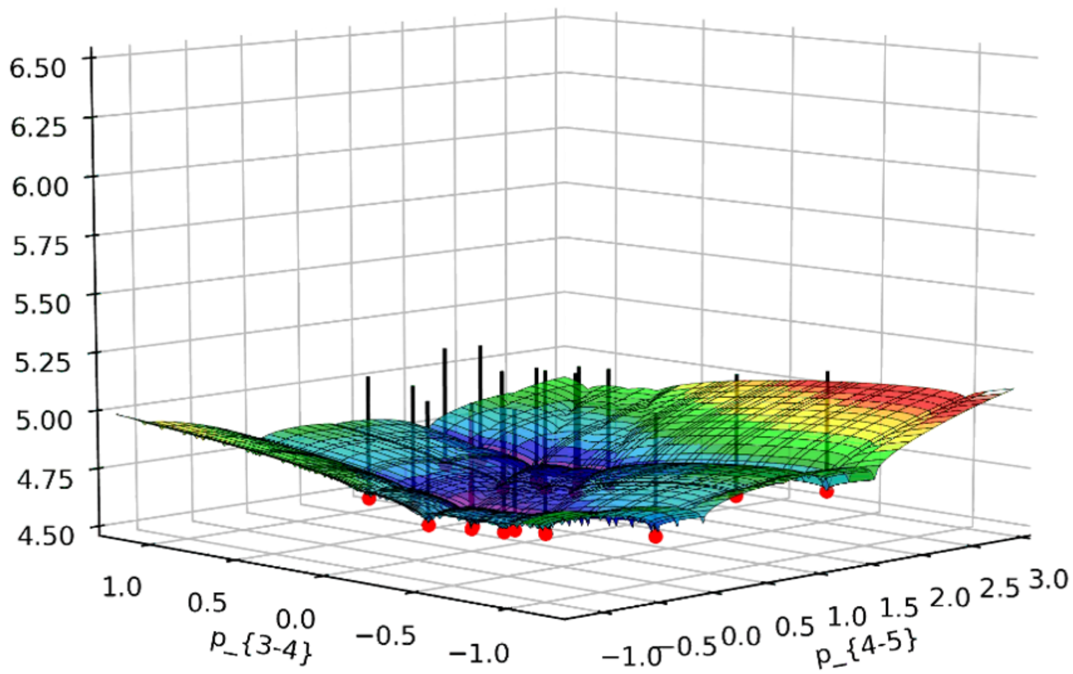
EKSTERN RAPPORT

Nr 5/2019

Computing the power distance parameter

Commissioned by NVE

THEMA Consulting Group



Ekstern rapport nr 5-2019

Computing the power distance parameter

Utgitt av: Norges vassdrags- og energidirektorat
Redaktør: Ole-Petter Kordahl
Forfatter: THEMA Consulting Group

Trykk: NVEs hustrykkeri
Forsidefoto: THEMA Consulting Group
ISBN: 978-82-410-1820-6

Sammendrag: In this report, THEMA provides a method for estimating measures of electric power distance and energy distance, as well as a discussion of the applicability of the proposed method.

Emneord: Power distance, energy distance, economic regulation, benchmarking, exogenous outputs, DSO, transportation of electricity, non-convex optimization

Norges vassdrags- og energidirektorat
Middelthunsgate 29
Postboks 5091 Majorstua
0301 OSLO

Telefon: 22 95 95 95
Epost: nve@nve.no
Internett: www.nve.no

12.02.2019

Preface

The Norwegian Water resources and Energy Directorate (NVE) determines the distribution system operators' (DSOs) allowed revenues every year. NVE applies Data Envelopment Analysis (DEA) as benchmarking method when determining the DSOs' allowed revenues. The DEA model calculates each company's relative efficiency by comparing all companies' output/input ratios. The outputs are proxies for the tasks the DSOs have to solve when building, maintaining and operating the grid. The input is a measure of the DSOs' costs related to their tasks.

The power sector is changing and so is the tasks of the DSO. We want to explore new output variables for describing the task related to building, maintaining and operating grid infrastructure. It is important that the variables take into account that customers have different demand for power and energy. They should also take into account where demand, generation and injection from adjacent grids are located in the electricity network. The cost of transporting electricity depends on both volume of power/energy and the transportation distance for the power/energy. We call a compounded variable that includes both the distance and the volume for the electric power/energy distance.

NVE has asked THEMA Consulting Group to provide a method for estimating measures of electric power and energy distance. The report presents different mathematical approaches for finding the optimal power distance and gives a discussion on advantages and challenges with these. The conclusion is that it is computationally difficult to calculate the optimal power distance, but that we should investigate some alternative approaches.

