fitness	5.7779E-06	4.3594E-06	3.8842E-06	1.2109E-06	1.8077E-06
Avg Runtime [s]	1.09E+02	1.15E+02	1.21E+02	2.84E+02	4.80E+02
Times Converged	10	10	10	10	10
Avg Std Q's [MVAr]	1.6955	1.5028	1.3834	1.0798	0.9645

Table B.2: KPI Results for Various Archive Sizes

Archive Size	2	**3**	4	5
Best Fitness	1.5523E-03	1.5528E-03	1.5557E-03	1.5543E-03
Avg Fitness	1.5593E-03	1.5569E-03	1.5602E-03	1.5598E-03
Std Fitness	8.1151E-06	3.4172E-06	4.1224E-06	3.0968E-06
Avg Runtime [s]	9.33E+01	8.95E+01	7.83E+01	8.05E+01
Times Converged	10	10	10	10
Avg Std Q's [MVAr]	1.7235	1.2636	1.6462	1.3434

Table B.3: KPI Results for Various Initial Mutation Sizes

Initial Mutations	11(0.85)	10(0.77)	9(0.69)	8(0.62)	**7 (0.54)**
Best Fitness	1.5541E-03	1.5539E-03	1.5540E-03	1.5522E-03	1.5538E-03
Avg Fitness	1.5609E-03	1.5598E-03	1.5628E-03	1.5596E-03	1.5590E-03
Std Fitness	6.3629E-06	3.6423E-06	7.9250E-06	5.3148E-06	2.8282E-06
Avg Runtime [s]	8.50E+01	8.09E+01	7.83E+01	7.95E+01	8.10E+01
Times Converged	10	10	10	10	10
Avg Std Q's [MVAr]	1.6419	1.4594	1.7897	1.5392	1.4994

Table B.4: KPI Results for Various Final Mutation Sizes

Final Mutations	6 (0.46)	5 (0.38)	**4 (0.31)**	3 (0.23)	2 (0.15)
Best Fitness	1.5523E-03	1.5574E-03	1.5561E-03	1.5552E-03	1.5593E-03
Avg Fitness	1.5594E-03	1.5597E-03	1.5582E-03	1.5590E-03	1.5617E-03
Std Fitness	6.0213E-06	2.4327E-06	1.8099E-06	3.0313E-06	2.6677E-06
Avg Runtime [s]	8.82E+01	9.07E+01	8.71E+01	8.65E+01	8.95E+01
Times Converged	10	10	10	10	10
Avg Std Q's [MVAr]	1.3016	1.4019	1.3773	1.659	1.7511

Table B.5: KPI Results for Various Final Scaling Parameters

Final scaling parameter	**1**	5	10	20
Best Fitness	1.5540E-03	1.5556E-03	1.5551E-03	1.5547E-03
Avg Fitness	1.5594E-03	1.5624E-03	1.5638E-03	1.5643E-03
Std Fitness	4.4933E-06	5.8378E-06	6.7457E-06	7.3463E-06
Avg Runtime [s]	5.51E+01	5.34E+01	6.39E+01	5.33E+01
Times Converged	10	10	10	10
Avg Std Q's [MVAr]	1.5453	1.8606	1.9491	1.9664

