Erasmus University Rotterdam

Erasmus School of Economics

Master Thesis Operations Research & Quantitative Logistics

A Robust Approach to the Two-Stage AC Optimal Power Flow using Linear Approximations

Author: Alex van Gessel (454458)

Supervisor: Dr. (Olga) O. Kuryatnikova

Second Assessor: Dr. (Remy) R. Spliet

Abstract

Electric power systems are increasingly being exposed to uncertainty in power load and generation. It is important that these uncertain fluctuations are taken into account when an operating point is determined. We aim to develop a computationally efficient approach for finding a robust operating point for the Alternating Current Optimal Power Flow (ACOPF) problem. We consider the ACOPF problem as a two-stage problem with first-stage and second-stage variables that need to be decided upon before and after uncertainty is known, respectively. Our goal is to find a solution for the first-stage variables, such that a solution to the second stage exists when uncertainty is realized. Due to the nonlinear, non-convex power flow equations in the ACOPF problem, finding a robust solution is challenging. Therefore, we approximate the power flow equations by iteratively taking first-order Taylor Series approximations on small subsets of the solution space. As a result, we are able to eliminate the second-stage variables and the power flow equations from the problem and we obtain an approximately robust solution. We evaluate the performance of our solution on MATPOWER instances up to 1354 buses. We find that our solution increases the operating cost only marginally compared to the nominal solution. Moreover, by means of a simulation study we find that our solution performs well in terms of robust feasibility. Finally, our approach is promising in terms of computational efficiency and has potential to be scalable to even larger instances.

Keywords: AC optimal power flow, two-stage robust optimization, uncertainty in power systems.

November 7, 2022

Contents

1 Introduction **4**

2 Literature **8**

2.1 Adjustable robust optimization 8

2.2 The ACOPF problem 10

2.3 The ACOPF problem with uncertainty 12

3 Nominal ACOPF Model **13**

3.1 Components of the ACOPF problem 13

3.2 Reference bus, PV buses and PQ buses 15

3.3 Formulation of the nominal ACOPF model 16

4 Incorporating Uncertainty **20**

5 Methodology **24**

5.1 General idea of the approach 24

5.2 Solution procedure 26

5.3 Line flow expressions for PV and PQ buses 28

5.4 Line flow expressions for reference bus 28

5.5 Linear parts of line flow expressions for all buses 29

5.6 Linearizing the system of equalities 30

5.7 Eliminating the state variables 32

5.8 Searching in a small subset of the feasible space 33

5.9 Eliminating uncertainty 34

5.10 Dealing with the voltage at the reference bus 35

5.11 Stating the reformulated problem 36

5.12 Including the line thermal limits 37

5.13 Pseudocode of the algorithm 39

6 Data **40**

7 Results of computing robustly feasible solutions **41**

7.1 Results for all instances 41

7.2 Analysis of 'case6ww' and 'case30' 46

8 Performance of solutions under simulated uncertainty **47**

8.1 Simulation setup 47

8.2 Simulation results 49

8.3 Influence of excluding the line thermal constraints 55

9 Conclusion and Discussion **57**

9.1 Discussion of numerical experiments 58

9.2 Limitations and directions for further research 59

Appendix A Alternative modeling choices **65**

A.1 Infinity norm to construct the subsets of state variables 65

A.2 Alternative options for dealing with the voltage at the reference bus 65

Appendix B Robust counterpart of a linear constraint with conic uncertainty **66**

Appendix C Coefficients for line thermal limits **67**

Appendix D SDP reformulations of robust line thermal constraints **68**

Table 1: Nomenclature

Data	
N	Set of buses
G	Set of generator buses
U	Set of buses with uncertainty (in active power load)
L	Set of branches
L_r	Set of branches connected to the reference bus
$l = (k, m) \in L$	Branch with k as *from*-end and m as *to*-end
$p_k^d \in \mathbb{R}$	Active power load at bus k
$q_k^d \in \mathbb{R}$	Reactive power load at bus k
$g_k^{sh} \in \mathbb{R}$	Conductance of shunt element at bus k
$b_k^{sh} \in \mathbb{R}$	Susceptance of shunt element at bus k
$y_l \in \mathbb{C}$	Admittance of branch l
$g_l \in \mathbb{R}$	Conductance (real part of admittance) of branch l
$b_l \in \mathbb{R}$	Susceptance (imaginary part of admittance) of branch l
$b_l^{sh} \in \mathbb{R}$	Total shunt susceptance of branch l
$p_k^{\min}, p_k^{\max} \in \mathbb{R}$	Minimum/Maximum active power generation at bus k
$q_k^{\min}, q_k^{\max} \in \mathbb{R}$	Minimum/Maximum reactive power generation at bus k
$s_l^{\max} \in \mathbb{R}_+$	Maximum apparent power flow in branch l
$v_k^{\min}, v_k^{\max} \in \mathbb{R}_+$	Minimum/Maximum voltage magnitude at bus k
$c_{k,2}, c_{k,1}, c_{k,0} \in \mathbb{R}_+$	Coefficients of quadratic cost function of generated active power at bus k
Variables	
$p_k^g \in \mathbb{R}$	Active power generated by generator at bus k
$q_k^g \in \mathbb{R}$	Reactive power generated by generator at bus k
$v_k \in \mathbb{C}$	Voltage at bus k
$v_k^M \in \mathbb{R}_+$	Squared voltage magnitude at bus k
$v_k^r \in \mathbb{R}$	Real part of voltage at bus k
$v_k^j \in \mathbb{R}$	Imaginary part of voltage at bus k
$p_l^f \in \mathbb{R}$	Active power injected into branch $l = (k, m)$ at the *from*-end k
$q_l^f \in \mathbb{R}$	Reactive power injected into branch $l = (k, m)$ at the *from*-end k
$p_l^t \in \mathbb{R}$	Active power injected into branch $l = (k, m)$ at the *to*-end m
$q_l^t \in \mathbb{R}$	Reactive power injected into branch $l = (k, m)$ at the *to*-end m

1 Introduction

The Alternating Current Optimal Power Flow (ACOPF) problem is a well-known problem in the field of power systems optimization. It is a problem that needs to be solved multiple times per day, in order to provide corporations and households with a sufficient amount of electricity. The problem can be described as finding an operating point of a power system that minimizes a certain cost function, while satisfying a set of constraints that model the power flowing through the system, as well as voltage limits, generated power limits and thermal limits for transmission lines. The ACOPF problem is non-convex and strongly NP-hard (Lehmann et al. (2015), Bienstock and Verma (2019)), even without uncertainty. This non-convexity is a result of the quadratic system of equalities in the problem, also known as the *power flow equations*. As a result of this non-convexity and NP-hardness, exact solution methods for the ACOPF problem are computationally intractable and researchers rely on heuristics and approximations to solve the problem. On top of that, the real-life application of the ACOPF problem brings along a certain degree of uncertainty. For example, with the rise of renewable energy, uncertainty in power supply has increased throughout the years. Dealing with uncertainty in power demand is also of increasing importance, since more extreme weather conditions cause large fluctuations in peak demands. This uncertainty in power supply and demand can cause mismatches in the power system, and can even result in complete power outages (Kundur et al. (2004)). Such power outages have increased throughout the years, as Wirfs-Brock (2016) shows for the United States. As a result, solution techniques that deal with uncertainty in power systems have increased in popularity in recent years.

For instance, Molzahn and Roald (2018) propose to implement convex relaxations for the non-linear power flow equations. While their method guarantees robust feasibility of the operational constraints (that is, the inequality constraints of the problem that represent physical limits of the power network), their method lacks scalability to large instances. Lee et al. (2021) propose a method that uses convex inner approximations of the feasible space, which they call convex restrictions. This approach seems promising, considering the relatively low computation time needed to solve medium-sized instances. However, only low levels of uncertainty are considered for these instances. Furthermore, they mention that their convex restrictions can still be conservative, and their approach is still in development. Kuryatnikova et al. (2021) tackle the non-convexity of the problem by approximating the nonlinear equality constraints on small subsets of the feasible space, using

first-order Taylor Series approximations. Their results are promising in terms of robust feasibility. However, the ACOPF formulation they use results in nonlinearities in the inequality constraints of the approximated problem, limiting the scalability of their approach. Finally, Roald and Andersson (2017) propose a computationally tractable chance-constraints based method; they obtain a solution for a large Polish power network in approximately 30 seconds. However, their method requires information on the distribution of the uncertain data, which is often unavailable or incomplete. Also, their approach provides no guarantees for converging to a (local) optimum. To summarize, while some methods for the ACOPF problem with uncertainty yield promising results, research in the field is relatively new and therefore still in progress. There is no single method that simultaneously provides robust feasibility guarantees, high quality solutions, and scalability to large instances.

The current state of research in the field of ACOPF with uncertainty, gives rise to the following research questions that we aim to investigate and answer in this thesis:

- How can we construct a high-quality and scalable approach to the ACOPF problem with uncertainty, that is suitable for moderate levels of uncertainty?
- How does this approach perform in terms of computation time and cost of the solution, and how does it compare to existing benchmark methods in terms of robust feasibility?

In the first research question, with high quality we mean that our approach finds a solution that is feasible to the nominal ACOPF model (that is, the model without uncertainty), robustly feasible (that is, feasible when uncertainty is realized) and in practice results in an objective value that is close to the objective value of the nominal ACOPF solution. With scalability we mean that our algorithm has a low computation time; ideally we would have as many linear and as few nonlinear constraints in our model as possible. With moderate levels of uncertainty, we mean uncertainty levels ranging from 1% up to 50% of the load (that is, the power demand). In the second research question, the benchmark methods we compare our approach with, are the solution to the nominal problem (with tightened bounds to make it more robust), and the Direct Current Optimal Power Flow (DCOPF) solution. The latter is explained in Section 2.2.

In order to research and answer these questions, we consider the ACOPF problem with uncertainty as a two-stage problem, and formulate it in the framework of *adjustable robust optimization* (ARO). In such a problem, in the first stage the uncertainty is not known and decisions need to

be made without this information. However, in the second stage the uncertainty is revealed, and decisions can be made with complete information. Decision variables in the first stage are called *control variables*, and second-stage variables are called *state variables*. In which category a variable belongs is discussed when the ACOPF problem is explained in detail in Section 3. Now, the goal in such an ARO setting is to find a solution for the control variables, such that there exists a solution for the state variables for every possible realization of uncertainty. To achieve this, ideally we would link the state variables to the control variables using an explicit *decision rule*. This means that given the values of the control variables and the revealed uncertainty, the values of the state variables follow directly from this decision rule. In other words, we would like to express the state variables as a function of the control variables and uncertainty. Finding this function in the ACOPF problem requires solving the earlier mentioned system of nonlinear equalities. As a result, we have no analytical representation for the rule under discussion; the decision rule is only given implicitly, via the system of equalities.

The method proposed in this thesis aims to solve the latter issue by iteratively approximating the full quadratic system of equalities with linear approximations. These approximations are taken on a small subset around specified points of approximation, so that they are close to the original quadratic constraints. Having a linear system of equalities, we eliminate all state variables and equalities, by expressing the state variables as an explicit function of control variables and uncertainty. As a final step, the uncertainty is eliminated from the problem using standard techniques in robust and convex optimization, and a formulation with only control variables is left. We end up with a second-order conic programming (SOCP) problem. However, if it is necessary to explicitly model the (quadratic) thermal limits for transmission lines as well, we model these constraints as semidefinite programming (SDP) constraints. For most instances, this turns out to be not necessary since the solution to the remaining problem already satisfies those thermal limits. When the remaining problem is solved, we obtain our solution for the control variables. The corresponding solution for the state variables follows in practice from solving the power flow equations after uncertainty is known. The implementation and evaluation of this method is the main contribution of this thesis.

Our approach differs from Lee et al. (2021) in the sense that we exploit the inherent two-stage structure of the ACOPF problem. Moreover, we do not use convex restrictions of the feasible space, and we do not approximate the inequality constraints of the problem. We do also not iteratively

tighten the inequality bounds as in Molzahn and Roald (2018) and Roald and Andersson (2017). Convex relaxations as in Molzahn and Roald (2018) are also not considered in our approach. Our approach also differs from Roald and Andersson (2017) in the sense that we do not use a stochastic approach and thus do not require distributional assumptions on the uncertainty. However, since we also propose linearizations around specified points, our method has similarities with the method proposed in Roald and Andersson (2017) as well. The solution method in this thesis more closely follows the approach of Kuryatnikova et al. (2021) for solving the ACOPF problem under uncertainty. However, Kuryatnikova et al. (2021) use a formulation of the ACOPF problem that contains many quadratic inequalities. Since they only approximate the equality constraints, their approximated problem still contains these nonlinear inequalities. We use a different formulation for the ACOPF problem, with as many linear inequality constraints as possible. The only inequalities that cannot be stated other than quadratic constraints, are the line thermal limits for the transmission lines. Because of this different formulation, we end up with more state variables and therefore more approximations, but we gain computational efficiency. Note that our formulation of the ACOPF model is equivalent to the formulation used in Kuryatnikova et al. (2021), but the problems become different after taking approximations.

We find that our method performs well in terms of computation time, cost and robust feasibility. For medium-sized instances, our algorithm is able to find a solution within seconds. For the largest instance we consider, our algorithm finds a solution within 20 minutes. However, there is potential to decrease computation time even further by setting up the model more efficiently. We also find that our robust solution increases the cost in the situation of no uncertainty only marginally: most cost increases are below 0.1%. Furthermore, we find that our method performs very well in terms of robust feasibility, compared to the two benchmark methods. For many instances, our solution is feasible for 100% of simulated uncertainty realizations from the specified uncertainty set, especially for lower levels of uncertainty.

The rest of this paper is organized as follows. Section 2 provides a review of the literature on adjustable robust optimization, as well as the literature on the ACOPF problem, and the ACOPF problem with uncertainty. In Section 3, we explain the ACOPF problem in more detail, and we provide a formulation for the model without uncertainty. In Section 4, we discuss how we model uncertainty in the problem and we formulate the model with uncertainty. In Section 5, we explain

the solution method in detail. We discuss the data we use for our experiments in Section 6. Section 7 provides the results of our algorithm. In Section 8, we simulate realizations of uncertainty and we evaluate the performance of our method in terms of robust feasibility. Finally, in Section 9 we state the main conclusions of this thesis, we provide a final discussion of our solution method and we propose directions for further research.

2 Literature

2.1 Adjustable robust optimization

Optimization problems can be formulated as mathematical models containing an objective function, constraints, variables and data. In real-life applications of these problems, data is often not completely known or uncertain. Problems for which this is the case, inherit the complexity of the deterministic problem as well as the additional complexity of the data being uncertain. Methods for optimization under uncertainty are generally developed in the framework of *stochastic optimization* or *robust optimization*. Stochastic optimization operates under the assumption that there is some knowledge on the distribution of the uncertain data. For example, *chance-constrained optimization* aims to obtain a solution that is feasible with a certain probability for some subset of constraints. To model such chance-constraints, we need information about the distribution of the uncertain parameters in those constraints. Robust optimization (RO), on the other hand, does not require prior information on the probability distribution of the uncertain data. The assumption that is made, is that this data lies in a certain uncertainty set. The goal of robust optimization is to obtain a solution that is feasible even in the worst-case uncertainty scenario that belongs to this uncertainty set.

Within the robust optimization framework, *adjustable robust optimization* (ARO) was introduced by Ben-Tal et al. (2004). ARO can be used to model problems that inherently consist of two stages. In such a two-stage problem, the realization of the uncertainty in the data is known in the second stage, but not yet in the first stage. As we mentioned before, the first-stage variables are also called *control variables*, and the second-stage variables are called *state variables*. The goal in an ARO setting is to find a solution for the control variables, such that a solution for the state variables exists for every possible realization of uncertainty from the prespecified uncertainty set. Unless some specific conditions hold, the solution to a problem differs depending on whether the problem is considered in an RO setting, or an ARO setting (Marandi and Den Hertog (2018)).

If the problem is suitable for ARO, formulating it as an ARO problem is likely to result in a solution that is less conservative, compared to considering the same problem in a standard RO setting.

To solve an ARO problem, one can consider methods to obtain an exact reformulation of the problem, such as cutting-plane algorithms (Bertsimas et al. (2012)) or vertex enumeration algorithms (Bienstock and Özbay (2008)). However, exact reformulations of ARO problems are often computationally intractable, except for cases that have *decision rules* of a specific form. In an ARO setting, decision rules define the relation between the control variables and the state variables. When such a decision rule can be stated as a function of the state variables in terms of control variables and uncertainty, it allows for a single-stage reformulation of the problem in which the state variables are eliminated. Even when explicit decision rules are known, ARO formulations are still often computationally intractable and researchers rely on approximations to solve the problem at hand. For example, a linear ARO can still turn into a non-convex quadratic programming problem if the decision rule is quadratic. Moreover, in many ARO problems explicit decision rules are not known in advance, adding an additional layer of complexity. One then needs to choose and optimize a decision rule to obtain the best objective value.

Now, in practice ARO problems can also contain systems of equalities in state variables. In that case, the choice for a decision rule is highly limited because the system of equalities needs to be robustly feasible. Optimization problems with equality constraints are generally known to be less suitable for robust optimization. One way to deal with equality constraints is by substituting certain variables by some expression defined by the system of equalities, eliminating both these variables and equality constraints from the problem. However, this is not possible when these expressions cannot be found analytically, which can be the case when the system is nonlinear. Such irremovable systems of equalities often occur in chemical process systems and in electric power systems. Recently, Isenberg et al. (2021) proposed a generalization of a robust cutting-set algorithm to deal with these nonlinear equalities, with applications on varying chemical process systems. Such a robust cutting-set algorithm iterates between solving a master problem and a separation problem. In the master problem, an optimal solution is found for a certain subset of uncertainty realizations, and in the separation problem violated uncertainty scenarios are identified and added to this subset. Lee et al. (2019) propose an approach to solve problems with nonlinear systems of equalities by solving a sequence of convex optimization problems. Under certain sufficient conditions they provide, this

method guarantees robustness and feasibility. A different approach is given in Kuryatnikova et al. (2021), who provide a general framework to solving ARO problems with systems of equalities. They propose to approximate the equality constraints by linear approximations on small subsets of the domain of the state variables. As we mentioned in Section 1, they also provide an application of their method to the ACOPF problem, which is discussed in Section 2.3.

2.2 The ACOPF problem

The Optimal Power Flow (OPF) problem aims to optimize a given cost function within an electric power system by deciding on the generated power and the voltages, which determine the power flow in the network. This operating point has to satisfy operational constraints such as voltage magnitude limits and limits on generated power, as well as the *power flow equations*. With power flow equations, we mean the system of equalities that controls the power flowing through the system. Before we discuss the history of OPF problems, we need to distinguish between *power flow analysis* and *optimal power flow analysis*. Classical power flow analysis only considers the power flow equations and aims to find the solution to it. That is, no objective function is optimized, and no other inequality constraints are considered. To find a unique solution to the power flow equations, the system needs to have as many variables as constraints. Because the system is actually underconstrained (it contains more variables than constraints), a certain set of variables is fixed in power flow analysis, and the resulting power flow equations need to be solved. The solution to the power flow equations is feasible for these equations, but not necessarily operationally feasible; it does not take into account limits on generated power or voltage magnitudes, for instance. Optimal power flow, on the other hand, considers the full underconstrained power flow equations, together with the operational limits. It turns the power flow problem into an optimization problem.

The OPF problem was first formulated by Carpentier (1962). Before that, power systems were often optimized based on experience of engineers, rules of thumb and tools such as network analyzers. Since the contribution of Carpentier (1962), many different formulations for the OPF problem have been developed, as well as many solution methods. Formulations can differ in objective function, constraints and underlying assumptions on the power network. However, every OPF formulation contains some form of the power flow equations. The power flow equations as formulated by Carpentier (1962), are often referred to as the Alternating Current (AC) power flow equations. These AC power flow equations consist of a nonlinear system of equalities. We call OPF

formulations that contain the full AC power flow equations ACOPF formulations. However, even these ACOPF models can be formulated in many different ways. Since the ACOPF problem contains voltages represented by complex numbers, formulations in polar form and rectangular form exist. Formulations also differ in whether power or current is used in the power flow equations, and whether power flows are represented by explicit variables. A review on the history of classical ACOPF formulations is presented in Cain et al. (2012). A more technical summary on the different mathematical programming ACOPF formulations is given in Bienstock et al. (2022). In Sadat and Kim (2021), numerical performance of different formulations of the ACOPF problem is studied.

Now, while formulations of the ACOPF may vary, they are all non-convex and NP-hard, which makes them computationally intractable. Since power systems can be very large and they need to be solved multiple times per day in most control rooms, simpler formulations than the ACOPF have been developed. A well-known version of the OPF that simplifies the AC power flow constraints, is the Direct Current Optimal Power Flow (DCOPF) (see e.g. Stott et al. (2009)). This name can be misleading, because it does not solve a direct current network (which means that current flows in one direction). Instead, the DCOPF is a linearized version of the ACOPF problem. This makes it suitable for obtaining a fast solution to an OPF problem, which is the reason why it is still often used in the power flow industry. However, the assumptions of a DCOPF model are generally very unrealistic. This makes the DCOPF only applicable for smaller networks, but for large power systems the levels of error become unacceptable.