Oslo, February 2019

Ove Flataker
Director,
Energy Regulatory Authority

Tore Langset
Head of Section,
Section for Economic Regulation

Computing the power distance parameter

Commissioned by NVE

THEMA
CONSULTING GROUP

Project info

Project name
New methods for an exogenous demand measure in power distribution

Client
Norges vassdrags- og energidirektorat (NVE)

Project number
NVE-18-08

Report info

Report availability
NVE internal

Date of publication
11th January 2019

About the project

Currently, the income regulation of DSOs in Norway considers the number of customers, the line length and the number of stations operated by each DSO as the task or output generated by the DSO. These measures do not take the demand distribution into account, and are to an extend under the control of the DSO. Hence, NVE has proposed the power distance as an alternative measure. This report investigates 1) how the power distance can be computed for complex networks, and 2) what properties the power distance has in respect to grid regulation.

Project team

Project manager
Dr. Theodor Borsche

Contributors (alphabetically)
Kristin Arnesen
Jacob Koren Brekke
Åsmund Jensen

Contents

1. The power distance as a new measure in revenue regulation

Norges Vassdrags- og Energidirektorat (NVE) is responsible for regulating Norwegian electricity network companies (Distribution System Operator (DSO)). A key element of the regulation of distribution grids is NVE's Data Envelopment Analysis (DEA) models. NVE's DEA models are designed to benchmark the costs of a network company given a set of outputs that describe the tasks of the given company. In the distribution grid, the outputs are the number of customers, kilometres of lines and the number of substations as proxies for demand facing each DSO.

NVE is now considering whether and how to replace the existing output measures by exogenous measures for electric power and energy distance. In this report, THEMA provides a method for estimating measures of electric power distance and energy distance, as well as a discussion of the applicability of the proposed method. The main part of the report concerns the calculation of the power distance, and in Chapter 3 we discuss how to move from the electric power distance to a measure of the energy distance.

1.1. Motivation for studying the electric power distance

The current outputs of the distribution grid, being number of customers, kilometres of lines and the number of substations, are all proxies for describing the actual task of the DSO. The actual task is to cover demand at its customers, subject to geographical limitations and challenges with weather and climate. NVE wishes to define a more precise measure of the actual task of each DSO, that captures the increasingly heterogeneous companies.

The electric power distance P_d is defined as the distance of a line L multiplied with the amount of power P transported on the line. The relationship between the cost of transporting power and the amount of power transported is however a logarithmic relationship. To account for the logarithmic relationship between cost and power we introduce the α-parameter as a value between 0 and 1, and define the electric power distance as the distance L multiplied with P to the power of α.

$$P_d = LP^{\alpha}$$

The relevant measure for the task of a DSO is the minimum electric power distance within its grid area, given an obligation to cover all loads and to handle power fed into the grid from distributed generation. The minimum electric power distance is found through an optimisation procedure minimising the relevant distances the power must be distributed to satisfy the exogenous demand.

The magnitude of the power distance strongly depends on the underlying distribution of demand, and therefore captures that the task of supplying power to clients depends on the demand distribution. A simple example of two DSOs with the same topology but different distribution of demand illustrate how the tasks of the two companies are in fact very different.

Figure 1.1 depict a simple radial grid for the two DSOs. Both companies have demand nodes at a distance of 1 and 11 km from the substation, the same grid topology, amount of customers, and number of substations. The difference between DSO A and DSO B is that A has to provide 1 MW of demand at node 1, and 10 MW of demand at node

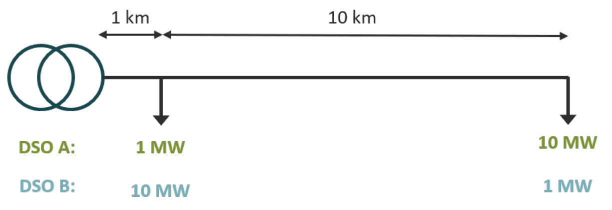


Figure 1.1.: *Example of two DSOs with the same topology, but different demand distribution*

2, while B has the opposite demand distribution.

It can be argued that DSO B has an easier task of supplying energy to its clients - or in other words that DSO B has a lower output in terms of transported power and energy when accounting for the distance. Using the definition of electric power distance as proposed above, this effect will be captured:

$$P_{d,\mathrm{A}} = (11\,\mathrm{MW})^{\alpha} \cdot 1\,\mathrm{km} + (10\,\mathrm{MW})^{\alpha} \cdot 10\,\mathrm{km} = 111\,\mathrm{MW\,km}$$

and

$$P_{d,\mathrm{B}} = (11\,\mathrm{MW})^{\alpha} \cdot 1\,\mathrm{km} + (1\,\mathrm{MW})^{\alpha} \cdot 10\,\mathrm{km} = 21\,\mathrm{MW\,km}$$

Here we used α equal 1.0, but other values would give a similar effect.