B.2 Feasibility of Optimisation

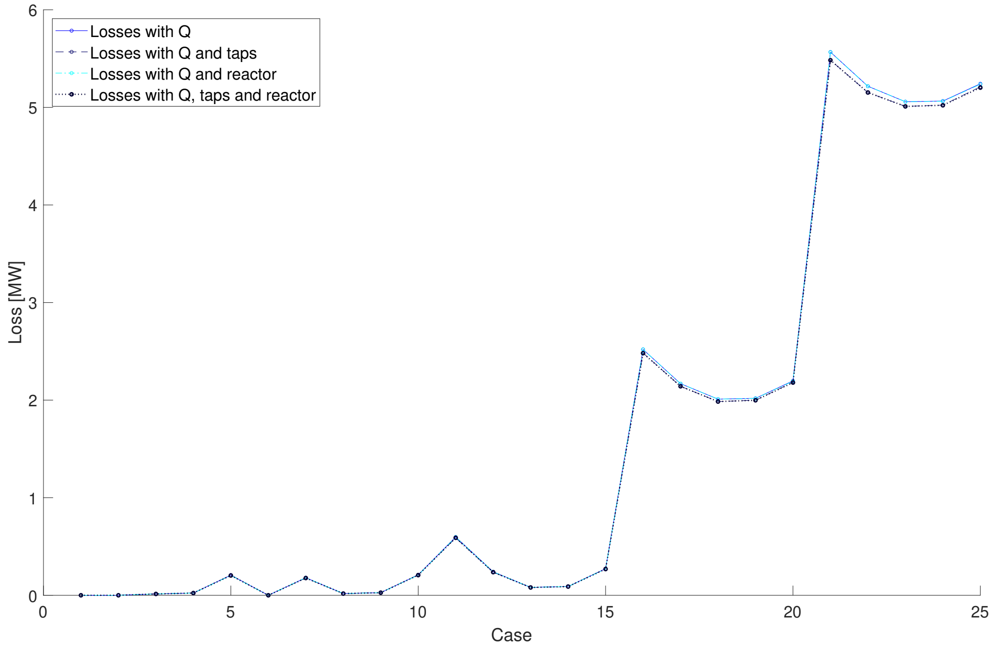


Figure B.1: Active Power Losses for Different Controllable Devices

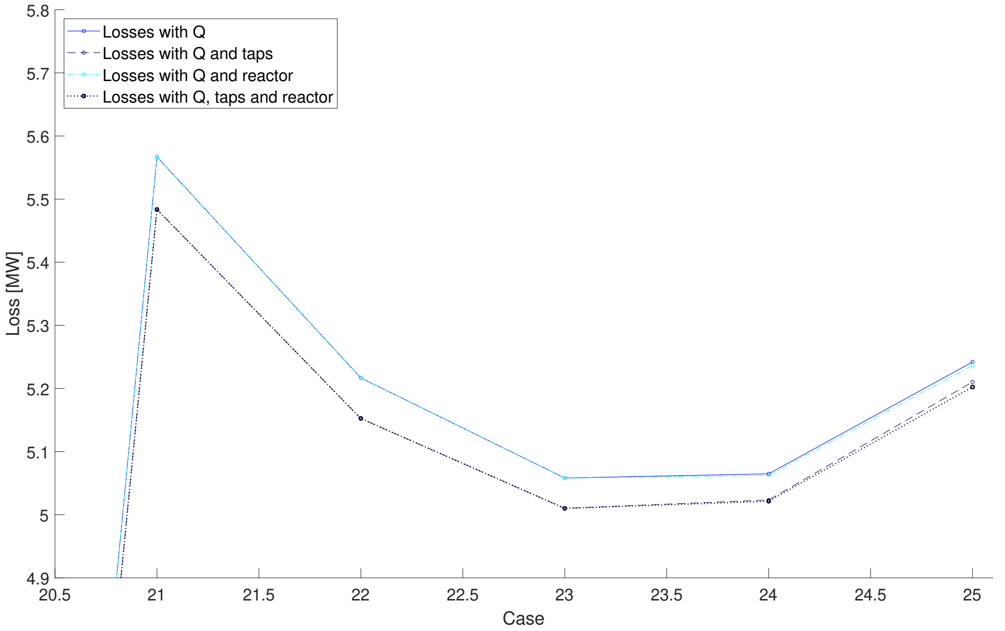


Figure B.2: Active Power Losses for Different Controllable Devices: *zoomed in around case 20-25*