Methods to solve the ACOPF in its original nonlinear form have been of interest since the introduction of the formulation. Since the ACOPF is NP-hard and power systems can be large, researchers rely heavily on approximations and heuristics to find optimal solutions. There are many important factors that an ACOPF solution method needs to take into account; Wang et al. (2007) argue that robustness, scalability and accuracy are the most important when developing new algorithms. According to Frank et al. (2012), almost every suitable mathematical programming approach has been applied to the ACOPF problem. More efficient solution methods are still constantly being developed. Even slightly better solution methods can save a lot of cost and time on a large scale, especially with increasing sizes of power systems all over the world and their increasing complexity. According to Cain et al. (2012), the approximate methods of today may still unnecessarily cost billions of dollars per year, even though they are already much more efficient than their predecessors.

2.3 The ACOPF problem with uncertainty

Since the ACOPF problem is already hard to solve to optimality, ACOPF problems with uncertainty in the data have not been studied extensively. Methods that do address this uncertainty, often simplify the nonlinear AC power flow equations. Many approaches use the DCOPF formulation of the problem. Since it contains a linearized version of the power flow equations, it is a computationally attractive method. Moreover, the DCOPF also has the benefit of being widely used in the classical OPF literature and in practice. For instance, Ding et al. (2016) consider the DCOPF in an ARO setting and obtain results for large instances within seconds. However, the DCOPF relies on unrealistic assumptions, which can cause the obtained solution to be unreliable in practice. Other methods that simplify the AC power flow equations, often relax the problem to obtain a convex formulation. Molzahn and Roald (2018) propose a robust method that uses such an approach; they replace the nonlinear power flow equations by convex relaxations. Their algorithm alternates between solving a deterministic ACOPF problem and computing tightenings of the inequality constraints. While their method lacks computational efficiency, they are the first to guarantee feasibility of the inequality constraints for all uncertainty realizations. More recent robust approaches that convexify the feasible space are proposed in Molzahn and Roald (2019) and Lee et al. (2021). Instead of convex relaxations, Lee et al. (2021) propose convex restrictions, which means that all points within this restricted space are feasible for all uncertainty realizations. They also derive a sufficient condition that is the first to guarantee robust feasibility of both the AC power flow equations, as well as the operational limits.

As with many robust optimization approaches, these methods can be conservative compared to stochastic or approximative approaches. Less conservative approaches to ACOPF with uncertainty can involve chance-constraints (see for example Zhang and Li (2011), Baker et al. (2016) and Venzke et al. (2017)). Roald and Andersson (2017) propose an analytical reformulation of the model with chance-constraints into closed form, which they achieve by linearizing the power flow equations around a specified operating point. They use an iterative algorithm to solve the problem, similar to Molzahn and Roald (2018). As mentioned before, Kuryatnikova et al. (2021) propose an approximative approach that also linearizes the power flow equations. They consider the ACOPF problem in the ARO framework and approximate the power flow equations by their first-order Taylor Series to obtain a linear formulation. This way, all state variables and equalities can be eliminated from the

problem. The resulting problem contains only control variables, but consists of both semidefinite programming (SDP) constraints and non-convex constraints. An alternating projections algorithm is used to solve this remaining problem.

3 Nominal ACOPF Model

3.1 Components of the ACOPF problem

As mentioned before, the Alternating Current Optimal Power Flow problem is a problem that occurs in electric power systems optimization. Such a power system can be seen as a network of electrical components that deals with the generation and distribution of power. Electric power is generated by *generators* and transmitted along *transmission lines*. We also refer to the latter as *lines* or *branches* throughout this paper. Electric power must be supplied to the consumers of the network. These consumers make up the nodes of the power network and are called *buses*. The amount of power demanded by a bus is called the *load*. Every generator is connected to a bus; we call those buses *generator buses*. In general, one bus can be connected to multiple generators, but in this thesis we restrict ourselves to at most one generator at a bus. However, our proposed approach can easily be extended to include multiple generators per bus. Next to generated power, engineers can also control *voltage* (electric pressure) in the network. Voltages can be controlled at every bus. Since the nonlinearities in the power flow equations of the problem are caused by products of voltages, deciding upon their values is a very important part of the ACOPF problem.

Now, the ACOPF problem has both inequality and equality constraints. As mentioned before, the equality constraints are called the power flow equations, and are the main cause of the non-convexity of the problem. These equations follow from a basic law in physics called Kirchhoff's Circuit Law and the π-model for modeling transmission lines (see Chatzivasileiadis (2018) for more background on this). The inequality constraints represent physical restrictions of the power system. These include limits on generated power, voltage limits and line thermal limits. The latter makes sure that transmission lines do not get too hot, since this can cause them to sag and eventually burn down. These equality and inequality constraints are described formally when the nominal ACOPF model is formulated in Section 3.3.

In Figure 1, an example of a power network is shown by means of a single-line diagram. Such

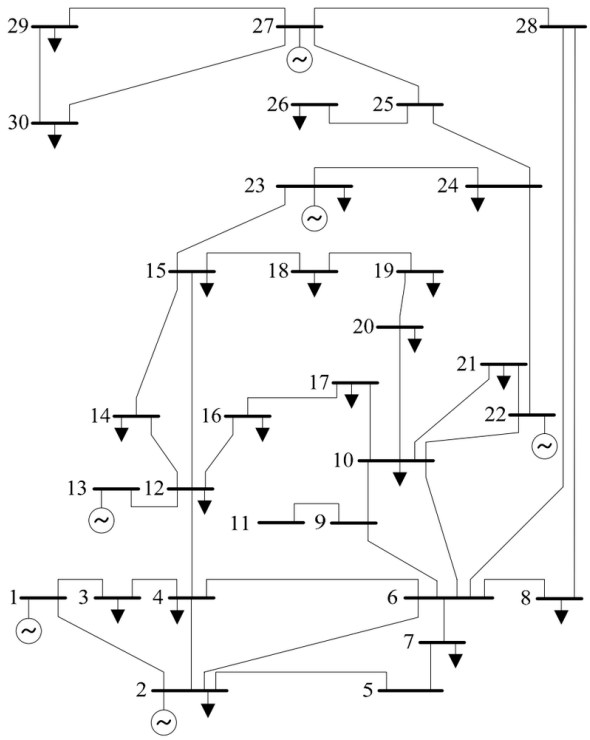

Figure 1: Single-line diagram of an IEEE test power system with 30 buses

a diagram is a simple representation of a rather complex electric power system. In the network displayed in Figure 1, there are six generators, represented by the tildes. We see that the power network can be seen as a connected graph. This is indeed the case, but certain elements of the ACOPF problem are very different compared to other graph problems. For example, power losses at branches may occur, which means the power flowing into a branch cannot simply be considered equal to the power flowing out of a branch. The power flow equations take these factors into account.

Another aspect of the ACOPF problem that is uncommon for optimization in other fields, is that power and voltages are represented by complex numbers. This is because the problem of optimizing generated power in a network is a dynamic problem, and thus the formulation of the problem is time-dependent. This dependency on time comes from the fact that power and voltage oscillate with a certain frequency. Formulating the ACOPF problem as a time-dependent problem would be, computationally speaking, not feasible with current technology (Bienstock et al. (2022)). However, representing these oscillating power and voltage waves by their average value would cause too much loss of information. Therefore, power and voltage (and other physical quantities that occur

in power systems, but those are not relevant for this research) are represented by complex values in the ACOPF literature. We consider the rectangular (also called Cartesian) form of these complex numbers. This means that the complex numbers are defined in terms of their real and imaginary part, and not in polar form. The real part of power is called *active power* and the imaginary part is called *reactive power*. Generation of active and reactive power can directly be controlled at generator buses. Voltages are controlled at all buses by means of their *voltage magnitude* and *phase angle*. The voltage magnitude is the magnitude of the complex voltage. The phase angle can be interpreted as the angle between the complex voltage and the positive real axis. Note that in our formulation the phase angle does not directly occur, since we decided to use the rectangular form of the complex number. However, the phase angles at buses can be easily derived if the real and imaginary parts of the voltage are known.

The goal of the ACOPF problem is to find an operating point for the power system in terms of generated active/reactive power, voltage magnitude and phase angles, that minimizes the cost function while all aforementioned equality and inequality constraints hold. We define the cost function as a quadratic function of active power generation, as is common in power flow optimization.

3.2 Reference bus, PV buses and PQ buses

The ACOPF problem contains many variables. In practice, only some of the variables are set up by the system engineers, at generator buses. These variables are the control variables of the problem. The number of remaining state variables is equal to the number of equality constraints defined by the power flow equations. The values of the state variables are not predetermined by the engineers, but they follow from solving the system of power flow equations. To know in which category a decision variable belongs, we need to discuss the three different kinds of buses of a power system.

We start with the *reference bus*. In general, the ACOPF model closely matches the behaviour of the real power system, but there are still some assumptions and approximations in the model. These can result in power mismatches in the system, which are spread over a number of generators. The generated power is not determined in advance for these buses, to allow the mismatches to be cancelled. The buses for which this is the case, are called reference buses. In our model we have one reference bus, since this is common in the data we use. The model can easily be generalized to any given set of multiple reference buses. In terms of variables, the active and reactive power are free

to vary (state variables) at the reference bus, while the voltage magnitude is predetermined by the engineer (control variable). The phase angle at the reference bus is fixed, and is set to zero. The rest of the generator buses are called *PV buses*. At PV buses, the generated active power and voltage magnitude are selected by the engineer (control variables), while the generated reactive power and voltage phase angle (and thus its real/imaginary part) are free to vary (state variables). The buses that are left are not connected to a generator, and are called *PQ buses*. Active and reactive power generated at PQ buses is thus zero, and the voltage magnitude and phase angle (and thus its real/imaginary part) can vary (state variables). PQ buses can, for example, represent locations that demand electricity but cannot produce it. The distinction between state variables and control variables is very important for our solution method, since we define the model with uncertainty in the ARO framework. Therefore, the vector of state variables and the vector of control variables are also stated in Section 5, when all further terminology has been introduced.

3.3 Formulation of the nominal ACOPF model

In this section, we describe our formulation of the nominal ACOPF model, which is the ACOPF model without uncertainty. The ACOPF problem can be formulated in a lot of different ways; we choose to follow the formulation provided by Bingane et al. (2018). Differences between our and their formulation occur because we assume that every bus has at most one generator connected to it. Also, their formulation contains tap ratios, which we do not consider (this implies that all tap ratios are equal to one). Since they are not relevant to our research, we do not dive deeper into what tap ratios are.

Before we formulate the nominal model, we introduce some relevant notation, data and variables. We denote by the $(\cdot)^*$ operator the conjugate of a complex number. We let N denote the set of all buses, $G \subset N$ the set of generator buses (which consists of all PV buses and the reference bus). Therefore, $N \setminus G$ denotes the set of PQ buses. The reference bus is chosen to have index 1. We denote by L the set of all branches. A branch has a $from$-end and a *to*-end, which can just be considered modeling choices. The need for this comes mainly from the fact that the ACOPF formulation considers power losses along a branch. We denote a branch by $l = (k, m)$, in which bus k is the $from$-end and bus m is the *to*-end. If we write the complex voltage at bus k in its rectangular formulation $v_k = v_k^r + \mathrm{j}v_k^j$, where j is the imaginary unit, we can define the variable $v_k^M := |v_k|^2 = (v_k^r)^2 + (v_k^j)^2$ as the squared voltage magnitude. We denote generated active power

at bus k by p_k^g and generated reactive power by q_k^g. We denote the loads (demands) at bus k by p_k^d and q_k^d for active and reactive power, respectively.

We also define the *line flow variables*, which we denote by p_l^f, q_l^f, p_l^t and q_l^t. These variables indicate the flow of (re)active power into branch l, injected by its $from$-end or its to-end. p_l^f is the active power flowing into branch l injected by its $from$-end. q_l^f is the reactive power flowing into branch l injected by its $from$-end. p_l^t and q_l^t indicate the active and reactive power flowing into branch l injected by its to-end, respectively. Note that these variables can take on both positive and negative values. For example, a positive value for p_l^f means that power is flowing into branch l, injected by the $from$-end. A negative value for p_l^f indicates that power flows along branch l into the $from$-end.

Now, there is also some data on electric components in the power system that is somewhat more technical. The reader is not expected to be familiar with the terms that follow; they can just be considered constants that are part of the constraints in the model. For every branch l, the *line admittance* y_l is given. The line admittance is a measure of how easily electric current is allowed to flow through a transmission line. The admittance is a complex number and can be written as $y_l = g_l + \mathrm{j}b_l$. Its real part g_l is called the *conductance*, and the imaginary part b_l is called the *susceptance*. Next to line admittances, a *shunt element* can be connected to both ends of a line. Such a shunt element can serve several purposes, such as circuit protection or creating a path of low resistance for electric current to flow around, 'shunting' the current to a different path. For a branch l, this shunt element is incorporated at both ends of the branch, in terms of susceptance only (usually, the shunt conductance of line l is negligible). The susceptance at both ends of the branch is considered equal, and denoted by $\frac{b_l^{sh}}{2}$, where b_l^{sh} is the total shunt susceptance of branch l. Finally, an additional shunt element at bus k is incorporated in the model in terms of its conductance g_k^{sh} and susceptance b_k^{sh}.

Below, a first version of the nominal ACOPF model is given:

$$\min \quad \sum_{k\in G} c_{k,2}(p_k^g)^2 + c_{k,1}p_k^g + c_{k,0} \tag{1a}$$

$$\text{s.t.} \quad p_k^g - p_k^d - g_k^{sh}v_k^M = \sum_{l=(k,m)\in L} p_l^f + \sum_{l=(m,k)\in L} p_l^t \qquad \forall k \in G \tag{1b}$$

$$q_k^g - q_k^d + b_k^{sh}v_k^M = \sum_{l=(k,m)\in L} q_l^f + \sum_{l=(m,k)\in L} q_l^t \qquad \forall k \in G \tag{1c}$$

$$- p_k^d - g_k^{sh}v_k^M = \sum_{l=(k,m)\in L} p_l^f + \sum_{l=(m,k)\in L} p_l^t \qquad \forall k \in N \setminus G \tag{1d}$$

$$- q_k^d + b_k^{sh}v_k^M = \sum_{l=(k,m)\in L} q_l^f + \sum_{l=(m,k)\in L} q_l^t \qquad \forall k \in N \setminus G \tag{1e}$$

$$p_l^f + \mathrm{j}q_l^f = v_k \left[\left(\mathrm{j}\frac{b_l^{sh}}{2} + y_l\right) v_k - y_l v_m\right]^* \qquad \forall l = (k,m) \in L \tag{1f}$$

$$p_l^t + \mathrm{j}q_l^t = v_m \left[\left(\mathrm{j}\frac{b_l^{sh}}{2} + y_l\right) v_m - y_l v_k\right]^* \qquad \forall l = (k,m) \in L \tag{1g}$$

$$p_k^{\min} \le p_k^g \le p_k^{\max} \qquad \forall k \in G \tag{1h}$$

$$q_k^{\min} \le q_k^g \le q_k^{\max} \qquad \forall k \in G \tag{1i}$$

$$\left|p_l^f + \mathrm{j}q_l^f\right| \le s_l^{\max} \qquad \forall l = (k,m) \in L \tag{1j}$$

$$\left|p_l^t + \mathrm{j}q_l^t\right| \le s_l^{\max} \qquad \forall l = (k,m) \in L \tag{1k}$$

$$v_k^{\min} \le |v_k| \le v_k^{\max} \qquad \forall k \in N \tag{1l}$$

$$v_1^j = 0 \tag{1m}$$

$$v_1^r \ge 0. \tag{1n}$$

In model (1), the objective (1a) denotes a quadratic cost function of generated active power, as is common in the literature and in practice. Constraints (1b)-(1e) are the power flow equations. Constraints (1b) and (1c) model these power flow equations for buses that are connected to a generator, for active and reactive power respectively. Here, the power generated at bus k minus the load at bus k, must be equal to the net power flowing out of bus k (modeled as the power flowing into a branch coming from bus k), adjusted for the shunt elements connected to bus k. If the left-hand side is negative, the generated power is not enough for the load needed. Then, the right-hand side is also negative, indicating that net power is injected into bus k from other buses, instead of flowing out of it. Constraints (1d) and (1e) model the power flow equations for buses that are not connected to a generator. Constraints (1f) and (1g) are the line flow constraints; they follow from the formula

$s = vi^*$ that is well known in physics. Here, s denotes complex power, i denotes current and v denotes voltage. For a branch $l = (k, m)$, it holds that the current flowing into the $from$-end k of that line, depends on the admittance and the voltages as follows: $i_l^f = y_{l,k}^{sh} v_k + y_l(v_k - v_m)$. Since $y_{l,k}^{sh} = \mathrm{j}\frac{b_l^{sh}}{2}$, we get that $s_l^f = v_k \cdot \left[\mathrm{j}\frac{b_l^{sh}}{2} v_k + y_l(v_k - v_m)\right]^*$, which is exactly what is represented in constraint (1f). For more information on the theory behind this, we refer to Chatzivasileiadis (2018). Constraint (1g) is derived in a similar way, but for the *to*-end of a branch $l = (k, m)$. Constraints (1h)-(1i) correspond to limits on active and reactive power, respectively. Note that $p_k^{\min}$ and $q_k^{\min}$ can be negative. For reactive power, this is more often the case than for active power. However, in some instances generators are modeled as aggregations of multiple loads and generators, resulting in a possibly negative value for $p_k^{\min}$. The only assumption we make here, is that the lower bound on active power generation at the reference bus ($p_1^{\min}$) is nonnegative. This assumption is necessary to obtain an objective function that is increasing in p_1^g, which is convenient when formulating the model with uncertainty in ARO form (see Section 4). This assumption holds for all instances of power systems we consider. Constraints (1j)-(1k) indicate the *line thermal limits*, which represent the heat limits on transmission lines. Constraint (1l) models the bounds on the voltage magnitudes. Constraints (1m) and (1n) set the voltage phase angle at the reference bus to zero, by restricting the imaginary part of that voltage to be zero and by restricting the real part of the voltage to be positive.

All quantities (of both variables and parameters) in model (1) are expressed in per-unit (p.u.). This means that quantities are given as a fraction of their base value. The per-unit system is common in power systems analysis, since it simplifies calculations and it diminishes large differences in absolute values. To convert back to other units, one needs to know the base units of the system at hand. However, this differs per instance and is not relevant to our research. Furthermore, an overview of all variables and parameters in the model is given in Table 1.

One may notice that v_k is a decision variable that takes on the value of a complex number. Moreover, there are many constraints that contain complex numbers. However, constraints (1b)-(1e) and (1j)-(1l) contain only the (squared) magnitude of a complex number or variable, which is a real number. Constraints (1f)-(1g) on the other hand are explicit complex constraints. We can avoid having to work with complex decision variables by decomposing both sides of (1f) and (1g) into a real and imaginary part. For the left-hand side this is obvious, but for the right-hand side this requires some simple but cumbersome derivations. As a result, we can rewrite constraint (1f) and

(1g) as the following four constraints:

$$\begin{aligned}
p_l^f &= g_l v_k^M - g_l \left(v_k^r v_m^r + v_k^j v_m^j\right) + b_l \left(v_k^r v_m^j - v_k^j v_m^r\right) && \forall l = (k,m) \in L \\
q_l^f &= \left(-\frac{b_l^{sh}}{2} - b_l\right) v_k^M + g_l \left(v_k^r v_m^j - v_k^j v_m^r\right) + b_l \left(v_k^r v_m^r + v_k^j v_m^j\right) && \forall l = (k,m) \in L \\
p_l^t &= g_l v_m^M - g_l \left(v_k^r v_m^r + v_k^j v_m^j\right) + b_l \left(v_k^j v_m^r - v_k^r v_m^j\right) && \forall l = (k,m) \in L \\
q_l^t &= \left(-\frac{b_l^{sh}}{2} - b_l\right) v_m^M + g_l \left(v_k^j v_m^r - v_k^r v_m^j\right) + b_l \left(v_k^r v_m^r + v_k^j v_m^j\right) && \forall l = (k,m) \in L.
\end{aligned} \tag{2}$$

We call expressions (2) the *line flow expressions*. Note that here the admittance appears in terms of conductance and susceptance ($y_l = g_l + \mathrm{j}b_l$). We replace constraints (1f)-(1g) in the model by constraints (2) so that we have constraints that only contain real numbers. Also, since constraint (1l) contains only positive terms, we can replace that constraint by:

$$(v_k^{\min})^2 \leq v_k^M \leq (v_k^{\max})^2 \quad \forall k \in N. \tag{3}$$

This is convenient since we treat the squared magnitude v_k^M as a decision variable. The definition of this squared magnitude needs to be modeled explicitly too. Thus, we include the following constraint:

$$v_k^M = (v_k^r)^2 + (v_k^j)^2 \quad \forall k \in N. \tag{4}$$

We also square constraints (1j) and (1k) to get rid of the magnitude (which would contain a square root):

$$\begin{aligned}
(p_l^f)^2 + (q_l^f)^2 &\leq (s_l^{\max})^2 && \forall l = (k,m) \in L \\
(p_l^t)^2 + (q_l^t)^2 &\leq (s_l^{\max})^2 && \forall l = (k,m) \in L.
\end{aligned} \tag{5}$$

The nominal ACOPF model, without complex numbers, can then be defined by (1a)-(1e), (1h)-(1i), (1m)-(1n), (2)-(5). In the next section, we introduce uncertainty, and we provide a complete overview of the model with uncertainty.