Some relevant observations regarding the value of α:

- $\alpha = 0$: Power demand becomes irrelevant, and P_d is equal to the minimum length of lines ensuring that all nodes are connected – a problem known as the minimum spanning tree. This is a classical problem in the class of mixed-integer problems.
- $\alpha = 1$: Assumes that the cost of transporting power increase linearly with amount of power transported.
- $0 < \alpha < 1$: *The area studied in this report.* This parameter choice models economies of scale in increasing power transfer capacity of a line. The problem becomes non-linear with an α between 0 and 1.

The example with DSO A and DSO B illustrates how the two companies might look similar with the current outputs in the DEA model, while the actual tasks are quite different. Due to the non-linear relationship between cost and power, the tasks of the two companies are not as different as suggested when $\alpha = 1$, but a power distance given an α between 0 and 1 might be better equipped to describe the actual task of the companies than the existing outputs.

1.2. Properties of the power distance parameter

The definition of the power distance parameter in this report is based on existing grid topology. That is, power can only travel between nodes where there is actual physical grid, but there are no constraints on the amount of power that is allowed to flow on each line. The mathematical model described in Section 1.4 does not compute an optimal topology based on the distribution of nodes and demand, but only the minimal grid needed to supply demand within the given topology. If α is smaller than 1, any optimal grid will be a tree (in a graph theoretic sense) - meaning an optimal grid minimising the electric power distance is not meshed and have no loops.

However, computing the power distance is not trivial. With an α-parameter between 0 and 1, the problem, which is formalised in Section 1.4, becomes non-linear and non-convex. This implies that minimising the electric power distance is a NP-hard problem with both global and local optima.

To highlight the issue created by non-convexity, a simple example is described in Figure 1.2. The figure illustrates a node with demand 1 MW connected by two different paths of length 2 and 1. α is assumed equal to 0.5. If we compute the electric distance for different fractions of power sent via each line, we find two optima marked by circles in

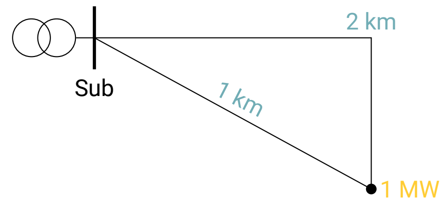


Figure 1.2.: *Example highlighting the non-convex property of the problem*

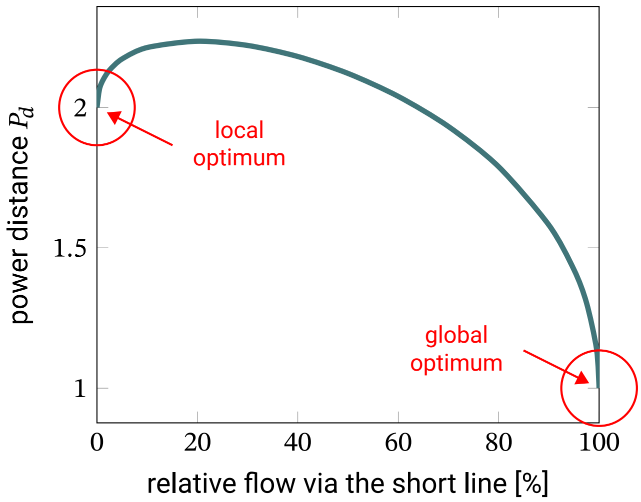


Figure 1.3.: *Power distance as a function of the share of power transported via the short line*

Figure 1.3. Either, all power is sent via line 1, or all power is sent via line 2. All though it seems obvious from Figure 1.3 that the optimal solution is to send all power via the short line, a numerical algorithm trying to find the power distance might end up with all power flowing via the long line, which is a globally suboptimal local optimum.

The implications of the properties of the power distance parameter are revisited and explained in more detail later in this report. In short, finding the optimal electric power distance for a given grid is a complex and time consuming task.

1.3. Test cases

To test the optimisation problem and solution procedure, we have defined two different test cases based on publicly available data.

1. Radial Test Case
2. Meshed Test Case

Both the radial and the meshed test case can be defined with one or more substations, and with or without distributed generation.

1.3.1. Radial Test Case

The radial test case is based on an IEEE distribution test system with 34 nodes and one feed-in source. The system was created in 1994 and is based on an actual distribution network in Arizona with a nominal voltage of 24.9 kV. The system is a long and lightly loaded radial system.