B.3 Multiple Objective Optimisation

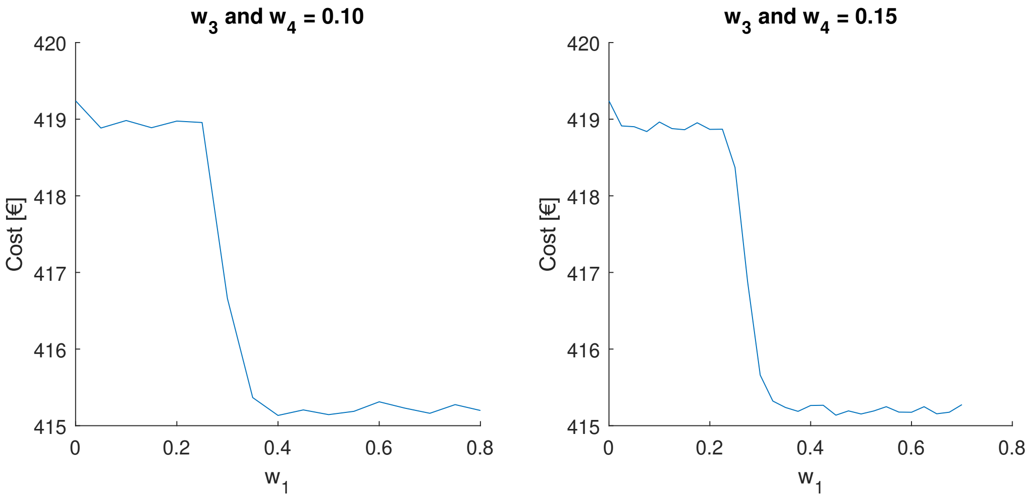


Figure B.3: Sweep of w_1 while Keeping w_3 and w_4 Fixed

Table B.6: Costs with Different Values of $w_3 = w_4$

Value of w_3 and w_4	0.00	0.05	0.10	0.15	0.20
Total Costs [€]	415.29	415.24	415.33	415.30	415.33

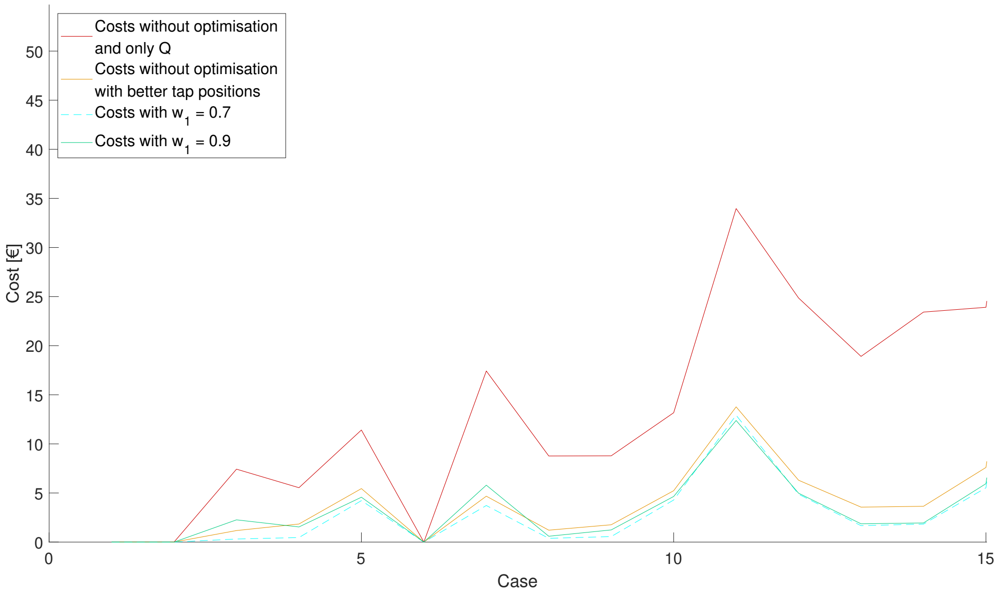


Figure B.4: Comparison of Costs for Different Cases of the Test Profile:
zoomed in around cases 1-15