4 Incorporating Uncertainty

In the literature on ACOPF with uncertainty, the uncertainty is usually modeled in the load of the buses. This means that an uncertainty term ζ_k is contained in the power flow equations for bus k, if the load is nonzero. One can also choose to consider uncertainty in the generation of power, which

means that ζ_k is contained in the equations for the generator buses. We only consider uncertainty in the load, but our solution method can be easily applied to a situation with uncertain generation as well. We denote the set of buses with uncertainty by U. This set exists of all buses that have nonzero active power load.

To account for the changes in the amount of power in the system of equalities due to uncertainty, the amount of generated power needs to be adjusted. However, active power is generated at PV buses before uncertainty is known (see Section 3.2). This means that the reference bus needs to take into account the sum of all ζ_k, since this is the only generator bus for which the amount of generated active power is decided after uncertainty is revealed. Thus, p_1^g is replaced by $p_1^g - \sum_{k \in U} \zeta_k$ in the power flow equations. This is called a linear corrective control policy and lies within the framework of Automatic Generation Control (AGC) (Kumar et al. (2005)). AGC is a system to model the distribution of the power mismatches that occur.

Regarding reactive power, we assume the uncertainty can be expressed as $\alpha_k \zeta_k$ for bus $k \in U$. We follow the literature here (see e.g. Venzke et al. (2017) and Molzahn and Roald (2018)) by modeling the reactive power fluctuations via a constant *power factor*. This power factor is denoted by $\cos\theta_k$, in which $\theta_k = \tan^{-1}\left(q_k^d / p_k^d\right)$. The power factor is the ratio of the active power load to the magnitude of complex power; $\cos\theta_k$ is just one way to calculate this. Again, we follow Venzke et al. (2017) and Molzahn and Roald (2018) by modeling α_k as follows:

$$\alpha_k = \frac{\sqrt{1 - \cos^2\theta_k}}{\cos\theta_k} \quad \forall k \in U,$$

and $\alpha_k = 0$ for every bus $k \in N \setminus U$. All variables q_k^g are state variables, so their value is decided upon after uncertainty factors in. Therefore, there is no need to take into account all uncertainty values for reactive power in the reference bus.

Now, we assume that the uncertainty itself comes from an ellipsoidal set described by

$$\Omega = \{\boldsymbol{\zeta} \in \mathbb{R}^{|U|} : \boldsymbol{\zeta}^T \Sigma^{-1} \boldsymbol{\zeta} \leq r^2\}, \tag{6}$$

with a positive definite matrix Σ and radius r. Since the normal distribution is commonly used to model uncertainty in load (see for example Verbic and Canizares (2006)), we relate our uncer-

tainty set to the normal distribution as follows. First, we use the 95th percentile of the normal distribution as value for r. This means that $r = q_{0.95} = 1.645$. If ζ is normally distributed with variance-covariance matrix Σ, making one linear constraint robust against uncertainty set (6) with $r = q_{0.95}$ is equivalent to enforcing that constraint to hold with at least 95% (Ben-Tal et al. (2009)). However, as the number of constraints gets big for large instances, the probability that all constraints hold simultaneously tends to zero. Raising the value of r to account for this can result in an overconservative approach, so we choose to use $r = 1.645$. Second, Σ is the variance-covariance matrix of a normal distribution. We consider two different scenarios for Σ: with and without correlation.

In the case of no correlation, we define Σ as a diagonal matrix with the variances of the uncertainty elements on the diagonal, denoted by σ_k^2 for bus k. We define the standard deviations by $\sigma_k = \omega \cdot p_k^d$ for uncertainty ζ_k. Here, the uncertainty factor ω determines what proportion of the load we use to model the uncertainty. The different values used for ω are described in Section 7.

In the case of correlation, we define $\boldsymbol{\sigma}$ to be the $|U|$-dimensional vector of standard deviations σ_k. We generate a $|U| \times |U|$ matrix Φ, of which every element is drawn from a standard normal distribution. Then, $\Phi\Phi^T$ is a symmetric and positive semidefinite matrix. We use this matrix as variance-covariance matrix, but we rescale it to obtain the correct variances σ_k^2 on the diagonal. This is done as follows:

$$\Sigma = \operatorname{diag}\left(\boldsymbol{\sigma} \oslash \left(\operatorname{diag}(\Phi\Phi^T)\right)^{\circ\frac{1}{2}}\right) \Phi\Phi^T \operatorname{diag}\left(\boldsymbol{\sigma} \oslash \left(\operatorname{diag}(\Phi\Phi^T)\right)^{\circ\frac{1}{2}}\right).$$

Here, the operator $\oslash$ indicates element-wise division and the operator $(\cdot)^{\circ\frac{1}{2}}$ takes the element-wise square root of a matrix or vector. The $\operatorname{diag}(\cdot)$ operator turns a matrix into a vector of its diagonal, or it turns a vector into a diagonal matrix.

Now that we know how uncertainty is modeled, we can state the ACOPF model with uncertainty. This model is formulated in classical ARO form, and is given below:

$$\min \quad \sum_{k \in G \setminus \{1\}} \left[c_{k,2}(p_k^g)^2 + c_{k,1}p_k^g + c_{k,0}\right] + c_{1,2}t^2 + c_{1,1}t + c_{1,0} \tag{7a}$$

$$\text{s.t.} \quad p_k^{\min} \leq p_k^g \leq p_k^{\max} \qquad \forall k \in G \setminus \{1\} \tag{7b}$$

$$(v_k^{\min})^2 \leq v_k^M \leq (v_k^{\max})^2 \qquad \forall k \in G \qquad (7c)$$

$$t \leq p_1^{\max} \qquad (7d)$$

$$v_1^r \geq 0 \qquad (7e)$$

$$v_1^j = 0 \qquad (7f)$$

$$v_1^M = (v_1^r)^2 \qquad (7g)$$

and for each $\boldsymbol{\zeta} \in \Omega$, there exists

$$\left\{ q_k^g \ \forall k \in G, \quad v_k^M \ \forall k \in N \setminus G, \quad v_k^r \ \forall k \in N \setminus \{1\}, \quad v_k^j \ \forall k \in N \setminus \{1\}, \quad p_1^g \right\}$$

such that

$$p_1^g - \sum_{k \in U \setminus \{1\}} \zeta_k - p_1^d - g_1^{sh} v_1^M = \sum_{l=(1,m) \in L} p_l^f + \sum_{l=(m,1) \in L} p_l^t \qquad (7h)$$

$$p_k^g + \zeta_k - p_k^d - g_k^{sh} v_k^M = \sum_{l=(k,m) \in L} p_l^f + \sum_{l=(m,k) \in L} p_l^t \qquad \forall k \in G \setminus \{1\} \qquad (7i)$$

$$- p_k^d + \zeta_k - g_k^{sh} v_k^M = \sum_{l=(k,m) \in L} p_l^f + \sum_{l=(m,k) \in L} p_l^t \qquad \forall k \in N \setminus G \qquad (7j)$$

$$q_k^g + \alpha_k \zeta_k - q_k^d + b_k^{sh} v_k^M = \sum_{l=(k,m) \in L} q_l^f + \sum_{l=(m,k) \in L} q_l^t \qquad \forall k \in G \qquad (7k)$$

$$- q_k^d + \alpha_k \zeta_k + b_k^{sh} v_k^M = \sum_{l=(k,m) \in L} q_l^f + \sum_{l=(m,k) \in L} q_l^t \qquad \forall k \in N \setminus G \qquad (7l)$$

$$p_l^f = g_l v_k^M - g_l \left(v_k^r v_m^r + v_k^j v_m^j \right) + b_l \left(v_k^r v_m^j - v_k^j v_m^r \right) \qquad \forall l = (k,m) \in L \qquad (7m)$$

$$q_l^f = \left(-\frac{b_l^{sh}}{2} - b_l \right) v_k^M + g_l \left(v_k^r v_m^j - v_k^j v_m^r \right) + b_l \left(v_k^r v_m^r + v_k^j v_m^j \right) \qquad \forall l = (k,m) \in L \qquad (7n)$$

$$p_l^t = g_l v_m^M - g_l \left(v_k^r v_m^r + v_k^j v_m^j \right) + b_l \left(v_k^j v_m^r - v_k^r v_m^j \right) \qquad \forall l = (k,m) \in L \qquad (7o)$$

$$q_l^t = \left(-\frac{b_l^{sh}}{2} - b_l \right) v_m^M + g_l \left(v_k^j v_m^r - v_k^r v_m^j \right) + b_l \left(v_k^r v_m^r + v_k^j v_m^j \right) \qquad \forall l = (k,m) \in L \qquad (7p)$$

$$p_1^{\min} \leq p_1^g \leq t \qquad (7q)$$

$$q_k^{\min} \leq q_k^g \leq q_k^{\max} \qquad \forall k \in G \qquad (7r)$$

$$(p_l^f)^2 + (q_l^f)^2 \leq (s_l^{\max})^2 \qquad \forall l = (k,m) \in L \qquad (7s)$$

$$(p_l^t)^2 + (q_l^t)^2 \leq (s_l^{\max})^2 \qquad \forall l = (k,m) \in L \qquad (7t)$$

$$(v_k^{\min})^2 \leq v_k^M \leq (v_k^{\max})^2 \qquad \forall k \in N \setminus G \qquad (7u)$$

$$v_k^M = (v_k^r)^2 + (v_k^j)^2 \qquad \forall k \in N \setminus \{1\}. \qquad (7v)$$

In model (7), constraints (7b)-(7g) denote the domain of the control variables; they do not contain any state variables. The text below that indicates that we want to find a solution to the control variables such that for all possible values of uncertainty, there exists a solution for the state variables that satisfies constraints (7h)-(7v). Note that the uncertainty appears only in the power flow constraints (7h)-(7l). Technically, the uncertainty should only appear in these constraints for buses $k \in U$, but for readability of the model we decided to write it in this way. Note that $\zeta_k = 0$ for $k \in N \setminus U$.

One might also notice that in model (7), an additional variable t is included. This variable is necessary because in our objective function (7a), the state variable p_1^g would otherwise occur. Since we cannot have state variables in the objective function of an ARO formulation, we need to replace this variable with a control variable. The control variable t represents the worst-case value for p_1^g and can be contained in the objective function instead of p_1^g. To see why, note that the objective is convex and non-decreasing in p_1^g. This is true because all costs $c_{k,2}, c_{k,1}, c_{k,0}$ are nonnegative and we assumed $p_1^{\min}$ to be nonnegative.

The power flow equations (7h)-(7l) result in a quadratic system of equalities due to constraints (7m)-(7p). Therefore, exact methods to solve model (7) are computationally intractable and we try to solve the model with an approximate approach. This approach is explained in the following section.

5 Methodology

5.1 General idea of the approach

In this section, we explain our approach to solving the ACOPF under uncertainty in more detail. The goal of our method is to obtain a solution for the control variables, such that there exists a solution for the state variables after uncertainty has contaminated the power system (this solution for the state variables is obtained by solving the resulting power flow equations when uncertainty is known). Since we are interested in a solution for the control variables, we want to reformulate the problem such that it contains only control variables. This would be easy if we had a way to express the state variables as a function of control variables, since then we could eliminate the state variables from the problem by substitution. However, there is no explicit analytical form for this

function, because the power flow equations are nonlinear in the state variables. Therefore, we linearize these power flow equations. We do this by taking first-order Taylor Series approximations of the equality constraints in certain points of approximation. The intuition behind this is as follows: the system of equations defines the decision rule for the state variables as an implicit function of control variables. The first-order Taylor Series approximations of this implicit function closely resemble the original function around these points of approximation. For more details on the intuition and theory behind this, see Kuryatnikova et al. (2021). Since the Taylor Series approximations are only a good representation of the implicit function in the space around the points of approximation, we restrict the space in which we search for a better solution to be close to these points (how we do this exactly is discussed in Section 5.8). We consider multiple points of approximation iteratively, instead of simultaneously. For now, consider one point of approximation. As a result of the linearization in this point, we obtain a linear system of equations that has as many equations as state variables. Therefore, we can express the state variables as a function of the control variables and the uncertainty, and we can eliminate the equations and state variables from the problem by substitution. Consequently, we obtain a standard robust optimization problem with inequalities. After eliminating the uncertainty from the problem as well, using techniques from robust and convex optimization, the approximated problem contains only control variables and we solve the resulting problem. If the power flow equations in the nominal problem are solvable using this solution for control variables, we obtain a solution that is both nominally feasible and robustly feasible to the approximated problem. Here we mean robust feasibility of the inequalities in the problem, since robust feasibility of the original equalities can only be guaranteed for the nominal case. However, if small enough subsets are used for approximations and the uncertainty set is small, the equalities become robustly feasible as well, see Kuryatnikova et al. (2021).

As mentioned before, we iteratively consider different points of approximation. When we find a solution for the control variables and for the state variables by solving the power flow equations in the nominal problem, we use this solution as the next point of approximation. This way, we iteratively change the neighbourhood in which we search for a robust solution.

5.2 Solution procedure

From now on we distinguish the following vector of state variables $\boldsymbol{x}$ and vector of control variables $\boldsymbol{y}$:

$$\boldsymbol{x} = \begin{bmatrix} q_k^g & \forall k \in G \\ v_k^M & \forall k \in N \setminus G \\ v_k^r & \forall N \setminus \{1\} \\ v_k^j & \forall N \setminus \{1\} \\ p_1^g & \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} p_k^g & \forall k \in G \setminus \{1\} \\ v_k^M & \forall k \in G \\ v_1^r & \\ t & \end{bmatrix}. \tag{8}$$

Note that the line flow variables $p_l^f, q_l^f, p_l^t, q_l^t$ do not appear in (8). This is because we substitute (the linearized versions of) these variables directly into the power flow equations, so the line flow variables themselves do not appear in the problem anymore. Finally, note that the imaginary part of the voltage at the reference bus (v_1^j) is included in neither $\boldsymbol{x}$ nor $\boldsymbol{y}$. Remember that this variable is set to zero by constraint (7f). We included it in model (7) for readability of the model, but from now on we disregard this variable.

Next we describe the main steps of the suggested algorithm. Later on, Section 5.13 gives details of the algorithm and Algorithm 1 shows a pseudocode for the final implementation. The proposed algorithm iteratively considers small subsets of the state variables and linearizes the equality constraints over these subsets. We begin with an initial solution $(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{y}}_1)$. We also define an initial small subset $S_{\hat{\boldsymbol{x}}_1}$ around $\hat{\boldsymbol{x}}_1$, such that $\hat{\boldsymbol{x}}_1 \in S_{\hat{\boldsymbol{x}}_1}$. The choice for this subset is discussed in Section 5.8. We take first-order Taylor Series approximations in this point. We do these approximations for the power flow equations (to be precise, we take approximations of the line flow expressions (7m)-(7p) and substitute them in the power flow constraints (7h)-(7l)), but also for the nonlinear voltage constraints (7v), to obtain a linear system of equalities. Then we express $\boldsymbol{x}$ in terms of $\boldsymbol{y}$ and $\boldsymbol{\zeta}$ and we substitute the state variables so that only $\boldsymbol{y}$ and $\boldsymbol{\zeta}$ are left in the problem. Subsequently, we eliminate $\boldsymbol{\zeta}$ and an equivalent SOCP (or SDP, depending on if the line thermal constraints are included) with only control variables $\boldsymbol{y}$ is left. The solution $\hat{\boldsymbol{y}}_2$ to this problem is robustly feasible for the approximated problem. The corresponding $\hat{\boldsymbol{x}}_2$ is found by solving the original power flow equations with $\boldsymbol{y} = \hat{\boldsymbol{y}}_2$ and $\boldsymbol{\zeta} = \boldsymbol{0}$. If this solution exists, we have found an approximately robustly feasible solution $\hat{\boldsymbol{y}}_2$. Then, we have our next point $(\hat{\boldsymbol{x}}_2, \hat{\boldsymbol{y}}_2)$ and corresponding small subset $S_{\hat{\boldsymbol{x}}_2}$,

we obtain a new linear approximation in this point, and the steps mentioned above are repeated. If the decrease in objective value of the approximated problem is less than a prespecified value δ, we stop the algorithm and we have found a locally optimal approximately robust solution $\boldsymbol{y}^*$. Note that a solution for the state variables $\hat{\boldsymbol{x}}$ is needed to linearize the system of equalities in the next iteration, but does not denote the optimal solution for the state variables for the ACOPF problem with uncertainty, since this solution depends on the realized uncertainty.

To obtain the initial solution $(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{y}}_1)$, we use MATPOWER (Zimmerman et al. (2010)). MATPOWER is an open-source MATLAB-based package used for simulation and optimization of power systems. It contains many instances of power systems; we discuss the instances we use in Section 6. MATPOWER contains an option 'runopf', which finds a (locally) optimal solution to the nominal ACOPF problem within seconds. The method that MATPOWER uses here, is a primal/dual interior point method called Matpower Interior Point Solver (MIPS). For more detailed information on this solver and on MATPOWER in general, we refer to Zimmerman et al. (2010), who provide a comprehensive manual. We also use MATPOWER to solve the power flow equations once we have found a solution for the control variables. The option we use for this is called 'runpf'. The method that is used to solve the power flow, is Newton's method (Tinney and Hart (1967), Peschon et al. (1968)).

To obtain the state variables as a function of the control variables, first we write the line flow expressions (7m)-(7p) in matrix form in Sections 5.3 until 5.5. In Section 5.6, we take Taylor Series approximations of these expressions, substitute them in the power flow equations (7h)-(7l) and thus eliminate the line flow variables and their corresponding constraints (7m)-(7p) from the problem. We derive the expression of $\boldsymbol{x}$ in terms of $\boldsymbol{y}$ and $\boldsymbol{\zeta}$ and we eliminate $\boldsymbol{x}$ from the problem in Section 5.7. In Section 5.8, we explain how we define the small subset around the point of approximation. In Section 5.9, $\boldsymbol{\zeta}$ is eliminated from the problem. In Section 5.10, we explain how to deal with the non-convex constraint (7g). We state the full approximated problem in Section 5.11. In Section 5.12, we explain how the line thermal constraints (7s)-(7t) can be included in the model. Finally, we give a pseudocode of the algorithm in Section 5.13.

5.3 Line flow expressions for PV and PQ buses

First, we consider the line flow expressions (7m)-(7p) for all branches that are not connected to the reference bus, meaning that $k \neq 1$ and $m \neq 1$ for $l = (k, m)$. We can also denote this by $l \in L \setminus L_r$, where L_r denotes the set of branches connected to the reference bus. We can write constraints (7m)-(7p) in the form

$$\begin{aligned}
p_l^f &= \boldsymbol{x}^T W_{p_l^f} \boldsymbol{x} + \boldsymbol{w}_{p_l^f}^T \boldsymbol{x} + \boldsymbol{z}_{p_l^f}^T \boldsymbol{y} \qquad \forall l = (k, m) \in L \setminus L_r \\
q_l^f &= \boldsymbol{x}^T W_{q_l^f} \boldsymbol{x} + \boldsymbol{w}_{q_l^f}^T \boldsymbol{x} + \boldsymbol{z}_{q_l^f}^T \boldsymbol{y} \qquad \forall l = (k, m) \in L \setminus L_r \\
p_l^t &= \boldsymbol{x}^T W_{p_l^t} \boldsymbol{x} + \boldsymbol{w}_{p_l^t}^T \boldsymbol{x} + \boldsymbol{z}_{p_l^t}^T \boldsymbol{y} \qquad \forall l = (k, m) \in L \setminus L_r \\
q_l^t &= \boldsymbol{x}^T W_{q_l^t} \boldsymbol{x} + \boldsymbol{w}_{q_l^t}^T \boldsymbol{x} + \boldsymbol{z}_{q_l^t}^T \boldsymbol{y} \qquad \forall l = (k, m) \in L \setminus L_r.
\end{aligned} \tag{9}$$

Here, $W_{p_l^f}, W_{q_l^f}, W_{p_l^t}, W_{q_l^t}$ are large symmetric matrices which exist of only zeros, except for at most eight elements in each matrix. In Table 2, the indices of these elements are indicated, together with the corresponding values in W. For example, the index pair (v_k^r, v_m^r) implies the entry in W that corresponds to the indices of v_k^r and v_m^r in $\boldsymbol{x}$.

Table 2: Index pairs and corresponding values for matrices W in (9).

Index pair	$W_{p_l^f}$ value	$W_{q_l^f}$ value	$W_{p_l^t}$ value	$W_{q_l^t}$ value
(v_k^r, v_m^r)	$-g_l/2$	$b_l/2$	$-g_l/2$	$b_l/2$
(v_k^r, v_m^j)	$b_l/2$	$g_l/2$	$-b_l/2$	$-g_l/2$
(v_k^j, v_m^r)	$-b_l/2$	$-g_l/2$	$b_l/2$	$g_l/2$
(v_k^j, v_m^j)	$-g_l/2$	$b_l/2$	$-g_l/2$	$b_l/2$

Here, the values g_l and b_l denote the conductance and susceptance of branch l, respectively (see Table 1). Note that in Table 2, if an index pair implies that entry (i, j) has a certain value, it also implies that entry (j, i) in the matrix W has that value, since the matrices are symmetric.

5.4 Line flow expressions for reference bus

In the more specific case when either k or m denotes the reference bus, constraints (7m)-(7p) cannot be written in the same form as (9). This is because v_1^r is not a state variable but a control variable.