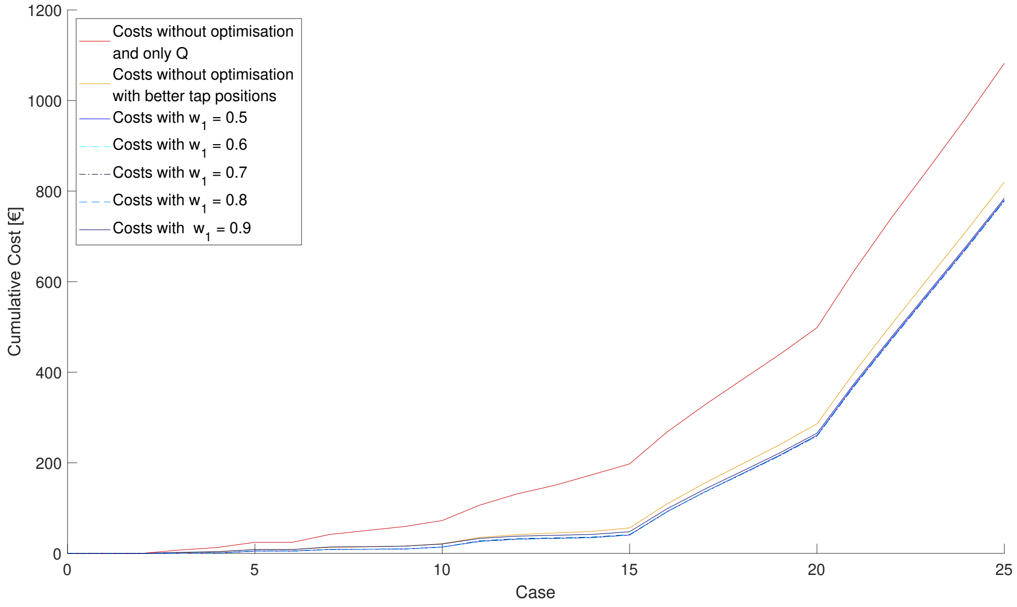


Figure B.5: Cumulative Costs for the Test Profile of Subsection 3.4.3

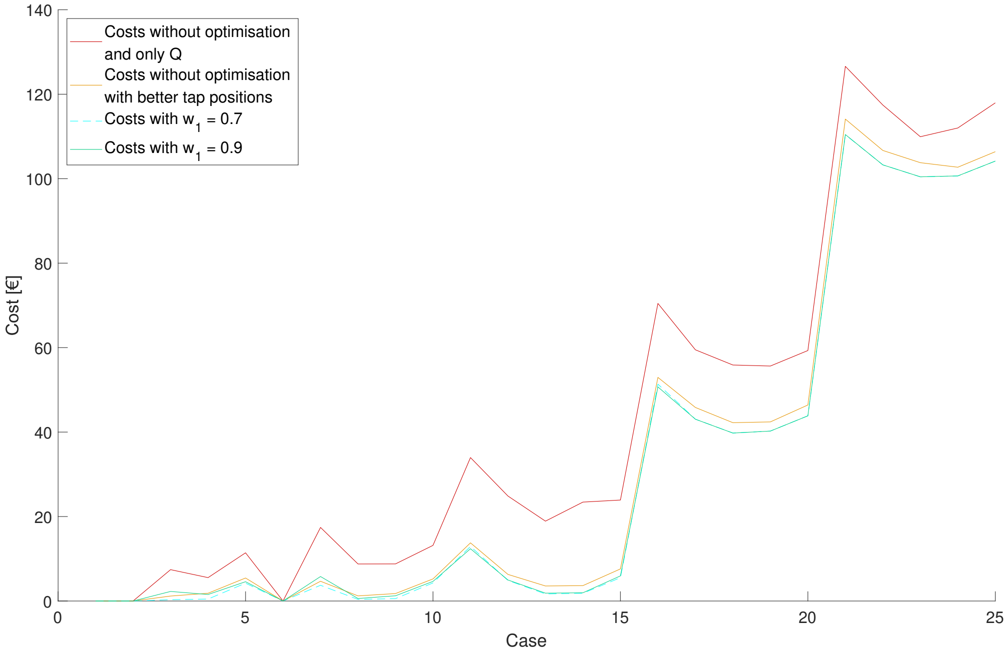


Figure B.6: Comparison of Costs for Different Cases of the Test Profile

B.4 Parameter Tuning for the Extended Topology

Table B.7: KPI Results for Various Particle Population Sizes

Population Size	1	5	10	20	35	**50**	65	75	85	100	200
Best Fitness	8.20586	7.95923	7.92667	7.88969	7.87790	7.84870	7.84357	7.83957	7.83528	7.83983	7.83420
Avg Fitness	8.64707	8.10474	7.99779	7.95799	7.89663	7.87294	7.86150	7.85833	7.85736	7.85317	7.84232
Std Fitness	0.2445	0.1223	0.0599	0.0434	0.0211	0.0166	0.0126	0.0113	0.0133	0.0081	0.0048
Avg Runtime [s]	1.466	6.285	11.837	23.433	42.020	61.995	83.869	95.087	105.681	129.883	472.141
Times Converged	10	10	10	10	10	10	10	10	10	10	10
Avg Std Q's [MVAr]	4.9838	2.9867	2.9000	2.2316	1.4209	1.1039	0.9441	0.9196	0.9161	0.8213	0.6907

Table B.8: KPI Results for Various Archive Sizes

Archive Size	**2**	3	4	5
Best Fitness	7.84283	7.85277	7.85544	7.86425
Avg Fitness	7.86538	7.87073	7.89011	7.88262
Std Fitness	0.0150	0.0109	0.0279	0.0100
Avg Runtime [s]	124.176	124.929	127.695	116.329
Times Converged	10	10	10	10
Avg Std Q's [MVAr]	1.0916	1.1750	1.2145	1.0487