Constraints (7m)-(7p) can be written as follows:

$$\begin{aligned} p_l^f &= \boldsymbol{x}^T W_{p_l^f} \boldsymbol{y} + \boldsymbol{w}_{p_l^f}^T \boldsymbol{x} + \boldsymbol{z}_{p_l^f}^T \boldsymbol{y} \qquad \forall l = (k,m) \in L_r \\ q_l^f &= \boldsymbol{x}^T W_{q_l^f} \boldsymbol{y} + \boldsymbol{w}_{q_l^f}^T \boldsymbol{x} + \boldsymbol{z}_{q_l^f}^T \boldsymbol{y} \qquad \forall l = (k,m) \in L_r \\ p_l^t &= \boldsymbol{x}^T W_{p_l^t} \boldsymbol{y} + \boldsymbol{w}_{p_l^t}^T \boldsymbol{x} + \boldsymbol{z}_{p_l^t}^T \boldsymbol{y} \qquad \forall l = (k,m) \in L_r \\ q_l^t &= \boldsymbol{x}^T W_{q_l^t} \boldsymbol{y} + \boldsymbol{w}_{q_l^t}^T \boldsymbol{x} + \boldsymbol{z}_{q_l^t}^T \boldsymbol{y} \qquad \forall l = (k,m) \in L_r. \end{aligned} \tag{10}$$

The difference between (9) and (10) is that in the case of the reference bus, the line flow expressions contain a product of state variables with a control variable. When defining the matrices W, we end up with matrices that have at most two non-zero elements, instead of eight. This is because $v_1^j = 0$ and is thus not a variable in $\boldsymbol{x}$ nor $\boldsymbol{y}$, together with the fact that the matrices W in (10) are not symmetric anymore. Below, the specifications for W are given for the case when $k = 1$ and when $m = 1$ in $l = (k,m) \in L_r$:

Table 3: Index pairs and corresponding values for matrices W in (10).

	Index pair	$W_{p_l^f}$ value	$W_{q_l^f}$ value	$W_{p_l^t}$ value	$W_{q_l^t}$ value
$l = (1,m)$	(v_m^r, v_1^r)	$-g_l$	b_l	$-g_l$	b_l
	(v_m^j, v_1^r)	b_l	g_l	$-b_l$	$-g_l$
$l = (k,1)$	(v_k^r, v_1^r)	$-g_l$	b_l	$-g_l$	b_l
	(v_k^j, v_1^r)	$-b_l$	$-g_l$	b_l	g_l

5.5 Linear parts of line flow expressions for all buses

Until this point, we have only defined the matrices W in (9) and (10), but we still have to specify the vectors $\boldsymbol{w}$ and $\boldsymbol{z}$ in these expressions. These specifications depend only on whether the voltage magnitude in the expression is a state variable (in the case of PQ buses) or a control variable (in the case of PV buses and the reference bus). This means we don't have to treat the specific case of the reference bus differently from other generator buses. Thus, the values of the vectors $\boldsymbol{w}$ and $\boldsymbol{z}$

differ depending on whether bus k in $l = (k, m)$ is a generator or a non-generator bus:

$$\boldsymbol{w}_{p_l^f} = [..., g_l, ...]^T, \boldsymbol{w}_{q_l^f} = \left[..., -\frac{b_l^{sh}}{2} - b_l, ...\right]^T, \quad \boldsymbol{z}_{p_l^f} = \boldsymbol{z}_{q_l^f} = \boldsymbol{0} \quad \forall l = (k, m) \in L, \quad k \in N \setminus G$$

$$\boldsymbol{z}_{p_l^f} = [..., g_l, ...]^T, \quad \boldsymbol{z}_{q_l^f} = \left[..., -\frac{b_l^{sh}}{2} - b_l, ...\right]^T, \quad \boldsymbol{w}_{p_l^f} = \boldsymbol{w}_{q_l^f} = \boldsymbol{0} \quad \forall l = (k, m) \in L, \quad k \in G$$

$$\boldsymbol{w}_{p_l^t} = [..., g_l, ...]^T, \quad \boldsymbol{w}_{q_l^t} = \left[..., -\frac{b_l^{sh}}{2} - b_l, ...\right]^T, \quad \boldsymbol{z}_{p_l^t} = \boldsymbol{z}_{q_l^t} = \boldsymbol{0} \quad \forall l = (k, m) \in L, \quad m \in N \setminus G$$

$$\boldsymbol{z}_{p_l^t} = [..., g_l, ...]^T, \quad \boldsymbol{z}_{q_l^t} = \left[..., -\frac{b_l^{sh}}{2} - b_l, ...\right]^T, \quad \boldsymbol{w}_{p_l^t} = \boldsymbol{w}_{q_l^t} = \boldsymbol{0} \quad \forall l = (k, m) \in L, \quad m \in G.$$

In the top line here, the index of the only possibly non-zero element in both vectors $\boldsymbol{w}$ corresponds to the index of v_k^M in $\boldsymbol{x}$. In the second line, the index of the possibly non-zero element in both vectors $\boldsymbol{z}$ corresponds to the index of v_k^M in $\boldsymbol{y}$. In the third line, it corresponds to the index of v_m^M in $\boldsymbol{x}$. In the last one it corresponds to the index of v_m^M in $\boldsymbol{y}$.

5.6 Linearizing the system of equalities

Now that we wrote the line flow expressions in matrix form, we can start linearizing them. We can write constraints (9) in a general form as follows:

$$F_1(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T W \boldsymbol{x} + \boldsymbol{w}^T \boldsymbol{x} + \boldsymbol{z}^T \boldsymbol{y}.$$

We can take the first-order Taylor Series of $F_1(\boldsymbol{x}, \boldsymbol{y})$ in a certain point of approximation $\hat{\boldsymbol{x}}$ (we do not need $\hat{\boldsymbol{y}}$ since there are no nonlinearities in $\boldsymbol{y}$ here) as follows:

$$T_1(\hat{\boldsymbol{x}}) = \left(2\hat{\boldsymbol{x}}^T W + \boldsymbol{w}^T\right) \boldsymbol{x} - \hat{\boldsymbol{x}}^T W \hat{\boldsymbol{x}} + \boldsymbol{z}^T \boldsymbol{y}. \tag{11}$$

Now, remember that for the reference bus, the line flow variables could only be written in the form (10). These can be written in a general form as follows:

$$F_2 = (\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T W \boldsymbol{y} + \boldsymbol{w}^T \boldsymbol{x} + \boldsymbol{z}^T \boldsymbol{y}.$$

The first-order Taylor Series of $F_2(\boldsymbol{x}, \boldsymbol{y})$ in the point of approximation $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ is as follows:

$$T_2(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) = \left(\hat{\boldsymbol{y}}^T W^T + \boldsymbol{w}^T\right) \boldsymbol{x} + \left(\hat{\boldsymbol{x}}^T W + \boldsymbol{z}^T\right) \boldsymbol{y} - \hat{\boldsymbol{x}}^T W \hat{\boldsymbol{y}}. \tag{12}$$

We denote by $T_{p_l^f}, T_{p_l^t}, T_{q_l^f}, T_{q_l^t}$ the first-order Taylor Series of the line flow variables $p_l^f, p_l^t, q_l^f, q_l^t$. They are of the form (11) if $l \in L \setminus L_r$, and they are of the form (12) if $l \in L_r$. If we substitute the line flow variables by these first-order Taylor Series and substitute those in the power flow equations (7h)-(7l), we get a linear system of equalities in terms of all state variables. For example, for a certain bus $k \in G \setminus \{1\}$, constraint (7i) becomes the following:

$$\begin{aligned} p_k^g + \zeta_k - p_k^d - g_k^{sh} v_k^M = & \left[\sum_{l=(k,m)\in L} \boldsymbol{w}_{p_l^f}^T + \sum_{l=(m,k)\in L} \boldsymbol{w}_{p_l^t}^T \right] \boldsymbol{x} + \left[\sum_{l=(k,m)\in L} \boldsymbol{z}_{p_l^f}^T + \sum_{l=(m,k)\in L} \boldsymbol{z}_{p_l^t}^T \right] \boldsymbol{y} \\ & + 2 \left[\hat{\boldsymbol{x}}^T \left(\sum_{l=(k,m)\in L\setminus L_r} W_{p_l^f} + \sum_{l=(m,k)\in L\setminus L_r} W_{p_l^t} \right) \right] \boldsymbol{x} \\ & + \left[\hat{\boldsymbol{y}}^T \left(\sum_{l=(k,m)\in L_r} W_{p_l^f}^T + \sum_{l=(m,k)\in L_r} W_{p_l^t}^T \right) \right] \boldsymbol{x} \\ & + \left[\hat{\boldsymbol{x}}^T \left(\sum_{l=(k,m)\in L_r} W_{p_l^f} + \sum_{l=(m,k)\in L_r} W_{p_l^t} \right) \right] \boldsymbol{y} \\ & - \hat{\boldsymbol{x}}^T \left(\sum_{l=(k,m)\in L\setminus L_r} W_{p_l^f} + \sum_{l=(m,k)\in L\setminus L_r} W_{p_l^t} \right) \hat{\boldsymbol{x}} \\ & - \hat{\boldsymbol{x}}^T \left(\sum_{l=(k,m)\in L_r} W_{p_l^f} + \sum_{l=(m,k)\in L_r} W_{p_l^t} \right) \hat{\boldsymbol{y}}. \end{aligned}$$

Similar expressions can be derived for the remaining power flow equations. Note that the left-hand side of this constraint is also a linear expression. Then, the active power flow equations (7h)-(7j) can all be rewritten in the form

$$\boldsymbol{\gamma}_{p,k}^T \boldsymbol{x} + \boldsymbol{\lambda}_{p,k}^T \boldsymbol{y} + \boldsymbol{\psi}_{p,k}^T \boldsymbol{\zeta} + \kappa_{p,k} = 0 \quad \forall k \in N. \tag{13}$$

In a similar matter, we can write the reactive power flow equations (7k) and (7l) as follows:

$$\boldsymbol{\gamma}_{q,k}^T \boldsymbol{x} + \boldsymbol{\lambda}_{q,k}^T \boldsymbol{y} + \boldsymbol{\psi}_{q,k}^T \boldsymbol{\zeta} + \kappa_{q,k} = 0 \quad \forall k \in N. \tag{14}$$

Note that the specifications for $\boldsymbol{\gamma}, \boldsymbol{\lambda}$, $\boldsymbol{\psi}$ and κ differ depending on which bus they apply to, and whether they concern the active or reactive power flow equations.

Now, we also have to deal with the nonlinear equality constraint (7v), which is $v_k^M = (v_k^r)^2 + (v_k^j)^2$

for all $k \in N \setminus \{1\}$. This is a quadratic equality constraint containing state variables. Therefore, we also need to take the first-order Taylor Series of this expression. After linearizing the right side of (7v), we end up with the following constraint:

$$v_k^M = 2\hat{v}_k^r v_k^r + 2\hat{v}_k^j v_k^j - (\hat{v}_k^r)^2 - (\hat{v}_k^j)^2 \quad \forall k \in N \setminus \{1\},$$

which can be added to our linear system of equalities. The right-hand side of this expression contains only state variables, but v_k^M can be a control variable as well. Therefore, we rewrite this as follows:

$$\boldsymbol{\gamma}_{v,k}^T \boldsymbol{x} + \boldsymbol{\lambda}_{v,k}^T \boldsymbol{y} + \kappa_{v,k} = 0 \quad \forall k \in N \setminus \{1\}. \tag{15}$$

Finally, we can write (13)-(15) in matrix form:

$$\Gamma \boldsymbol{x} + \Lambda \boldsymbol{y} + \Psi \boldsymbol{\zeta} + \boldsymbol{\kappa} = \boldsymbol{0}. \tag{16}$$

Note that there is one equality constraint from the original problem (7) that is not contained in (16): $v_1^M = (v_1^r)^2$. This is a non-convex constraint including two control variables. We discuss how to deal with this constraint in Section 5.10.

5.7 Eliminating the state variables

Now that we have a linear expression in all state variables, if we have a certain point of approximation $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$, we can isolate the vector $\boldsymbol{x}$ as follows. From (16) it holds that $\boldsymbol{x} = -\Gamma^{-1}[\Lambda \boldsymbol{y} + \Psi \boldsymbol{\zeta} + \boldsymbol{\kappa}]$. If we denote $A := -\Gamma^{-1}\Lambda$, $B := -\Gamma^{-1}\Psi$ and $\boldsymbol{c} := -\Gamma^{-1}\boldsymbol{\kappa}$, we obtain the following form:

$$\boldsymbol{x} = A\boldsymbol{y} + B\boldsymbol{\zeta} + \boldsymbol{c}. \tag{17}$$

Note that Γ is indeed a square matrix, since its number of rows is the number of constraints in (13)-(15), which is $3|N| - 1$. The number of columns is equal to the dimension of $\boldsymbol{x}$, which is also $3|N| - 1$, as can be seen in (8). We assume that Γ is full rank so that it is invertible. This is a reasonable assumption, because no cross-products between the same two state variables occur in different constraints. Now, what we want to do with expression (17), is to substitute it in every inequality constraint that contains state variables. Those constraints can be summarized in a vector $\boldsymbol{g}$, for which it holds that $\boldsymbol{g}_i \geq 0$ for every element i. We denote the dimension of $\boldsymbol{g}$ by m_{in}; this

denotes the number of inequality constraints involving state variables. For now, we disregard the line thermal limits (7s)-(7t), so that all our inequality constraints are linear in $\boldsymbol{x}$. They can be added to the model later as they are not always needed. We discuss this in Section 5.12. As a result, $\boldsymbol{g}$ is as follows:

$$\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} (v_k^{\max})^2 - v_k^M & \forall k \in N \setminus G \\ v_k^M - (v_k^{\min})^2 & \forall k \in N \setminus G \\ t - p_1^g & \\ p_1^g - p_1^{\min} & \\ q_k^{\max} - q_k^g & \forall k \in G \\ q_k^g - q_k^{\min} & \forall k \in G \end{bmatrix} \geq \mathbf{0}. \tag{18}$$

Now, since we have an explicit linearized expression for the vector of state variables given by (17), we can substitute every state variable in $\boldsymbol{g}$ by a linear expression defined in (17). This way, we eliminate all state variables in the system, and we have an explicit way of describing the full linearized ACOPF problem with uncertainty, in terms of $\boldsymbol{y}$ and $\boldsymbol{\zeta}$ only. Before we explain how to eliminate uncertainty from the problem, we explain how we include in our model the fact that we search in a small space around our point of approximation.

5.8 Searching in a small subset of the feasible space

Now, we need to make sure that the solution to the linearized problem is close to our point of approximation $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$. This is because the approximations closely resemble the original system of equalities around this point. For points farther away from the point of approximation, the approximations become worse. How to do this in the best way, is not exactly obvious. Kuryatnikova et al. (2021) do this by restricting the state variables $\boldsymbol{x}$ to be close to $\hat{\boldsymbol{x}}$, but no explicit restrictions for searching in a small space around $\hat{\boldsymbol{y}}$ are included in the model. As they mention, their method is therefore most efficient if all equality constraints are separable in $\boldsymbol{x}$ and $\boldsymbol{y}$. The ACOPF problem as we state it contains mostly products of state variables. Only for branches connected to the reference bus, the system of equalities contains products of $\boldsymbol{x}$ and $\boldsymbol{y}$. Since the vast majority of all equality constraints are separable in $\boldsymbol{x}$ and $\boldsymbol{y}$, we choose to follow Kuryatnikova et al. (2021) and only restrict the feasible space of the state variables. The small area around $\hat{\boldsymbol{x}}$ in which we search for a better solution, can be described by $||\boldsymbol{x} - \hat{\boldsymbol{x}}|| \leq \varepsilon$, for which a suitable norm and value

for ε must be chosen. For now we choose to use the Euclidean norm. Using this norm, we get $||\boldsymbol{x} - \hat{\boldsymbol{x}}||_2 \leq \varepsilon$. Of course, the state variables need to be eliminated to include these constraints in our model. Here, we choose to substitute $\boldsymbol{x}$ by its nominal linear expression $\boldsymbol{x} = A\boldsymbol{y} + \boldsymbol{c}$, so we get $||A\boldsymbol{y} + \boldsymbol{c} - \hat{\boldsymbol{x}}||_2 \leq \varepsilon$. This is equivalent to the constraint

$$\boldsymbol{y}^T A^T A\boldsymbol{y} + 2(\boldsymbol{c} - \hat{\boldsymbol{x}})^T A\boldsymbol{y} + (\boldsymbol{c} - \hat{\boldsymbol{x}})^T(\boldsymbol{c} - \hat{\boldsymbol{x}}) \leq \varepsilon^2. \tag{19}$$

This is a convex quadratic constraint and can easily be added to the problem. One can of course consider to include $\boldsymbol{\zeta}$ in the substitution of $\boldsymbol{x}$. However, the constraint then turns into an SDP constraint when $\boldsymbol{\zeta}$ is eliminated from the problem, which is what we try to avoid as much as possible for computational efficiency.

Instead of the Euclidean norm, other norms can be chosen as well, such as the infinity norm. Appendix A.1 discusses how to include the latter in the model of the approximated problem.

5.9 Eliminating uncertainty

The next step is to eliminate the uncertainty in the problem so that only the control variables are left. We assume uncertainty comes from an ellipsoidal set (6), with Σ a positive definite matrix. This can be rewritten as $\Omega = \{\boldsymbol{\zeta} \in \mathbb{R}^{|U|} : ||\Sigma^{-\frac{1}{2}}\boldsymbol{\zeta}||_2 \leq r\}$. If we define $M := \begin{bmatrix} \Sigma^{-\frac{1}{2}} \\ \\ \mathbf{0}^T \end{bmatrix}$ and $\boldsymbol{m} := \begin{bmatrix} \mathbf{0} \\ \\ r \end{bmatrix}$, we can also rewrite the uncertainty set as a conic set as follows:

$$\Omega = \{\boldsymbol{\zeta} \in \mathbb{R}^{|U|} : M\boldsymbol{\zeta} + \boldsymbol{m} \in \mathfrak{L}^{|U|+1}\}, \tag{20}$$

where $\mathfrak{L}$ is the second-order cone.

Since we do not consider the line thermal limits for now, the function $\boldsymbol{g}_i \geq 0$ in (18) is a linear function in both $\boldsymbol{\zeta}$ and $\boldsymbol{y}$, and can be written in the following general form:

$$\boldsymbol{g}_i(\boldsymbol{y}, \boldsymbol{\zeta}) = (\boldsymbol{a^i})^T\boldsymbol{y} + (\boldsymbol{b^i})^T\boldsymbol{\zeta} + c^i \geq 0 \quad \forall \boldsymbol{\zeta} \in \Omega,\ i = 1, \dots m_{in}.$$

Note that coefficients $\boldsymbol{a^i}, \boldsymbol{b^i}$ and c^i are not equal to the rows of A, B and $\boldsymbol{c}$ in expression (17), but they can be derived easily from (17) and (18). By conic duality this constraint is equivalent to there

existing a certain $\boldsymbol{z}$, $\boldsymbol{y}$ for which:

$$\begin{aligned} &- \boldsymbol{m}^T \boldsymbol{z^i} + (\boldsymbol{a^i})^T \boldsymbol{y} + c^i \geq 0 \\ &M^T \boldsymbol{z^i} = \boldsymbol{b^i} \\ &\boldsymbol{z^i} \in \mathfrak{L}^{|U|+1} \end{aligned} \tag{21}$$

for all $i = 1, \ldots, m_{in}$. The proof for this is given in Appendix B. For more details on conic duality, see Ben-Tal et al. (2009). Of course, we also need to consider the domain of the control variables, which is summarized as follows:

$$\boldsymbol{h}(\boldsymbol{y}) = \begin{bmatrix} p_k^{\max} - p_k^g & \forall k \in G \setminus \{1\} \\ p_k^g - p_k^{\min} & \forall k \in G \setminus \{1\} \\ p_1^{\max} - t & \\ (v_k^{\max})^2 - v_k^M & \forall k \in G \\ v_k^M - (v_k^{\min})^2 & \forall k \in G \\ v_1^r & \end{bmatrix} \geq \mathbf{0}. \tag{22}$$

This corresponds to constraints (7b)-(7e) in model (7). At this point, we have eliminated the state variables and the uncertainty from the problem. Before we state the full reformulated problem, we first need to discuss how to deal with constraint (7g): $v_1^M = (v_1^r)^2$.

5.10 Dealing with the voltage at the reference bus

We need to replace the non-convex constraint $v_1^M = (v_1^r)^2$ with some convex inequalities. Therefore, we add the following constraints:

$$\begin{aligned} &v_1^M \geq (v_1^r)^2 \\ &v_1^r \geq v_1^{\min} \\ &v_1^r \geq \frac{v_1^M + v_1^{\min} v_1^{\max}}{v_1^{\min} + v_1^{\max}}. \end{aligned} \tag{23}$$

First of all, we can add $v_1^M \geq (v_1^r)^2$ since this is convex. The second and third constraint in (23) are modeled to push $(v_1^r)^2$ up towards v_1^M. The second constraint holds because we restricted v_1^r to be positive. The third expression follows from the McCormick envelopes (McCormick (1976)) of $v_1^M = (v_1^r)^2$. Especially the third constraint in (23) provides a good bound if the range of the voltage magnitude is not too wide. In almost every case we consider, the range of voltage magnitudes is not wider than 0.9 to 1.1, so these constraints indeed provide good bounds. Hence, in our results, constraints (23) are sufficient to push $(v_1^r)^2$ to the exact same value as v_1^M. However, if one encounters a problem with wider voltage magnitude ranges and the previous constraints are not sufficient, one can also implement one of the two options described in Appendix A.2.

5.11 Stating the reformulated problem

In Sections 5.3 until 5.10, we have linearized the power flow equations and eliminated the state variables and the uncertainty from the problem. This results in constraints (19) and (21)-(23). The problem that is left can then be formulated as:

$$\min \quad \sum_{k \in G \setminus \{1\}} \left[c_{k,2}(p_k^g)^2 + c_{k,1} p_k^g + c_{k,0} \right] + c_{1,2} t^2 + c_{1,1} t + c_{1,0} \tag{24a}$$

$$\text{s.t.} \quad -\boldsymbol{m}^T \boldsymbol{z^i} + (\boldsymbol{a^i})^T \boldsymbol{y} + c^i \geq 0 \qquad i = 1, \dots m_{in} \tag{24b}$$

$$M^T \boldsymbol{z^i} = \boldsymbol{b^i} \qquad i = 1, \dots m_{in} \tag{24c}$$

$$\boldsymbol{z^i} \in \mathfrak{L}^{|U|+1} \qquad i = 1, \dots m_{in} \tag{24d}$$

$$\boldsymbol{h}(\boldsymbol{y}) \geq \boldsymbol{0} \tag{24e}$$

$$\boldsymbol{y}^T A^T A \boldsymbol{y} + 2(\boldsymbol{c} - \hat{\boldsymbol{x}})^T A \boldsymbol{y} + (\boldsymbol{c} - \hat{\boldsymbol{x}})^T (\boldsymbol{c} - \hat{\boldsymbol{x}}) \leq \varepsilon^2 \tag{24f}$$

$$v_1^M \geq (v_1^r)^2 \tag{24g}$$

$$v_1^r \geq v_1^{\min} \tag{24h}$$

$$v_1^r \geq \frac{v_1^M + v_1^{\min} v_1^{\max}}{v_1^{\min} + v_1^{\max}}. \tag{24i}$$

Problem (24) is a second-order conic programming problem. Note that problem (24) is solved for each point of approximation we consider. The coefficients of problem (24) are different at every iteration of our algorithm (see Section 5.13 for the pseudocode), due to the fact that the point of approximation $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ differs per iteration.

5.12 Including the line thermal limits

Up until now, we have excluded the constraints on the line thermal limits (7s)-(7t). In many instances this is fine, since the line flow variables $p_l^f, q_l^f, p_l^t, q_l^t$ in these constraints depend highly on the voltages, which themselves are already strictly constrained by the voltage limits. However, it might still turn out that the solution to the approximated problem results in a power flow which is infeasible due to violations of the line thermal limits. In that case, it could help to add those constraints to the problem. When this is necessary, we replace $p_l^f, q_l^f, p_l^t, q_l^t$ by their Taylor Series approximations $T_{p_l^f}, T_{q_l^f}, T_{p_l^t}, T_{q_l^t}$, so we obtain $(T_{p_l^f})^2+(T_{q_l^f})^2 \leq (s_l^{\max})^2$ and $(T_{p_l^t})^2+(T_{q_l^t})^2 \leq (s_l^{\max})^2$ for every $l \in L$ for which we want to model the line thermal constraints. Each Taylor Series approximation can be formulated as a linear expression in $\boldsymbol{x}$ and $\boldsymbol{y}$. Substituting $\boldsymbol{x}$ by its linear form (17), the Taylor Series can be expressed linearly in $\boldsymbol{y}$ and $\boldsymbol{\zeta}$. Squaring these expressions and adding them up, we obtain a quadratic form. For a given branch l, this quadratic form can generally be written as follows:

$$\boldsymbol{g}_l(\boldsymbol{y}, \boldsymbol{\zeta}) = \boldsymbol{y}^T A_l \boldsymbol{y} + \boldsymbol{\zeta}^T B_l \boldsymbol{\zeta} + \boldsymbol{y}^T C_l \boldsymbol{\zeta} + (\boldsymbol{a_l})^T \boldsymbol{y} + (\boldsymbol{b_l})^T \boldsymbol{\zeta} + c_l - (s_l^{\max})^2 \leq 0 \quad \forall \boldsymbol{\zeta} \in \Omega. \tag{25}$$

That is, both $(T_{p_l^f})^2 + (T_{q_l^f})^2 \leq (s_l^{\max})^2$ and $(T_{p_l^t})^2 + (T_{q_l^t})^2 \leq (s_l^{\max})^2$ can be written in the general quadratic form (25). To improve readability, we leave the subscript f or t out of the derivations below, but it must be kept in mind that constraint (25) should be modeled for both the $from$-end and the *to*-end of a branch. To see what the coefficients in constraints (25) exactly are, we refer to Appendix C, in which full derivations for these are given. Now, because this is a quadratic expression in $\boldsymbol{\zeta}$, and the uncertainty set Ω can be written as a quadratic constraint too, the well-known S-Lemma can be used to obtain an equivalent expression for (25), in which the uncertainty is eliminated.

Theorem 1 (S-Lemma (Yakubovich (1971))). *Let $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ be quadratic polynomials in $\boldsymbol{x}$ and suppose there exists an $\boldsymbol{x} \in \mathbb{R}^n$ such that $f(\boldsymbol{x}) < 0$. Then the following two statements are equivalent:*

(1) $g(\boldsymbol{x}) \leq 0$ *for all* $\boldsymbol{x} \in \mathbb{R}^n$ *such that* $f(\boldsymbol{x}) \leq 0$

(2) $\exists \mu \geq 0$ *such that* $g(\boldsymbol{x}) - \mu f(\boldsymbol{x}) \leq 0$ *for all* $\boldsymbol{x} \in \mathbb{R}^n$.

To use S-Lemma, we write $\Omega = \{\boldsymbol{\zeta} \in \mathbb{R}^{|U|} : \boldsymbol{\zeta}^T \Sigma^{-1} \boldsymbol{\zeta} - r^2 \leq 0\}$, which is equivalent to uncertainty

set (6). However, we see now that the uncertainty set can be described as the domain of $\boldsymbol{\zeta}$ for which a certain quadratic function is less than or equal to zero. Since $\boldsymbol{g}_l(\boldsymbol{y}, \boldsymbol{\zeta})$ is also a quadratic function of $\boldsymbol{\zeta}$, we can use S-Lemma. Via S-Lemma, we know that the following holds:

For $\boldsymbol{\zeta} = \mathbf{0}$, it holds that $\boldsymbol{\zeta}^T \Sigma^{-1} \boldsymbol{\zeta} - r^2 < 0$. Then, the following two statements are equivalent:

$$\begin{aligned} &(1) \quad \boldsymbol{g}_l(\boldsymbol{y}, \boldsymbol{\zeta}) \leq 0 \text{ for all } \boldsymbol{\zeta} \in \mathbb{R}^{|U|} \text{ such that } \boldsymbol{\zeta}^T \Sigma^{-1} \boldsymbol{\zeta} - r^2 \leq 0 \\ &(2) \quad \exists \mu_l \geq 0 \text{ such that } \boldsymbol{g}_l(\boldsymbol{y}, \boldsymbol{\zeta}) - \mu_l(\boldsymbol{\zeta}^T \Sigma^{-1} \boldsymbol{\zeta} - r^2) \leq 0 \text{ for all } \boldsymbol{\zeta} \in \mathbb{R}^{|U|}. \end{aligned} \tag{26}$$

That is, if we want the first statement to hold (which is our goal here), it is sufficient to make sure the second statement holds. From the second statement in (26), we can derive an equivalent SDP constraint, that can be stated as follows:

$$\begin{bmatrix} \mu_l \Sigma^{-1} - B_l & -\frac{1}{2}\left((C_l)^T \boldsymbol{y} + \boldsymbol{b_l}\right) \\ -\frac{1}{2}\left(\boldsymbol{y}^T C_l + (\boldsymbol{b_l})^T\right) & (s_l^{\max})^2 - (\boldsymbol{a_l})^T \boldsymbol{y} - c_l - \mu_l r^2 - \nu_l \end{bmatrix} \succeq 0, \quad \mu_l \geq 0, \quad \nu_l = \boldsymbol{y}^T A_l \boldsymbol{y}. \tag{27}$$

The proof that (27) is equivalent to the second statement in (26), is given in Appendix D. The problem of constraints (27) is that we cannot add the constraint $\nu_l = \boldsymbol{y}^T A_l \boldsymbol{y}$, since it is not a convex constraint. However, if $\nu_l \geq \boldsymbol{y}^T A_l \boldsymbol{y}$, the following holds:

$$(s_l^{\max})^2 - (\boldsymbol{a_l})^T \boldsymbol{y} - c_l - \mu_l r^2 - \nu_l \geq 0 \Rightarrow (s_l^{\max})^2 - (\boldsymbol{a_l})^T \boldsymbol{y} - c_l - \mu_l r^2 - \boldsymbol{y}^T A_l \boldsymbol{y} \geq 0.$$

Therefore, the constraint $\nu_l \geq \boldsymbol{y}^T A_l \boldsymbol{y}$ is sufficient to enforce the matrix in (27) to be positive semidefinite. Moreover, the constraint $\nu_l \geq \boldsymbol{y}^T A_l \boldsymbol{y}$ is convex, because A_l is a positive semidefinite matrix (it is the sum of two outer products of a vector with itself (see Appendix C)). Therefore, we replace constraints (27) by the following constraints:

$$\begin{bmatrix} \mu_l \Sigma^{-1} - B_l & -\frac{1}{2}\left((C_l)^T \boldsymbol{y} + \boldsymbol{b_l}\right) \\ -\frac{1}{2}\left(\boldsymbol{y}^T C_l + (\boldsymbol{b_l})^T\right) & (s_l^{\max})^2 - (\boldsymbol{a_l})^T \boldsymbol{y} - c_l - \mu_l r^2 - \nu_l \end{bmatrix} \succeq 0, \quad \mu_l \geq 0, \quad \nu_l \geq \boldsymbol{y}^T A_l \boldsymbol{y}. \tag{28}$$

Constraints (28) are added to problem (24) for every branch l for which one wants to explicitly model the line thermal limits in the approximated problem. This can be all branches $l \in L$, but to reduce the computational burden, one can also investigate which branches cause the most trouble

and add constraints (28) only for those branches. Note that constraints (28) need to be added for both the $from$-end and the to-end of the branch.

5.13 Pseudocode of the algorithm

Finally, we have discussed all relevant steps to solve the ACOPF problem with uncertainty. Here, a pseudocode of the full algorithm is given.

Algorithm 1

Input: Initial solution $(\hat{\boldsymbol{x}}_{\mathbf{1}}, \hat{\boldsymbol{y}}_{\mathbf{1}})$, $f_0 = 10^{20}$, $f_1 = 10^{19}$, $\delta = 10^{-5}$, a value for ε', uncertainty factor ω, radius r, variance-covariance matrix Σ, set of buses with uncertainty U, set of branches L' for which line thermal constraints need to be included in the model.

Output: $\boldsymbol{y}^*$

$i \leftarrow 1$
$\boldsymbol{y}^* \leftarrow \hat{\boldsymbol{y}}_{\mathbf{1}}$
$f^* \leftarrow \infty$
while $f_i + \delta < f_{i-1}$ **do**
    $i \leftarrow i + 1$
    $\varepsilon \leftarrow \sqrt{||\hat{\boldsymbol{x}}_{\boldsymbol{i-1}}||_2/\varepsilon'}$
    Determine $\Gamma, \Lambda, \Psi, \boldsymbol{\kappa}$ in expression (16) via linear approximations.
    Determine $A, B, \boldsymbol{c}$ in expression (17).
    Add constraints (28) to problem (24) for both the $from$-end and to-end of branches in L'.
    Determine the coefficients of problem (24).
    Solve the approximated problem (24).
    if problem (24) is feasible **then**
        Obtain its solution $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$.
        Obtain its objective value f_i.
        if $f_i > f_{i-1}$ **then**
            break
        end if
        Obtain the solution $\hat{\boldsymbol{x}}_{\boldsymbol{i}}$ to the power flow equations with $\boldsymbol{y} = \hat{\boldsymbol{y}}_{\boldsymbol{i}}$ by using 'runpf'.
        if $\hat{\boldsymbol{x}}_{\boldsymbol{i}}$ exists and $(\hat{\boldsymbol{x}}_{\boldsymbol{i}}, \hat{\boldsymbol{y}}_{\boldsymbol{i}})$ is feasible for the operational constraints (1h)-(1l) **then**
            $\boldsymbol{y}^* \leftarrow \hat{\boldsymbol{y}}_{\boldsymbol{i}}$
            $f^* \leftarrow f_i$
        else
            $f_i \leftarrow \infty$
        end if
    else
        $f_i \leftarrow \infty$
    end if
end while

Note that in Algorithm 1, we denote by L' the set of branches for which we include the line thermal constraints (28). Since we only include those constraints when it is necessary, this set can be empty

as well. This is often the case in our results.

If Algorithm 1 finds a solution $\boldsymbol{y^*}$ with $f^* \neq \infty$, it is guaranteed to be feasible for the nominal ACOPF problem. That is, when we substitute $\boldsymbol{y} = \boldsymbol{y^*}$ in problem (1), the solution for the power flow equations is feasible. The solution $\boldsymbol{y^*}$ is also robustly feasible to the approximated problem that is solved in the last iteration. If Algorithm 1 stops with $f^* = \infty$, either it could not find a feasible solution to the approximated problem, or it found a solution to the approximated problem that was not feasible for the nominal problem.

Proposition 1. *Algorithm 1 stops after a finite number of iterations.*

Proof. Note that while the problem changes every iteration, the objective function does not. There exists a lower bound on this quadratic (or linear) objective function, since all costs $c_{k,2}, c_{k,1}$ and $c_{k,0}$ are real numbers and variables p_k^g (and t) are defined on a closed interval. Every iteration of Algorithm 1, the objective value decreases with more than δ, otherwise we stop. Since $\delta > 0$, we reach the lower bound of the objective function in a finite number of iterations.

Now, the choice for ε' in Algorithm 1 is not trivial. The value of ε' is a reflection of the trade-off between finer approximations and a larger searching space for a robust solution. The latter also indicates that we do a local search, which is a limitation of Algorithm 1. If, for example, the solution with the best possible objective value is far away from the initial solution, Algorithm 1 cannot guarantee to find the solution with the best objective value.

6 Data

To evaluate the performance of Algorithm 1, we use data from MATPOWER (Zimmerman et al. (2010)). MATPOWER contains a wide variety of instances that represent power systems, ranging from instances with 3 buses up to instances with more than 10,000 buses. To have a sufficient number of test cases for our solution method, we choose to use nine different MATPOWER instances. These instances are: 'case6ww', 'case9', 'case14', 'case30', 'case39', 'case57', 'case118', 'case300' and 'case1354pegase (adjusted)'. The number contained in the name of each instance denotes the number of buses in that instance. We adjusted 'case1354pegase' by removing the line thermal limits from that instance, because the solution that we obtained violated those limits. We could have included the line thermal constraints (28), but for this instance the runtime would be too high.

Table 4: Summary statistics of nine MATPOWER instances.

Instance	Buses	Nonzero loads	Generators	Branches	Thermal limits	Nom. cost ($/hr)
case6ww	6	3	3	11	Yes	3,144
case9	9	3	3	6	Yes	5,297
case14	14	11	5	20	No	8,080
case30	30	20	6	41	Yes	577
case39	39	21	10	46	Yes	41,869
case57	57	42	7	80	No	41,738
case118	118	99	54	186	No	129,640
case300	300	199	69	411	No	719,725
case1354pegase (adj.)	1354	673	260	1991	No	74,015

In Table 4, a summary of the nine instances is given. We report the number of buses, nonzero loads (which equals the dimension of the uncertainty), generators and branches. Furthermore, we indicate whether an instance has line thermal limits or not. Four out of nine instances have thermal limits. Finally, we report the operational cost of the nominal solution in dollars per hour. This value is found using the option 'runopf' in MATPOWER.

7 Results of computing robustly feasible solutions

7.1 Results for all instances

In this section, we show the results of Algorithm 1 for the nine different MATPOWER instances. We implement our algorithm in MATLAB R2021b, on an Intel® Core™ i5-10500 CPU @ 3.10GHz with 16 GB RAM. To solve the approximated problem (24), we use CVX, version 2.2 (Grant and Boyd (2008), Grant and Boyd (2014)). We use the default solver of CVX, which is SDPT3, version 4 (Toh et al. (1999), Tütüncü et al. (2003)). We run Algorithm 1 with $r = 1.645$ in the uncertainty set (6), as mentioned in Section 4. We let $\epsilon' = 10$ for instances up to 100 buses and $\epsilon' = 100$ for larger instances, since the Euclidean norm of $\hat{\boldsymbol{x}}$ gets bigger for larger instances. We shrink the bounds of each MATPOWER case by 0.5% for all variables. This includes shrinking the line thermal limits. Using the 'runopf' option in MATPOWER, we find the initial solution $(\hat{\boldsymbol{x}}_{\boldsymbol{1}}, \hat{\boldsymbol{y}}_{\boldsymbol{1}})$ to the problem with these shrunk bounds. We do this to obtain an already somewhat robust solution. We also use these shrunk bounds in our approximated problem, to avoid problems with binding inequality constraints in the approximated problem. That is, if a solution is found based on inequality constraints that are binding, but our approximation was slightly off, then it might be that in the original problem

the solution slightly violates the bound of that constraint for certain values of uncertainty. By shrinking the bounds we try to avoid such an issue. For checking if an obtained solution is feasible for the original problem, naturally we use the original bounds. For some cases, an initial solution to the problem is not possible when the bounds are shrunk. We report the instances for which this is the case, and use the original bounds there. Finally, for most instances we do not include the constraints on the line thermal limits in the approximated problem. Whenever the obtained solution violates these limits, we include them. Again, we report whenever this is the case.

In Table 5, for each considered instance the results of running Algorithm 1 are given for different levels of uncertainty. In all cases, we start with 1% uncertainty in the load (corresponding to $\omega = 0.01$ in the definition of Σ given in Section 4), and increase the level of uncertainty to a maximum of 50% of the load. In many cases this level of uncertainty is infeasible for the approximated problem; for those cases the results are presented up to the maximum level of uncertainty that is still feasible. Only for 'case300', the minimum level of 1% uncertainty is not feasible, so for that instance we report the results for 0.1% uncertainty. The results in Table 5 include the objective value when our robust solution $\boldsymbol{y}^*$ is substituted in the nominal ACOPF problem and the power flow equations are solved. It also includes the percentage increase in objective value, compared to the nominal solution given in Table 4. This indicates the difference in cost in the situation of no uncertainty. Furthermore, we give the runtime per iteration and the number of iterations. The runtime of one iteration is the time Algorithm 1 spends in one while loop. Finally, we give results with and without correlation between uncertainty in loads.

Before discussing the main results, it is relevant to list the instances for which either the bounds could not be used shrunk, or in which the line thermal constraints turned out to be necessary to include:

- 'case6ww': Line thermal constraints are necessary, shrinking bounds is not possible.
- 'case30': Line thermal constraints are necessary.
- 'case57': Shrinking bounds by 0.5% is not possible; bounds are shrunk by 0.1%.

The main findings of running Algorithm 1 are listed below.

Main results of running Algorithm 1:

- The maximum amount of uncertainty varies a lot between instances. For the larger instances, only small amounts of uncertainty (no more than 5% of the load) are feasible, but there are also some smaller instances for which this is the case.
- For larger instances the runtime drastically increases when Σ is considered with correlation compared to no correlation.
- The operational cost increase for our approximately robust solution compared to the nominal solution is marginal. Most cost increases are below 0.1%, and the maximum increase in cost is 0.71%.
- For most instances, the objective value for the nominal problem stays relatively the same or increases only slightly throughout different levels of uncertainty. This means that the cost for setting up a more (approximately) robust solution does not increase that much in the situation of no uncertainty.
- The instances for which the line thermal constraints are included, both allow only 1% uncertainty.

Now we discuss the results in Table 5 in more detail. We first notice that the maximum amount of uncertainty for which our algorithm could find a solution to the approximated problem, varies a lot throughout the instances. For 'case9', 'case14' and 'case39', 10% uncertainty or more is feasible, but for the other six cases, approximately robust feasibility can only be achieved at lower levels of uncertainty. Those six instances include the four largest cases, but also two smaller cases: 'case6ww' and 'case30'. These are the same two cases in which line thermal limits need to be enforced. We think that the little flexibility of these two instances causes both the robust infeasibility for even small amounts of uncertainty, as well as the need for line themal constraints. We discuss these cases more in-depth in Section 7.2.

Now, to discuss some results that occur within specific cases, we take a look at 'case9' (without correlation) first, since it has results for a lot of different uncertainty levels. It can be seen that when uncertainty increases, the objective value of our solution in the situation of no uncertainty seems to stay relatively similar. This means that more approximately robust solutions do not necessarily result in much higher costs for the nominal ACOPF problem. This can be an important

Table 5: Results of Algorithm 1 for all nine instances, with and without correlation.