Table B.9: KPI Results for Various Initial Mutation Sizes

Initial mutations	17 (0.85)	**16 (0.80)**	15 (0.75)	14 (0.70)	13 (0.65)	12 (0.60)
Best Fitness	7.85382	7.85392	7.85537	7.85360	7.85266	7.84445
Avg Fitness	7.87261	7.87706	7.86791	7.86876	7.87372	7.86926
Std Fitness	0.0127	0.0123	0.0148	0.0156	0.0167	0.0189
Avg Runtime [s]	104.670	92.332	86.089	94.712	119.904	119.803
Times Converged	10	10	10	10	10	10
Avg Std Q's [MVAr]	1.1570	1.0609	1.2334	1.1734	1.2013	1.1393

Table B.10: KPI Results for Various Final Mutation Sizes

Final mutation size	9 (0.45)	7 (0.35)	**5 (0.25)**	3 (0.15)	2 (0.10)	1 (0.05)
Best Fitness	7.84707	7.85235	7.83793	7.84315	7.86328	7.85897
Avg Fitness	7.87802	7.88018	7.87213	7.88516	7.88415	7.90535
Std Fitness	0.0202	0.0159	0.0249	0.0239	0.0186	0.0419
Avg runtime [s]	92.700	76.229	84.931	90.734	139.590	139.369
Times Converged	10	10	10	10	10	10
Avg Std Q's [MVAr]	1.2499	1.2793	1.2808	1.4062	1.5752	1.7279

Table B.11: KPI Results for Various Final Scaling Parameters

Final mutation size	**1**	5	10	20
Best Fitness	7.84046	7.85387	7.88128	7.88163
Avg Fitness	7.86554	7.87022	7.90569	7.92222
Std Fitness	0.0173	0.0155	0.0179	0.0367
Avg runtime [s]	120.864	121.257	120.106	123.535
Times Converged	10	10	10	10
Avg Std Q's [MVAr]	1.0961	1.1915	1.1841	1.5702