	Without correlation				With correlation			
Uncertainty (% of load)	Obj. val. ($/hr)	Diff. with nom. (%)	Time per iter. (sec.)	Number of iterations	Obj. val. ($/hr)	Diff. with nom. (%)	Time per iter. (sec.)	Number of iterations
case6ww								
1%	3,153	0.29	1.5	2	3,153	0.29	1.3	2
case9								
1%	5,298	0.02	0.6	2	5,298	0.02	0.6	2
5%	5,298	0.02	0.6	2	5,298	0.02	0.6	2
10%	5,298	0.02	0.6	2	5,298	0.02	0.6	2
20%	5,309	0.23	0.6	5	5,298	0.02	0.6	2
30%	5,299	0.04	0.6	2	5,325	0.53	0.6	3
40%	5,300	0.06	0.6	2	5,299	0.04	0.6	2
50%	5,301	0.08	0.6	2	5,303	0.12	0.6	2
case14								
1%	8,096	0.20	0.7	3	8,088	0.10	0.8	4
5%	8,087	0.09	0.7	4	8,087	0.09	0.8	4
10%	8,086	0.07	0.7	4	8,086	0.07	0.8	4
case30								
1%	581	0.71	7.2	2	581[†]	0.71	8.5	2
case39								
1%	41,898	0.07	1.1	4	41,898	0.07	1.3	4
5%	41,900	0.07	1.1	5	41,900	0.07	1.2	5
10%	41,904	0.08	1.1	5	41,902	0.08	1.2	3
20%	41,933[†]	0.15	1.1	2	41,920	0.12	1.3	2
case57[*]								
1%	41,756	0.04	1.4	2	41,753	0.04	9.5	3
5%	41,758	0.05	1.4	3	41,758	0.05	9.4	3
case118								
1%	129,699	0.05	3.4	8	129,699	0.05	38.8	8
5%	129,723	0.06	3.4	8	129,723	0.06	38.0	8
case300[*]								
0.1%	723,042	0.46	16.7	2	723,028[†]	0.46	588.0	3
case1354pegase (adjusted)								
1%	74,032	0.02	1084.9	1[‡]	-	-	-	-

[*] Tap ratios are unchanged for 'case57' and 'case300', otherwise a nominal solution could not be obtained. For all other instances tap ratios are set equal to one, which is equivalent to excluding them from the model.
[†] Solution is obtained by continuing with 'Inaccurate/Solved' status of CVX. This comes down to constraint (24f) being slightly relaxed by CVX. Otherwise, a solution could not be found.
[‡] Algorithm 1 is stopped after one iteration due to the potentially high computation time of multiple iterations.

factor of consideration when one wants to make a trade-off between operational cost and robustness. From Table 5, we can even see that a lower cost can be obtained for a higher level of uncertainty. For example, the rise from 20% to 30% uncertainty, sees a decrease in objective value: from 5309 to 5299. This seems counterintuitive: if the solution is robust against 30% uncertainty, and it is less costly than the robust solution against 20% uncertainty, it seems subobtimal to ever use the latter solution. The reason this can occur, is because our algorithm finds a solution to an approximated problem. When uncertainty levels differ, the approximated problem differs too. Moreover, when the algorithm runs for multiple iterations, the space in which it searches for a solution changes too.

Now we look more into the time it takes for Algorithm 1 to find a solution. First of all, there is no difference in runtime between different levels of uncertainty. Second, for instances up to 118 buses the algorithm runs very fast, both with and without correlation in uncertainty. However, for 'case300' the difference between correlation and no correlation in terms of computation time becomes really clear. Running 'case300' without correlation takes 16.7 seconds per iteration, while running it with correlation takes 588.0 seconds per iteration: that is more than 35 times longer. Third, the runtime is in general much lower when the line thermal constraints are not added to the model. For 'case6ww' and 'case30', it is necessary to include them. This results in the approximated problem becoming an SDP problem, which is reflected in the computation time. For example, 'case30' without correlation needs 7.2 seconds for an iteration, while 'case39' needs only 1.1 seconds, even though the latter contains more buses with nonzero load. Finally, the largest instance, 'case1354pegase' needs 1084.9 seconds for an iteration, which is more than 18 minutes. It must be said that approximately 75% of this runtime is devoted to CVX setting up the model. Specifically, it takes the most time to model constraint (24d), which comes down to specifying that 2710 vectors belong to the second-order cone with dimension 673. While this is not desirable, it does mean that the runtime can potentially be decreased here: the model construction can be done only once so that only some numbers need to be adjusted for a new iteration or a new uncertainty level. The optimization part, however, needs to fully run every time. The generally higher computation time for the situation in which uncertainty is correlated, together with the high runtime for this instance, is the reason we do not consider 'case1354pegase' with correlation.

7.2 Analysis of 'case6ww' and 'case30'

As stated before, for 'case6ww' and 'case30' Algorithm 1 is not able to find a feasible solution for the approximated problem for any uncertainty level higher than 1% of the load. This is remarkable, because they are relatively small instances. For both these instances it is also necessary to explicitly include the line thermal constraints (28) in the approximated problem, otherwise the obtained solution is not feasible for the nominal ACOPF problem. It could of course be that this is not a coincidence.

First of all, it could be that these approximated line thermal constraints are bad approximations. Of course, the terms $T_{p_l^f}, T_{q_l^f}, T_{p_l^t}, T_{q_l^t}$ contain a certain error, since they are linearized versions of $p_l^f, q_l^f, p_l^t, q_l^t$. By modelling $(T_{p_l^f})^2 + (T_{q_l^f})^2 \leq (s_l^{\max})^2$ and $(T_{p_l^t})^2 + (T_{q_l^t})^2 \leq (s_l^{\max})^2$, the squared terms increase the error, which causes the approximations to be worse. However, if we also include these constraints when solving 'case9' and 'case39' (the other cases do not have any line thermal limits), the algorithm finds a solution, even for the highest uncertainty levels of those cases (for 'case39' at 20% uncertainty, this requires the value for ε' to be changed, see Section 8.3). This indicates that the SDP approximations of the line thermal constraints are not necessarily bad.

It could also be that the underlying factor that leads to the line thermal constraints being necessary to model, is the same driving factor that causes the problem to be infeasible for higher uncertainty levels than 1%. For both 'case30' and 'case6ww', the optimal solution to the nominal ACOPF problem has binding line thermal limits. That is, for some branches, the optimal value of constraints (1j)-(1k) takes on its limit. These are the only instances for which this is the case. If a solution for these cases is pushed towards these limits to achieve feasibility, then the instance at hand has little flexibility, which could be a problem for robust analysis. Of course, binding constraints can also be binding to achieve optimality instead of feasibility, in which case the problem at hand can still be flexible enough. However, in these cases feasibility seems to be the bigger factor. Comparing the average line thermal limit $s_l^{\max}$, expressed in megavolt-ampere (MVA), we get that this is a much lower value for 'case6ww' and 'case30' (48 MVA, 52 MVA), than for 'case9' and 'case39' (233 MVA, 734 MVA).

Lastly, 'case6ww' can also be considered a special case since the voltage magnitudes of the gen-

erator buses are all fixed. Those equality constraints on the voltages can lead to the problem at hand being unsuitable for robust analysis. When designing a power system, one should therefore make sure that there is some inherent robustness to the system. Otherwise, a similar situation might occur in practice, which is undesirable. We note that 'case6ww' is a specially made example case from Wood and Wollenberg (1996) and that real-life power systems often have some level of inherent robustness. If a situation like 'case6ww' does occur in practice, one should be additionally careful and try to prevent mismatches in the system even more than in systems with more flexibility.

8 Performance of solutions under simulated uncertainty

8.1 Simulation setup

In this section, a simulation study is performed to evaluate how our obtained solution performs when uncertainty is realized. Nominal feasibility is ensured by Algorithm 1, but we are of course especially interested in whether the solution is also feasible when uncertainty is realized. For an obtained solution $\boldsymbol{y}^*$ to be feasible after uncertainty is known, a solution to the power flow equations with the realized uncertainty should exist, and this solution should be feasible for the operational inequality constraints as well. Now, we have two different ways of sampling the uncertainty. One of them is sampling only points that are within our ellipsoid Ω defined by (6), which is what we tried to be robust against. To sample points uniformly from the ellipsoid, we use a method proposed by Dezert and Musso (2001). To implement their method, we use the code they provide in their paper. The other method of sampling is to use the variance-covariance matrix Σ and sample using a normal distribution with mean zero and variance-covariance matrix Σ. We use this second method of sampling as well, since this distribution is related to how we constructed Ω in the first place. Now, the sampling method makes a lot of difference, especially when the dimension of $\boldsymbol{\zeta}$ increases. For example, if the dimension of $\boldsymbol{\zeta}$ is 10 (relatively small), then $\boldsymbol{\zeta}^T\Sigma^{-1}\boldsymbol{\zeta} \sim \chi^2_{10}$ if $\boldsymbol{\zeta}$ follows the multivariate normal distribution with variance-covariance matrix Σ. If we use our radius $r = 1.645$, then points generated by this normal distribution are inside our ellipsoid with probability:

$$P(\boldsymbol{\zeta}^T\Sigma^{-1}\boldsymbol{\zeta} \leq r^2) = F_{\chi^2_{10}}(r^2) = F_{\chi^2_{10}}(2.7) = 0.0124.$$

That is, with this relatively small dimension, less than two percent of sampled points lie within the ellipsoid that we wanted to be robust against. However, it is still interesting to see how our method

performs against samples that can be outside of the ellipsoid as well, especially compared to other benchmark methods. For example, in practice it may be that assumptions on uncertainty turn out to be wrong, and realized uncertainty is larger than expected. The assumed ellipsoid can turn out to be too small in that case, and it is good to know how our method performs in such a situation.

In the simulation, we would like to compare the performance of our method with the performance of other methods. However, there are no state-of-the-art methods for solving the ACOPF under uncertainty, and there are no publicly available codes for the approaches mentioned in Section 2.3. It would be a topic for separate research to compare our approach against those. For this reason, we consider the following benchmark methods in this research. The first benchmark method is the nominal solution to the ACOPF problem (with the bounds shrunk by 0.5% as before). The shrunk bounds make this solution more robust already, and it turned out that shrinking the bounds matters a lot for the performance of this method, making it a more competitive benchmark method. The second benchmark method is the DCOPF solution. As mentioned in Section 2, the DCOPF is a linearized version of the ACOPF. It assumes that voltages are constant at 1 p.u., and that voltage angle differences are small (so that the sinus terms in the polar form of the ACOPF can be linearized). However, these assumptions are not very realistic in highly loaded systems, and it turns out that the DCOPF gives a lot of very infeasible results. We therefore only include the DCOPF method in cases where it gives at least some percentage of feasible results. We implement this model using MATLAB, MATPOWER and the code provided by Bienstock et al. (2014).

For each instance, uncertainty level and sample method, we simulate 1000 realizations of $\boldsymbol{\zeta}$. To avoid taking into account purely numerical violations that are not important to the power system, we first round all violations down to 3 decimal places. This is typical in the ACOPF literature, see for example Venzke et al. (2017). We then report the percentage of samples of $\boldsymbol{\zeta}$ for which our obtained solution $\boldsymbol{y^*}$ is feasible. That is, a solution to the power flow equations exists, and this solution is feasible for all inequality constraints. In practice, some small violations of the inequality constraints can be acceptable as well, since a given power system might be able to adjust to it. Therefore we also report the percentage of samples for which the solution is feasible if we allow violations of at most 0.1%, and the percentage of samples for which the solution is feasible if we allow violations of at most 1%. These violations are measured as a percentage of the interval between the lower and upper bound of a variable. To measure the violations of the voltage magnitude

constraints, we use the non-squared constraints (1l). For line thermal limits, we also use the original non-squared constraints (1j)-(1k). We also calculate the average number of violated constraints (over the 1000 samples), the average percentage of violation (over all violated constraints), and the maximum percentage of violation. Lastly, we consider only the situation of no correlation between uncertainties, since the results are rather similar to the situation with correlation (as demonstrated by Table 5).

8.2 Simulation results

Before the results of the simulation are discussed in more detail for each instance, the main findings are listed below.

Main results of the simulation:

- Overall, our approach outperforms the nominal solution and the DCOPF.
- The performance of our method decreases when the uncertainty level gets larger.
- The performance of our method depends on the instance at hand; 'case118' shows the most drastic difference in performance between our solution and the nominal solution, with our solution performing much better.
- Especially for normally distributed samples, it can make a lot of difference regarding feasibility if violations of 0.1% or 1% are allowed or not, for both our solution as well as the nominal solution.

In Table 6, Table 7, Table 8 and Table 9, the results of the simulation are given. Table 6 shows results for 'case6ww' and 'case9', Table 7 treats 'case14' and 'case30', Table 8 discusses 'case39' and 'case57', and Table 9 presents results for 'case118', 'case300' and the adjusted version of 'case1354pegase'. Below, we discuss every table briefly.

Looking at 'case6ww' in Table 6, we see that our approximately robust solution performs very well for both sampling methods, with 100% of the samples being fully feasible for points within the ellipsoid, and 95.1% of samples being fully feasible for the samples from the normal distribution. Our solution clearly outperforms the nominal solution, which has 49.9% and 50.1% of samples being fully feasible, respectively. For 'case9', we present the results in Table 6 starting from an uncertainty

Table 6: Simulation results for 'case6ww' and 'case9'.

Uncertainty (% of load):	Sample: method:	Solution	% feasible	% feasible (0.1% viol.)	% feasible (1% viol.)	Avg. num. of viol. constr.	Avg. viol. (%)	Max. viol. (%)
				case6ww				
1%	Ellipsoid:	Robust	100	100	100	0	0	0
		Nominal	49.9	54.5	92.0	0.50	0.6	1.5
	Normal distr.:	Robust	95.1	96.0	99.7	0.05	0.4	1.7
		Nominal	50.1	53.8	85.1	0.50	0.8	3.2
				case9				
10%	Ellipsoid:	Robust	100	100	100	0	0	0
		Nominal	100	100	100	0	0	0
		DCOPF	100	100	100	0	0	0
	Normal distr.:	Robust	100	100	100	0	0	0
		Nominal	100	100	100	0	0	0
		DCOPF	99.9	99.9	100	0.001	0.5	0.5
20%	Ellipsoid:	Robust	100	100	100	0	0	0
		Nominal	100	100	100	0	0	0
		DCOPF	100	100	100	0	0	0
	Normal distr.:	Robust	98.5	98.5	99.0	0.02	4.8	23.7
		Nominal	96.5	96.5	98.8	0.06	2.0	23.8
		DCOPF	97.2	97.2	97.4	0.03	7.3	29.5
30%	Ellipsoid:	Robust	98.8	98.9	99.4	0.01	1.6	4.6
		Nominal	95.9	95.9	99.4	0.05	0.8	4.3
		DCOPF	-	-	-	-	-	-
	Normal distr.:	Robust	92.2	92.4	92.8	0.08	9.3	37.6
		Nominal	86.3	86.3	91.3	0.29	3.6	37.3
		DCOPF	-	-	-	-	-	-
40%	Ellipsoid:	Robust	91.2	91.2	92.0	0.09	5.9	16.4
		Nominal	86.3	86.3	91.6	0.28	2.4	15.9
		DCOPF	-	-	-	-	-	-
	Normal distr.:	Robust	83.7	83.7	84.4	0.22	13.2	63.5
		Nominal	73.9	73.9	80.2	0.67	5.8	63.1
		DCOPF	-	-	-	-	-	-
50%	Ellipsoid:	Robust	83.9	83.9	85.0	0.16	10.1	25.4
		Nominal	75.5	75.5	82.8	0.60	3.6	24.8
		DCOPF	-	-	-	-	-	-
	Normal distr.:	Robust	75.2	75.3	75.8	0.37	17.4	101.2
		Nominal	63.3	63.3	68.6	1.05	8.1	100.8
		DCOPF	-	-	-	-	-	-

level of 10%. The results for lower uncertainty levels can be summarized by all three methods being fully feasible 100% of the time. The 10% level and 20% give similar results, with all three solutions being feasible for all samples within the ellipsoid, and for almost all normally distributed samples. At the 30% level, the DCOPF cannot find a solution anymore. Also, we see that throughout all uncertainty levels for 'case9', our solution performs better than or similar to the nominal solution. The difference in performance gets larger when the uncertainty level rises. However, average violations are much lower for the nominal solution. For example, at the uncertainty level of 50%, our solution violates a constraint with an average of 10.1% compared to 3.6% for the nominal solution, for samples within the ellipsoid. This could be something to take into account if one rather has many smaller violations (0.60 violated constraints on average for the nominal solution), instead of fewer large violations (0.16 violated constraints on average for our solution). This is a case-specific issue and not inherent to our method, since the opposite situation occurs at other instances.

We now discuss the results of Table 7. For 'case14', the difference in performance between the two methods is very clear. Even at the lowest uncertainty level (1%), the nominal solution is fully feasible for only 46.9% of samples within the ellipsoid. Our solution, on the other hand, is fully feasible for 85.7% of those samples. The average violation is also much higher for the nominal solution than for our solution: 0.8% versus 0.1%. For the samples from the normal distribution, both methods perform slightly worse, but the difference in performance between the two methods is analogous. Results for higher levels of uncertainty show an even larger difference in performance between the two methods. For 'case30', the differences between the two solutions are much smaller. However, our solution still performs slightly better. We obtain full feasibility for 96.0% of samples from the ellipsoid compared to 89.9% for the nominal solution. When considering the other sampling method, we see that these percentages drop to 56.6% and 56.2%, respectively. However, when allowing violations of 1% or lower, we obtain 99.1% feasibility for our solution and 91.9% for the nominal solution.

In Table 8, it can be seen that for most uncertainty levels of 'case39', the differences in results are larger again. However, both methods perform poorly compared to other instances. For the 1% uncertainty level, our solution achieves full feasibility for only 68.7% of the samples in the ellipsoid. The nominal solution has a similar percentage of 67.8% for these samples. For higher uncertainty levels, these percentages get even lower, but the difference in performance between the solutions

Table 7: Simulation results for 'case14' and 'case30'.

Uncertainty (% of load):	Sample: method:	Solution	% feasible	% feasible (0.1% viol.)	% feasible (1% viol.)	Avg. num. of viol. constr.	Avg. viol. (%)	Max. viol. (%)
			case14					
1%	Ellipsoid:	Robust	85.7	92.4	100	0.14	0.1	0.5
		Nominal	46.9	51.0	79.6	0.59	0.8	3.0
	Normal distr.:	Robust	67.1	72.4	97.6	0.36	0.4	2.5
		Nominal	41.0	44.9	65.8	0.75	1.4	7.2
5%	Ellipsoid:	Robust	99.9	100	100	0.001	0.01	0.01
		Nominal	33.6	35.9	51.1	0.91	2.8	16.6
	Normal distr.:	Robust	82.3	83.3	88.3	0.23	3.4	19.9
		Nominal	29.9	30.9	38.0	0.99	6.4	40.5
10%	Ellipsoid:	Robust	100	100	100	0	0	0
		Nominal	32.6	33.3	41.5	0.94	5.7	28.3
	Normal distr.:	Robust	86.4	86.6	88.9	0.18	6.5	40.7
		Nominal	30.9	31.0	35.2	0.98	12.2	71.0
			case30					
1%	Ellipsoid:	Robust	96.0	100	100	0.04	0.03	0.1
		Nominal	89.9	93.0	100	0.10	0.2	0.8
	Normal distr.:	Robust	56.6	70.7	99.1	0.54	0.2	2.0
		Nominal	56.2	61.5	91.9	0.59	0.5	3.4

grows in favour of our solution. For 'case57', results are even more clearly in favour of our solution method. Even at the highest uncertainty level of 5% here, full feasibility is achieved for 100% of samples in the ellipsoid. The nominal solution only achieves this for 25.0% of samples. Even if violations of 1% are allowed, the nominal solution achieves feasibility for only 40.4% of those samples. For the normally distributed samples, our solution is fully feasible for 83.1% of samples, against 24.1% for the nominal solution. Considering that none of those samples lie in the specified ellipsoid, our solution performs very well for 'case57'.

Overall, our solution performs better than the nominal solution in every instance so far. We still need to consider the three largest cases, of which the simulation results are summarized in Table 9. Results for 'case118' are similar to those for 'case57': for both 1% and 5% uncertainty, our solution

Table 8: Simulation results for 'case39' and 'case57'.

Uncertainty (% of load):	Sample: method:	Solution	% feasible	% feasible (0.1% viol.)	% feasible (1% viol.)	Avg. num. of viol. constr.	Avg. viol. (%)	Max. viol. (%)
				case39				
1%	Ellipsoid:	Robust	68.7	72.4	94.6	0.31	0.6	2.2
		Nominal	67.8	72.0	94.6	0.32	0.6	2.2
	Normal distr.:	Robust	56.7	57.9	69.5	0.43	2.1	8.5
		Nominal	39.6	48.6	69.4	0.62	1.5	8.5
5%	Ellipsoid:	Robust	57.5	58.1	63.7	0.43	3.5	13.2
		Nominal	28.9	34.7	61.5	0.77	2.2	13.5
	Normal distr.:	Robust	45.8	46.5	52.8	0.57	9.6	36.4
		Nominal	20.8	22.3	38.2	0.95	6.4	36.7
10%	Ellipsoid:	Robust	55.2	55.5	59.4	0.45	7.2	29.5
		Nominal	23.0	25.6	46.6	0.87	4.4	30.2
	Normal distr.:	Robust	34.0	35.0	39.9	1.05	12.3	97.9
		Nominal	16.2	16.7	24.6	1.43	10.0	98.5
20%	Ellipsoid:	Robust	35.2	36.1	40.8	0.84	9.5	53.5
		Nominal	18.9	19.5	30.8	1.13	7.7	54.9
	Normal distr.:	Robust	11.3	11.7	12.5	2.55	18.7	194.6
		Nominal	4.8	4.9	7.9	2.76	17.7	195.3
				case57				
1%	Ellipsoid:	Robust	100	100	100	0	0	0
		Nominal	28.4	37.9	82.4	0.91	0.6	3.4
	Normal distr.:	Robust	86.0	87.5	92.8	0.14	1.8	11.0
		Nominal	28.5	29.7	46.6	0.95	2.4	16.8
5%	Ellipsoid:	Robust	100	100	100	0	0	0
		Nominal	25.0	26.5	40.4	0.96	2.9	15.3
	Normal distr.:	Robust	83.1	83.3	86.7	0.18	7.0	42.0
		Nominal	24.1	24.5	28.2	1.00	12.2	74.6

Table 9: Simulation results for 'case118', 'case300', 'case1354pegase (adjusted)'.

Uncertainty (% of load):	Sample: method:	Solution	% feasible	% feasible (0.1% viol.)	% feasible (1% viol.)	Avg. num. of viol. constr.	Avg. viol. (%)	Max. viol. (%)
			case118					
1%	Ellipsoid:	Robust	100	100	100	0	0	0
		Nominal	40.7	53.7	99.4	0.82	0.3	1.5
	Normal distr.:	Robust	65.3	68.9	86.6	0.41	1.0	5.3
		Nominal	1.9	2.9	17.4	3.20	1.4	10.1
5%	Ellipsoid:	Robust	100	100	100	0	0	0
		Nominal	2.7	4.7	23.7	2.85	1.2	9.8
	Normal distr.:	Robust	41.5	43.4	54.5	0.84	3.6	22.2
		Nominal	0.2	0.2	0.7	5.23	6.3	49.5
			case300					
0.1%	Ellipsoid:	Robust	100	100	100	0	0	0
		Nominal	46.3	81.7	98.5	0.61	0.1	1.3
	Normal distr.:	Robust	85.3	91.8	97.6	0.16	0.7	9.1
		Nominal	4.7	17.0	58.0	2.59	1.0	20.0
			case1354pegase (adjusted)					
1%	Ellipsoid:	Robust	100	100	100	0	0	0
		Nominal	21.1	32.0	76.3	1.66	0.5	4.9
	Normal distr.:	Robust	6.8	9.4	44.2	2.49	1.3	37.8
		Nominal	0.0	0.0	0.0	25.2	1.9	74.1

is feasible for 100% of the samples within the ellipsoid. The nominal solution, on the other hand, performs really bad for this instance. At the 1% level, it achieves 40.7% full feasibility for samples from the ellipsoid, but at the 5% level it is only feasible for 2.7% of the samples. It must be said that when allowing 1% violations, the nominal solution reaches 99.4% feasibility for 1% uncertainty, but at the 5% level this is still only 23.7%. For the normally distributed samples, our solution achieves 65.3% feasibility and 41.5% feasibility for 1% and 5% uncertainty, respectively. For the nominal solution, these percentages are 1.9% and 0.2%, respectively. Thus far, 'case118' might be the case in which the results for the two methods differ the most, especially at the 5% uncertainty level.

Finally, we discuss the results for the two largest cases. When looking at points within the el-

lipsoid for 'case300', our solution is fully feasible for 100% of samples. This is in contrast to the 46.3% of the nominal solution. For samples of the normal distribution, our solution is feasible for 85.3% of the samples compared to 4.7% for the nominal solution. This pattern persists when allowing violations of 1%. Lastly, we look at the results for the largest instance: the adjusted version of 'case1354pegase'. For the only feasible uncertainty level of 1%, our solution is feasible for 100% of the uncertainty samples within the ellipsoid. The nominal solution is feasible for only 21.1% of the samples. For the normally distributed samples, our solution is fully feasible 6.8% of the time. When violations of 1% are allowed, it is feasible for 44.2% of the samples. The nominal solution is feasible for none of the normally distributed samples, even when 1% violations are allowed. This means that our solution outperforms the nominal solution for both sampling methods at this instance as well.

8.3 Influence of excluding the line thermal constraints

As mentioned before, for 'case6ww' and 'case30', it is necessary to include the line thermal constraints. Otherwise, the solution $\boldsymbol{y}^*$ found by Algorithm 1 results in a power flow for the nominal problem that violates the line thermal limits. For the two other instances with line thermal limits, 'case9' and 'case39', the solution found by Algorithm 1 causes no violation of line thermal limits in the nominal problem, so it is not necessary to include the line thermal constraints in the approximated problem. However, we only checked whether line thermal limits in the nominal problem are violated, but what about when uncertainty is realized? It might be that our solution is less robust than it would be when we explicitly model the line thermal constraints (28). We investigate this by including the line thermal constraints for 'case39'.

In Table 10, the results of the simulation for 'case39' are given. The simulation setup is the same as in the previous section. For each uncertainty level and sampling method, results are presented for the situation without line thermal constraints and for the situation in which line thermal constraints are added to the approximated problem. Before we discuss the results, it must be noted that for the 20% uncertainty level, the results without line thermal constraints slightly differ compared to Table 8. This is because Algorithm 1 cannot find a solution with line thermal constraints for this uncertainty level with the same simulation setup as before. To still make a valid comparison at the 20% level, we increase the searching space by setting $\varepsilon' = 5$, instead of $\varepsilon' = 10$. We only do this for the 20% uncertainty level and this results in only marginal differences compared to Table 8.

Table 10: Simulation results for 'case39', with and without the line thermal constraints.

Uncertainty (% of load):	Sample: method:	With/without therm. constr.	% feasible	% feasible (0.1% viol.)	% feasible (1% viol.)	Avg. num. of viol. constr.	Avg. viol. (%)	Max. viol. (%)
case39								
10%	Ellipsoid:	Without	55.2	55.5	59.4	0.45	7.2	29.5
		With	55.7	56.1	59.5	0.44	7.2	29.5
	Normal distr.:	Without	34.0	35.0	39.9	1.05	12.3	97.9
		With	42.1	42.9	47.7	0.84	14.3	97.8
20%	Ellipsoid:	Without	34.6	35.7	40.6	0.85	9.4	53.4
		With	51.6	51.8	53.6	0.48	14.5	53.7
	Normal distr.:	Without	11.4	11.6	12.5	2.55	18.7	194.6
		With	35.7	35.8	38.2	1.49	23.1	194.7

For uncertainty levels of 1% and 5%, both approaches find the exact same solution. Therefore we exclude these results from Table 10; they can be found in Table 8. For the 10% uncertainty level, the approach with line thermal constraints finds a different solution but performs similarly for the samples within the ellipsoid. However, for normally distributed samples, enforcing line thermal limits results in 42.1% of these samples being fully feasible, instead of 34.0%. At the 20% level, the difference is much clearer. When the line thermal limits are not modeled, our solution is fully feasible for 34.6% of samples in the ellipsoid. However, when we include the line thermal constraints, this percentage rises to 51.6%. For normally distributed samples, this percentage rises from 11.4% to 35.7%.

The results in Table 10 indicate that for lower levels of uncertainty, line thermal constraints do not necessarily have to be modeled, since the solution found by Algorithm 1 is the same. However, for larger uncertainty levels, the obtained solution differs, and it can make a difference in terms of robust feasibility. Whether this difference is large enough to include the line thermal constraints (28) in the problem, depends on the situation. One must keep in mind that including these constraints turns the problem into an SDP problem, which results in a much higher computation time. For example, one iteration of Algorithm 1 for 'case39' takes 1.1 seconds (see Table 5) when line thermal constraints are not considered. When including the thermal constraints, Algorithm 1 takes 10.5

seconds to complete one iteration, which is more than nine times as long. For larger instances than 'case39', the runtime can become too high if the model needs to be solved quickly. It must be said that we model the line thermal constraints for all branches; runtime can be potentially decreased by including only branches that are known to cause problems regarding line thermal limits.

9 Conclusion and Discussion

This thesis deals with the Alternating Current Optimal Power Flow (ACOPF) problem with uncertainty. The purpose of this thesis is to develop an approach for finding a robust solution to this problem that is scalable to large instances, suitable for moderate levels of uncertainty and has an operational cost that is comparable to the cost of the nominal solution. After proposing such an approach, we evaluate the quality of this method in terms of computation time and operational cost. Finally, we evaluate the performance of our solution method in terms of robust feasibility, by simulating realizations of uncertainty and comparing our solution with two benchmark methods: the nominal solution to the ACOPF problem with tightened bounds and the DCOPF solution.

In the approach we propose in this thesis, we linearize the quadratic system of equalities in the ACOPF problem, by iteratively taking first-order Taylor Series approximations on small subsets of the feasible space around specified points of approximation. As a result, we end up with an SOCP problem for most instances, which can be solved much faster than many other approaches using SDP approximations. We find that our approach performs well in terms of operational cost in the situation of no uncertainty, and we find that our solution performs much better than the two benchmark models in terms of feasibility when uncertainty is realized. Due to the SOCP formulation of the approximated problem, our approach is also efficient in terms of computation time. However, for some instances it turns out to be necessary to explicitly model the line thermal limits. The latter turns the problem into an SDP problem, which drastically increases computation time. Finally, the maximum amount of uncertainty in the approximated problem for which our algorithm is able to find a solution, depends highly on the instance at hand. Further research is needed to see if our approach can be improved to be more consistently suitable for moderate levels of uncertainty, but overall our solution method provides promising results in terms of robustness, scalability and operational cost.

9.1 Discussion of numerical experiments

Here we provide a brief discussion of the numerical experiments that the previous conclusions are based on. We test our approach on nine different MATPOWER instances. We consider uncertainty as a percentage of the load, and for each instance we run our algorithm for different levels of uncertainty. We consider the situation without correlation between uncertainties, as well as the situation with correlation. For most instances, our algorithm is able to find a solution within seconds in the situation of no correlation between uncertainties. For the largest instance of 1354 buses, it finds a solution within 20 minutes. This is a promising result, considering most papers in the literature on ACOPF with uncertainty do not even consider such large instances. We also see that the larger the size of an instance, the larger the difference in computation time between the situation with and without correlation. Furthermore, we find that our solution results in a marginal cost increase in the situation of no uncertainty: most cost increases are below 0.1%. This can be explained by the fact that we consider subsets of the feasible space around the solution to the nominal problem (with tightened bounds), so our solution is likely to be close to this initial solution. Finally, the maximum amount of uncertainty in the approximated problem for which our algorithm is able to find a solution, varies between only 0.1% and more than 50% of the load. Not every instance is as suitable for large levels of uncertainty, but in Section 9.2 we discuss one way to potentially make our method better for larger uncertainty sets.

In the simulation study, we consider two different sampling methods: drawing from the specified ellipsoid which we aimed to be robust against, and drawing from a multivariate normal distribution. For almost every instance, our solution outperforms the nominal solution with tightened bounds and the DCOPF solution in terms of feasibility after realized uncertainty. This is true for both the samples inside our specified ellipsoid, as well as the samples from the normal distribution. Moreover, the difference in performance is large for many instances. Only for the 1% uncertainty level at 'case30' and 'case39' and low uncertainty levels of 'case9', results are similar for our solution and the nominal solution. Overall, the performance of our method decreases when uncertainty gets larger. The performance is also very dependent on the instance at hand, for both our solution as well as for the nominal solution. In some cases it can make quite a difference in performance if violations of at most 0.1% or 1% are allowed or not. Since the ACOPF itself is also based on assumptions and does not reflect reality completely, acceptable violation sizes should be discussed with experts

in the field of power system optimization.

For two instances with line thermal limits, it is not necessary to include these constraints in the model to obtain a solution that is feasible for the nominal ACOPF problem. For one of these instances, we do evaluate whether a more robust solution can be obtained when these line thermal limits are explicitly modeled. We see that adding these constraints results in the exact same solution for low levels of uncertainty, but for higher levels of uncertainty a different solution is found. For the highest uncertainty level of 20%, including these limits in the model results in a more robust solution for both sampling methods. The downside of including these constraints is that it turns the problem into an SDP problem, with a much higher computation time as a result.

9.2 Limitations and directions for further research

We conclude this paper with some limitations of our research and directions for further research. First of all, we notice that for larger uncertainty sets, either the performance of our obtained solution decreases, or no solution can be found at all. To improve performance for large uncertainty sets, we could choose to consider subsets of the uncertainty set as well. A method to iteratively split the uncertainty set is, for example, proposed by Postek and Hertog (2016). Second, we iteratively consider small subsets of the state variables starting with an initial subset around the nominal solution with tightened bounds. This restricts the searching space, since other more robust solutions might be found in subsets of the feasible space that are further away from the initial solution. It is interesting to see if our method can be improved by considering multiple initial subsets of the domain of the state variables, in order to expand the space in which we search for a robust solution. Third, it would be interesting to see how our method performs for instances with more than 1354 buses. We do not consider this in the thesis, since we consider the computation time too high for larger instances. However, there are some methods to potentially reduce runtime for those instances. Since our approach takes more time to set up the model than to solve it, one can exploit this by constructing the model only once. For different uncertainty levels or different iterations of Algorithm 1, the structure of the model is the same. One can therefore produce a more efficient code so that only the numbers in the model need to be adjusted. Of course, one could also consider a different interface than CVX to call the solver, to improve efficiency in constructing SOCP variables. An alternative way of potentially reducing computation time for instances with line thermal limits is to consider only modeling thermal constraints for branches that are likely to cause violations

of the limits. This still turns the problem into an SDP problem, but with less SDP constraints so that the computational burden is reduced. We have not considered this for 'case1354pegase'; instead we removed the line thermal limits altogether, due to the high runtime the SDP problem would have. It would be interesting to see how our method performs for this instance with line thermal constraints added to the model. One last way of reducing computation time for larger instances, is by considering uncertainty for a smaller specified set of buses. In this thesis, we consider uncertainty at every bus with nonzero load. Since the size of problem (24) depends on the dimension of uncertainty, considering fewer buses with uncertainty can reduce computation time by a lot. This is also a more realistic representation of uncertainty in situations where only some buses display considerable deviations from their forecasted power demand.

References

Baker, K., Dall'Anese, E., and Summers, T. (2016). Distribution-agnostic stochastic optimal power flow for distribution grids. In *2016 North American Power Symposium (NAPS)*, pages 1–6. IEEE.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust optimization*, volume 28. Princeton university press.

Ben-Tal, A., Goryashko, A., Guslitzer, E., and Nemirovski, A. (2004). Adjustable robust solutions of uncertain linear programs. *Mathematical programming*, 99(2):351–376.

Bertsimas, D., Litvinov, E., Sun, X. A., Zhao, J., and Zheng, T. (2012). Adaptive robust optimization for the security constrained unit commitment problem. *IEEE transactions on power systems*, 28(1):52–63.

Bienstock, D., Chertkov, M., and Harnett, S. (2014). Chance-constrained optimal power flow: Risk-aware network control under uncertainty. *Siam Review*, 56(3):461–495.

Bienstock, D., Escobar, M., Gentile, C., and Liberti, L. (2022). Mathematical programming formulations for the alternating current optimal power flow problem. *Annals of Operations Research*, pages 1–39.

Bienstock, D. and Özbay, N. (2008). Computing robust basestock levels. *Discrete Optimization*, 5(2):389–414.

Bienstock, D. and Verma, A. (2019). Strong np-hardness of ac power flows feasibility. *Operations Research Letters*, 47(6):494–501.

Bingane, C., Anjos, M. F., and Le Digabel, S. (2018). Tight-and-cheap conic relaxation for the ac optimal power flow problem. *IEEE Transactions on Power Systems*, 33(6):7181–7188.

Cain, M. B., O'neill, R. P., Castillo, A., et al. (2012). History of optimal power flow and formulations. *Federal Energy Regulatory Commission*, 1:1–36.

Carpentier, J. (1962). Contribution a l'etude du dispatching economique. *Bulletin de la Societe Francaise des Electriciens*, 3(1):431–447.

Chatzivasileiadis, S. (2018). Optimization in modern power systems. *Lecture Notes. Tech. Univ. of Denmark. Available online: https://arxiv. org/pdf/1811.00943. pdf.*

Dezert, J. and Musso, C. (2001). An efficient method for generating points uniformly distributed in hyperellipsoids. In *Proceedings of the Workshop on Estimation, Tracking and Fusion: A Tribute to Yaakov Bar-Shalom*, volume 7.

Ding, T., Bie, Z., Bai, L., and Li, F. (2016). Adjustable robust optimal power flow with the price of robustness for large-scale power systems. *IET Generation, Transmission & Distribution*, 10(1):164–174.

Frank, S., Steponavice, I., and Rebennack, S. (2012). Optimal power flow: a bibliographic survey i. *Energy systems*, 3(3):221–258.

Grant, M. and Boyd, S. (2008). Graph implementations for nonsmooth convex programs. In Blondel, V., Boyd, S., and Kimura, H., editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited. `http://stanford.edu/~boyd/graph_dcp.html`.

Grant, M. and Boyd, S. (2014). CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`.

Isenberg, N. M., Akula, P., Eslick, J. C., Bhattacharyya, D., Miller, D. C., and Gounaris, C. E. (2021). A generalized cutting-set approach for nonlinear robust optimization in process systems engineering. *AIChE Journal*, 67(5):e17175.

Kumar, P., Kothari, D. P., et al. (2005). Recent philosophies of automatic generation control strategies in power systems. *IEEE transactions on power systems*, 20(1):346–357.

Kundur, P., Paserba, J., Ajjarapu, V., Andersson, G., Bose, A., Canizares, C., Hatziargyriou, N., Hill, D., Stankovic, A., Taylor, C., et al. (2004). Definition and classification of power system stability ieee/cigre joint task force on stability terms and definitions. *IEEE transactions on Power Systems*, 19(3):1387–1401.

Kuryatnikova, O., Ghaddar, B., and Molzahn, D. K. (2021). Adjustable robust two-stage polynomial optimization with application to ac optimal power flow. *arXiv preprint arXiv:2104.03107*.

Lee, D., Turitsyn, K., Molzahn, D. K., and Roald, L. A. (2021). Robust ac optimal power flow with robust convex restriction. *IEEE Transactions on Power Systems*, 36(6):4953–4966.

Lee, D., Turitsyn, K., and Slotine, J.-J. (2019). Sequential convex restriction and its applications in robust optimization. *arXiv preprint arXiv:1909.01778*.

Lehmann, K., Grastien, A., and Van Hentenryck, P. (2015). Ac-feasibility on tree networks is np-hard. *IEEE Transactions on Power Systems*, 31(1):798–801.

Marandi, A. and Den Hertog, D. (2018). When are static and adjustable robust optimization problems with constraint-wise uncertainty equivalent? *Mathematical programming*, 170(2):555–568.

McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming*, 10(1):147–175.

Molzahn, D. and Roald, L. A. (2019). Grid-aware versus grid-agnostic distribution system control: A method for certifying engineering constraint satisfaction.

Molzahn, D. K. and Roald, L. A. (2018). Towards an ac optimal power flow algorithm with robust feasibility guarantees. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–7. IEEE.

Peschon, J., Piercy, D. S., Tinney, W. F., Tveit, O. J., and Cuenod, M. (1968). Optimum control of reactive power flow. *IEEE Transactions on Power Apparatus and Systems*, (1):40–48.

Postek, K. and Hertog, D. d. (2016). Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS Journal on Computing*, 28(3):553–574.

Roald, L. and Andersson, G. (2017). Chance-constrained ac optimal power flow: Reformulations and efficient algorithms. *IEEE Transactions on Power Systems*, 33(3):2906–2918.

Sadat, S. A. and Kim, K. (2021). Numerical performance of different formulations for alternating current optimal power flow. In *2021 31st Australasian Universities Power Engineering Conference (AUPEC)*, pages 1–6. IEEE.

Stott, B., Jardim, J., and Alsaç, O. (2009). Dc power flow revisited. *IEEE Transactions on Power Systems*, 24(3):1290–1300.

Tinney, W. F. and Hart, C. E. (1967). Power flow solution by newton's method. *IEEE Transactions on Power Apparatus and systems*, (11):1449–1460.

Toh, K.-C., Todd, M. J., and Tütüncü, R. H. (1999). Sdpt3—a matlab software package for semidefinite programming. *Optimization methods and software*, 11(1-4):545–581.

Tütüncü, R. H., Toh, K.-C., and Todd, M. J. (2003). Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical programming*, 95(2):189–217.

Venzke, A., Halilbasic, L., Markovic, U., Hug, G., and Chatzivasileiadis, S. (2017). Convex relaxations of chance constrained ac optimal power flow. *IEEE Transactions on Power Systems*, 33(3):2829–2841.

Verbic, G. and Canizares, C. A. (2006). Probabilistic optimal power flow in electricity markets based on a two-point estimate method. *IEEE transactions on Power Systems*, 21(4):1883–1893.

Wang, H., Murillo-Sanchez, C. E., Zimmerman, R. D., and Thomas, R. J. (2007). On computational issues of market-based optimal power flow. *IEEE Transactions on Power Systems*, 22(3):1185–1193.

Wirfs-Brock, J. (2016). Power outages on the rise across the us. `https://grid.insideenergy.org/outages-on-the-rise`.

Wood, A. J. and Wollenberg, B. F. (1996). *Power generation, operation, and control*. John Wiley & Sons, 2nd edition.

Yakubovich, V. A. (1971). S-procedure in nolinear control theory. *Vestnik Leninggradskogo Universiteta, Ser. Matematika*, pages 62–77.

Zhang, H. and Li, P. (2011). Chance constrained programming for optimal power flow under uncertainty. *IEEE Transactions on Power Systems*, 26(4):2417–2424.

Zimmerman, R. D., Murillo-Sánchez, C. E., and Thomas, R. J. (2010). Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on power systems*, 26(1):12–19.

Appendix A Alternative modeling choices

A.1 Infinity norm to construct the subsets of state variables

In Section 5.8, it is explained how we choose the small subset in which we search for a solution. For this, we choose to model $||\boldsymbol{x}-\hat{\boldsymbol{x}}|| \leq \varepsilon$, where we use the Euclidean norm. However, other norms can be chosen as well, such as the infinity norm $||\cdot||_\infty$. The infinity norm equals the maximum of the absolute values of its elements. To satisfy that the maximum of the absolute values is smaller than ε, we can restrict every absolute value to be smaller than this value. This results in $|x_j - \hat{x}_j| \leq \varepsilon$ for every j. Eliminating the absolute value, we obtain $x_j - \hat{x}_j \leq \varepsilon$ and $\hat{x}_j - x_j \leq \varepsilon$ for every j. Substituting $\boldsymbol{x}$ by its nominal linear expression $\boldsymbol{x} = A\boldsymbol{y} + \boldsymbol{c}$ and writing this compactly, we get:

$$\begin{aligned} A\boldsymbol{y} + \boldsymbol{c} - \hat{\boldsymbol{x}} &\leq \boldsymbol{\varepsilon} \\ \hat{\boldsymbol{x}} - A\boldsymbol{y} - \boldsymbol{c} &\leq \boldsymbol{\varepsilon} \end{aligned} \tag{29}$$

in which $\boldsymbol{\varepsilon} = [\varepsilon, \varepsilon, \ldots, \varepsilon]^T$. If the infinity norm is chosen instead of the Euclidean norm, one replaces constraint (19) by (29).

A.2 Alternative options for dealing with the voltage at the reference bus

In Section 5.10, we provide three inequality constraints to approximate the non-convex constraint $v_1^M = (v_1^r)^2$. While constraints (23) are sufficient in our results, there are no guarantees that the obtained solution satisfies the original non-convex constraint. In that case, one might consider to implement one of the two options below.

One alternative way of dealing with this non-convex constraint, is to put the variable v_1^r in the objective and add a negative penalty term, so that the higher v_1^r, the better the objective function. However, this penalty term should be chosen carefully, since it can also influence the solution to the problem in an unintended way.

Another way of dealing with this constraint, is to allow the voltage to be negative at the reference bus. We restricted v_1^r to be positive throughout this thesis. However, this is not necessary. In the case one allows it to be negative, the value of v_1^r does not follow directly from the value of v_1^M anymore. Therefore, an extra state variable v_1^r is obtained, together with an extra nonlinear

equality constraint in the state variables: $v_1^M = (v_1^r)^2$. This constraint can then be linearized in the exact same way as the other constraints, and can be included in the linear system (17). It is not clear if this would increase the precision of our approach, since the nonlinear constraint needs to be approximated. Moreover, in the ACOPF literature, it is common to set the phase angle of the voltage at the reference bus to zero, which implies that in the rectangular formulation, the real part of the voltage is positive (otherwise the phase angle could also be 180°).

Appendix B Robust counterpart of a linear constraint with conic uncertainty

After the state variables are eliminated, the problem exists of inequality constraints in $\boldsymbol{y}$ and $\boldsymbol{\zeta}$, of the form:

$$\boldsymbol{g}_i(\boldsymbol{y}, \boldsymbol{\zeta}) = (\boldsymbol{a^i})^T \boldsymbol{y} + (\boldsymbol{b^i})^T \boldsymbol{\zeta} + c^i \geq 0 \quad \forall i \in \{\boldsymbol{g}_i\}, \forall \boldsymbol{\zeta} \in \Omega,$$

with Ω as in (20). For fixed i, the following holds:

$$\begin{aligned}
& (\boldsymbol{a^i})^T \boldsymbol{y} + (\boldsymbol{b^i})^T \boldsymbol{\zeta} + c^i \geq 0 \quad \forall \boldsymbol{\zeta} \in \Omega \\
\iff & \min_{\boldsymbol{\zeta}} \Big\{ (\boldsymbol{a^i})^T \boldsymbol{y} + (\boldsymbol{b^i})^T \boldsymbol{\zeta} + c^i : M\boldsymbol{\zeta} + m \in \mathfrak{L}^{|U|+1} \Big\} \geq 0 \\
\iff & (\boldsymbol{a^i})^T \boldsymbol{y} + c^i + \min_{\boldsymbol{\zeta}} \Big\{ (\boldsymbol{b^i})^T \boldsymbol{\zeta} : M\boldsymbol{\zeta} + m \in \mathfrak{L}^{|U|+1} \Big\} \geq 0 \\
\iff & (\boldsymbol{a^i})^T \boldsymbol{y} + c^i + \max_{\boldsymbol{z^i} \in \mathfrak{L}^{|U|+1}} \min_{\boldsymbol{\zeta}} \Big\{ (\boldsymbol{b^i})^T \boldsymbol{\zeta} - (\boldsymbol{z^i})^T (M\boldsymbol{\zeta} + m) \Big\} \geq 0 \\
\iff & (\boldsymbol{a^i})^T \boldsymbol{y} + c^i + \max_{\boldsymbol{z^i} \in \mathfrak{L}^{|U|+1}} \Big\{ -(\boldsymbol{z^i})^T m + \min_{\boldsymbol{\zeta}} \Big\{ \Big((\boldsymbol{b^i})^T - (\boldsymbol{z^i})^T M \Big) \boldsymbol{\zeta} \Big\} \Big\} \geq 0 \\
\iff & (\boldsymbol{a^i})^T \boldsymbol{y} + c^i + \max_{\boldsymbol{z^i} \in \mathfrak{L}^{|U|+1}} \Big\{ -(\boldsymbol{z^i})^T m : (\boldsymbol{b^i})^T - (\boldsymbol{z^i})^T M = \boldsymbol{0}^T \Big\} \geq 0 \\
\iff & (\boldsymbol{a^i})^T \boldsymbol{y} + c^i - (\boldsymbol{z^i})^T m \geq 0 \\
& (\boldsymbol{z^i})^T M = (\boldsymbol{b^i})^T \\
& \boldsymbol{z^i} \in \mathfrak{L}^{|U|+1},
\end{aligned}$$

which is equivalent to (21). Here, in the third equivalence a dual variable $\boldsymbol{z^i} \in \mathfrak{L}^{|U|+1}$ (the second-order cone is self-dual) is introduced and the original objective that must be minimized is replaced by its Lagrangian function. This turns the minimization into an unconstrained problem in which the dual variable $\boldsymbol{z^i}$ serves as a penalty term if $(\boldsymbol{z^i})^T(M\boldsymbol{\zeta} + m)$ is negative. The reason this term should not be negative is that by definition of the dual cone, it holds that $\langle \boldsymbol{z^i}, M\boldsymbol{\zeta} + m \rangle \geq 0$. The

sixth line holds because the minimization problem would be unbounded if $(\boldsymbol{b^i})^T - (\boldsymbol{z^i})^T M \neq \mathbf{0}^T$.

Appendix C Coefficients for line thermal limits

In this section, we give specifications for the coefficients in expression (25), which represents the line thermal constraints $(T_{p_l^f})^2 + (T_{q_l^f})^2 \leq (s_l^{\max})^2$ and $(T_{p_l^t})^2 + (T_{q_l^t})^2 \leq (s_l^{\max})^2$. We derive the coefficients for the constraint $(T_{p_l^f})^2 + (T_{q_l^f})^2 \leq (s_l^{\max})^2$; to derive the coefficients for the expression $(T_{p_l^t})^2 + (T_{q_l^t})^2 \leq (s_l^{\max})^2$, similar steps with different coefficients can be followed. From (11) and (12), it follows that the Taylor Series can be written as a linear expression in $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$
\begin{aligned}
T_{p_l^f} &= \boldsymbol{\gamma}_{p_l^f}^T \boldsymbol{x} + \boldsymbol{\lambda}_{p_l^f}^T \boldsymbol{y} + \kappa_{p_l^f} \\
T_{q_l^f} &= \boldsymbol{\gamma}_{q_l^f}^T \boldsymbol{x} + \boldsymbol{\lambda}_{q_l^f}^T \boldsymbol{y} + \kappa_{q_l^f}.
\end{aligned}
$$

The coefficients $\boldsymbol{\gamma}_{p_l^f}^T$, $\boldsymbol{\lambda}_{p_l^f}^T$ and $\kappa_{p_l^f}$ can be derived from (11) and (12) and the specifications of the matrices in Sections 5.3 until 5.5. We eliminate the state variables by substituting $\boldsymbol{x} = A\boldsymbol{y} + B\boldsymbol{\zeta} + \boldsymbol{c}$ to obtain a linear form in $\boldsymbol{y}$ and $\boldsymbol{\zeta}$:

$$
\begin{aligned}
T_{p_l^f} &= \boldsymbol{a}_{p_l^f}^T \boldsymbol{y} + \boldsymbol{b}_{p_l^f}^T \boldsymbol{\zeta} + c_{p_l^f} \\
T_{q_l^f} &= \boldsymbol{a}_{q_l^f}^T \boldsymbol{y} + \boldsymbol{b}_{q_l^f}^T \boldsymbol{\zeta} + c_{q_l^f},
\end{aligned}
$$

where $\boldsymbol{a}_{p_l^f}^T = \boldsymbol{\gamma}_{p_l^f}^T A + \boldsymbol{\lambda}_{p_l^f}^T$, $\boldsymbol{b}_{q_l^f}^T = \boldsymbol{\gamma}_{p_l^f}^T B$ and $c_{q_l^f} = \boldsymbol{\gamma}_{p_l^f}^T c + \kappa_{p_l^f}$. For q, the same expressions hold but with subscript q. Now, squaring $T_{p_l^f}$ and $T_{q_l^f}$ gives:

$$
\begin{aligned}
\left(T_{p_l^f}\right)^2 &= \boldsymbol{y}^T \boldsymbol{a}_{p_l^f} \boldsymbol{a}_{p_l^f}^T \boldsymbol{y} + \boldsymbol{\zeta}^T \boldsymbol{b}_{p_l^f} \boldsymbol{b}_{p_l^f}^T \boldsymbol{\zeta} + 2\boldsymbol{y}^T \boldsymbol{a}_{p_l^f} \boldsymbol{b}_{p_l^f}^T \boldsymbol{\zeta} + 2c_{p_l^f} \boldsymbol{a}_{p_l^f}^T \boldsymbol{y} + 2c_{p_l^f} \boldsymbol{b}_{p_l^f}^T \boldsymbol{\zeta} + c_{p_l^f}^2 \\
\left(T_{q_l^f}\right)^2 &= \boldsymbol{y}^T \boldsymbol{a}_{q_l^f} \boldsymbol{a}_{q_l^f}^T \boldsymbol{y} + \boldsymbol{\zeta}^T \boldsymbol{b}_{q_l^f} \boldsymbol{b}_{q_l^f}^T \boldsymbol{\zeta} + 2\boldsymbol{y}^T \boldsymbol{a}_{q_l^f} \boldsymbol{b}_{q_l^f}^T \boldsymbol{\zeta} + 2c_{q_l^f} \boldsymbol{a}_{q_l^f}^T \boldsymbol{y} + 2c_{q_l^f} \boldsymbol{b}_{q_l^f}^T \boldsymbol{\zeta} + c_{q_l^f}^2.
\end{aligned}
$$

Finally, for a given branch l, we obtain the constraint:

$$
\begin{aligned}
\left(T_{p_l^f}\right)^2 + \left(T_{q_l^f}\right)^2 &= \boldsymbol{y}^T \left(\boldsymbol{a}_{p_l^f} \boldsymbol{a}_{p_l^f}^T + \boldsymbol{a}_{q_l^f} \boldsymbol{a}_{q_l^f}^T\right) \boldsymbol{y} + \boldsymbol{\zeta}^T \left(\boldsymbol{b}_{p_l^f} \boldsymbol{b}_{p_l^f}^T + \boldsymbol{b}_{q_l^f} \boldsymbol{b}_{q_l^f}^T\right) \boldsymbol{\zeta} + 2\boldsymbol{y}^T \left(\boldsymbol{a}_{p_l^f} \boldsymbol{b}_{p_l^f}^T + \boldsymbol{a}_{q_l^f} \boldsymbol{b}_{q_l^f}^T\right) \boldsymbol{\zeta} \\
&\quad + 2\left(c_{p_l^f} \boldsymbol{a}_{p_l^f}^T + c_{q_l^f} \boldsymbol{a}_{q_l^f}^T\right) \boldsymbol{y} + 2\left(c_{p_l^f} \boldsymbol{b}_{p_l^f}^T + c_{q_l^f} \boldsymbol{b}_{q_l^f}^T\right) \boldsymbol{\zeta} + c_{p_l^f}^2 + c_{q_l^f}^2 \leq (s_l^{\max})^2 \quad \forall \boldsymbol{\zeta} \in \Omega.
\end{aligned}
$$

Comparing this with expression (25), one can easily see what the definitions of $A_l, B_l, C_l, \boldsymbol{a}_l^T, \boldsymbol{b}_l^T$ and c_l are for the $from$-end of branch l.

Appendix D SDP reformulations of robust line thermal constraints

In this section, we prove that (27) follows from the second statement in (26). The following holds:

$$
\begin{aligned}
& \boldsymbol{g}_l(\boldsymbol{y},\boldsymbol{\zeta}) - \mu_l(\boldsymbol{\zeta}^T\Sigma^{-1}\boldsymbol{\zeta} - r^2) \leq 0, \quad \mu_l \geq 0 \quad \forall \boldsymbol{\zeta} \in \mathbb{R}^{|U|} \\
\iff & \boldsymbol{y}^T A_l \boldsymbol{y} + \boldsymbol{\zeta}^T B_l \boldsymbol{\zeta} + \boldsymbol{y}^T C_l \boldsymbol{\zeta} + \boldsymbol{a_l}^T \boldsymbol{y} + \boldsymbol{b_l}^T \boldsymbol{\zeta} + c_l - (s_l^{\max})^2 - \mu_l(\boldsymbol{\zeta}^T\Sigma^{-1}\boldsymbol{\zeta} - r^2) \leq 0, \quad \mu_l \geq 0 \\
& \forall \boldsymbol{\zeta} \in \mathbb{R}^{|U|} \\
\iff & \boldsymbol{\zeta}^T(B_l - \mu_l\Sigma^{-1})\boldsymbol{\zeta} + (\boldsymbol{y}^T C_l + \boldsymbol{b_l}^T)\boldsymbol{\zeta} + \boldsymbol{y}^T A_l \boldsymbol{y} + \boldsymbol{a_l}^T \boldsymbol{y} + c_l + \mu_l r^2 - (s_l^{\max})^2 \leq 0, \quad \mu_l \geq 0 \\
& \forall \boldsymbol{\zeta} \in \mathbb{R}^{|U|} \\
\iff & \boldsymbol{\zeta}^T(\mu_l\Sigma^{-1} - B_l)\boldsymbol{\zeta} - (\boldsymbol{y}^T C_l + \boldsymbol{b_l}^T)\boldsymbol{\zeta} - \boldsymbol{y}^T A_l \boldsymbol{y} - \boldsymbol{a_l}^T \boldsymbol{y} - c_l - \mu_l r^2 + (s_l^{\max})^2 \geq 0, \quad \mu_l \geq 0 \\
& \forall \boldsymbol{\zeta} \in \mathbb{R}^{|U|} \\
\iff & \begin{bmatrix} \boldsymbol{\zeta} \\ 1 \end{bmatrix}^T H \begin{bmatrix} \boldsymbol{\zeta} \\ 1 \end{bmatrix} \geq 0, \quad H = \begin{bmatrix} \mu_l\Sigma^{-1} - B_l & -\frac{1}{2}\left(C_l^T \boldsymbol{y} + \boldsymbol{b_l}\right) \\ -\frac{1}{2}\left(\boldsymbol{y}^T C_l + \boldsymbol{b_l}^T\right) & (s_l^{\max})^2 - \boldsymbol{a_l}^T \boldsymbol{y} - c_l - \mu_l r^2 - \boldsymbol{y}^T A_l \boldsymbol{y} \end{bmatrix}, \ \mu_l \geq 0 \\
& \forall \boldsymbol{\zeta} \in \mathbb{R}^{|U|} \\
\iff & \boldsymbol{u}^T H \boldsymbol{u} \geq 0, \quad H = \begin{bmatrix} \mu_l\Sigma^{-1} - B_l & -\frac{1}{2}\left(C_l^T \boldsymbol{y} + \boldsymbol{b_l}\right) \\ -\frac{1}{2}\left(\boldsymbol{y}^T C_l + \boldsymbol{b_l}^T\right) & (s_l^{\max})^2 - \boldsymbol{a_l}^T \boldsymbol{y} - c_l - \mu_l r^2 - \boldsymbol{y}^T A_l \boldsymbol{y} \end{bmatrix}, \ \mu_l \geq 0 \\
& \forall \boldsymbol{u} \in \mathbb{R}^{|U|+1} \\
\iff & \begin{bmatrix} \mu_l\Sigma^{-1} - B_l & -\frac{1}{2}\left(C_l^T \boldsymbol{y} + \boldsymbol{b_l}\right) \\ -\frac{1}{2}\left(\boldsymbol{y}^T C_l + \boldsymbol{b_l}^T\right) & (s_l^{\max})^2 - \boldsymbol{a_l}^T \boldsymbol{y} - c_l - \mu_l r^2 - \boldsymbol{y}^T A_l \boldsymbol{y} \end{bmatrix} \succeq 0, \quad \mu_l \geq 0 \\
\iff & \begin{bmatrix} \mu_l\Sigma^{-1} - B_l & -\frac{1}{2}\left(C_l^T \boldsymbol{y} + \boldsymbol{b_l}\right) \\ -\frac{1}{2}\left(\boldsymbol{y}^T C_l + \boldsymbol{b_l}^T\right) & (s_l^{\max})^2 - \boldsymbol{a_l}^T \boldsymbol{y} - c_l - \mu_l r^2 - \nu_l \end{bmatrix} \succeq 0, \quad \mu_l \geq 0, \quad \nu_l = \boldsymbol{y}^T A_l \boldsymbol{y}.
\end{aligned}
$$

Here, the fifth equivalence is not trivial, so we prove it. Obviously, ($\Leftarrow$) holds because if it holds for every vector $\boldsymbol{u}$, it also holds for every vector with last element equal to one. We need to prove ($\Rightarrow$), which comes down to proving that

$$
\begin{bmatrix} \boldsymbol{\zeta} \\ 1 \end{bmatrix}^T H \begin{bmatrix} \boldsymbol{\zeta} \\ 1 \end{bmatrix} \geq 0 \quad \forall \boldsymbol{\zeta} \in \mathbb{R}^{|U|} \quad \Rightarrow \quad \boldsymbol{u}^T H \boldsymbol{u} \geq 0 \quad \forall \boldsymbol{u} \in \mathbb{R}^{|U|+1}. \tag{30}
$$

To prove this, we first prove that this holds for all $\boldsymbol{u} \in \mathbb{R}^{|U|+1}$ with $u_{|U|+1} \neq 0$ ($u_{|U|+1}$ denotes the last element of $\boldsymbol{u}$). Let $\boldsymbol{u}$ be such a vector. Then we can just rescale $[\boldsymbol{\zeta}^T, 1]^T$ with a factor $u_{|U|+1}$ to $\boldsymbol{u}$ by choosing $\boldsymbol{\zeta} = \frac{1}{u_{|U|+1}}\boldsymbol{u}$. We get that:

$$\begin{bmatrix} \frac{1}{u_{|U|+1}}\boldsymbol{u} \\ 1 \end{bmatrix}^T H \begin{bmatrix} \frac{1}{u_{|U|+1}}\boldsymbol{u} \\ 1 \end{bmatrix} \geq 0 \quad \Longleftrightarrow \quad u_{|U|+1}^2 \begin{bmatrix} \frac{1}{u_{|U|+1}}\boldsymbol{u} \\ 1 \end{bmatrix}^T H \begin{bmatrix} \frac{1}{u_{|U|+1}}\boldsymbol{u} \\ 1 \end{bmatrix} \geq 0 \quad \Longleftrightarrow \quad \boldsymbol{u}^T H \boldsymbol{u} \geq 0.$$

Therefore, (30) holds for all $\boldsymbol{u} \in \mathbb{R}^{|U|+1}$ with $u_{|U|+1} \neq 0$.

Now, we need to prove that (30) holds for all $\boldsymbol{u} \in \mathbb{R}^{|U|+1}$ with $u_{|U|+1} = 0$. Let $\boldsymbol{u_1}$ be such a vector. Suppose by contradiction that $\boldsymbol{u_1}^T H \boldsymbol{u_1} < 0$. Then we know that there exists a $\boldsymbol{u_2}$ with nonzero last element such that $\boldsymbol{u_2}^T H \boldsymbol{u_2} < 0$, by continuity of $\boldsymbol{u}^T H \boldsymbol{u}$. However, we just proved that every vector $\boldsymbol{u}$ with nonzero last element satisfies $\boldsymbol{u}^T H \boldsymbol{u} \geq 0$, so this results in a contradiction. Therefore, (30) holds for all $\boldsymbol{u} \in \mathbb{R}^{|U|+1}$ with $u_{|U|+1} = 0$ as well, and consequently it holds for all $\boldsymbol{u} \in \mathbb{R}^{|U|+1}$.