We have adapted the system to fit our purpose, adding distributed generation, and adjusting loads in the system. The layout of the radial system is found in Figure 1.4

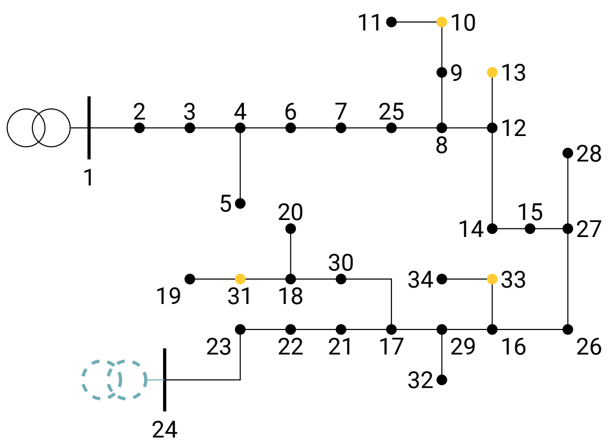


Figure 1.4.: *IEEE radial test case with one or two substations*

The radial test case has been studied for three different configurations, all listed below.

- One substation at node 1
- Two substations (node 1 and 24)
- Two substations (node 1 and 24) and distributed generation (node 10, 13, 31 and 33)

For the configurations with two substations, the radial grid holds similar properties to the meshed grid with respect to the solution procedures discussed in the remaining chapters of this report.

1.3.2. Meshed Test Case

The meshed test case consists of 22 nodes. It is based on a CIGRE medium voltage distribution network with 15 nodes, two substations and distributed generation. It has been adapted to fit our purpose, in part by adding 7 new nodes. As with the radial test case, the meshed test case is defined for three different configurations.

- One substation at node 1
- Two substations (node 1 and 12)
- Two substations (node 1 and 12) and distributed generation (node 3, 4, 5, 6, 7, 8 and 9)

Figure 1.5 provides a schematic description of the meshed test case.

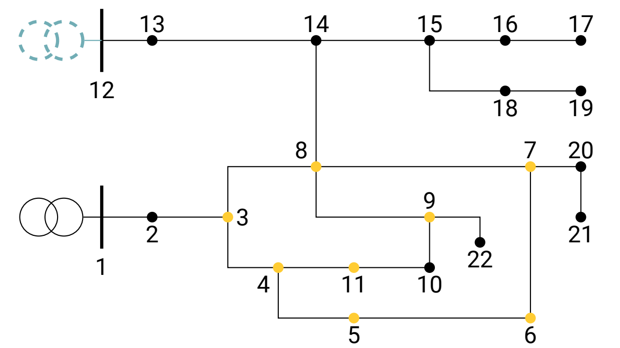


Figure 1.5.: *Meshed medium voltage distribution system test case from the Cigre library*

In Section 2.7 we discuss the results from the meshed grid with substation at node 1.

1.4. Mathematical formulation

The most relevant parameters and variables of the power distance problem are explained in the following lists. At the end of the section, we formulate the problem itself.

Sets

We consider the following sets

N	Set of all nodes, indexed i or j
N^S	Subset of substations, indexed j
E	Set of all lines
E_j^+	Subset of all import lines at node j
E_j^-	Subset of all export lines at node j

Note: we require the flows on lines to be positive. Hence, we have two directed lines between connected nodes i and j, one representing flow from i to j and the other representing flow from j to i. For an example network of three nodes, we have the set of lines E to be

$$E = \{p_{12}, p_{21}, p_{13}, p_{31}, p_{23}, p_{32}\} \quad . \tag{1.1}$$

The subset of import lines at node 2, E_2^+, is $\{p_{12}, p_{32}\}$, and the subset of export lines from node 2, E_2^-, is $\{p_{21}, p_{23}\}$. All lines appear once in E and once in exactly two of the subsets. For example, the line p_{12} is in E, E_1^- and E_2^+.

Parameters

We consider the following parameters

D_j	Demand in node j
G_j	Generation in node j
L_e	Length of line e
	is a line between i and j
$\overline{S}_j$	Upper limit on power through substation
$\underline{S}_j$	Lower limit on power through substation

Variables

We consider the following variables

$p_e \in \mathbb{R}_+$	Power flow on line e
$s_j \in \mathbb{R}$	Power to/from substation j

Objective function

The objective is to minimise the cost of power transfer, under the assumption that the marginal cost decreases ($\alpha < 1$). In addition, the direction of the flow does not affect the cost, hence the cost function uses the absolute value of the flow, denoted by $|\cdot|$. The cost function is given by

$$P_d = \min_{p_e} \sum_{e \in E} L_e \, |p_e|^{\alpha} \tag{1.2}$$

The absolute operator can be dropped if one poses the condition that $p_e \geq 0$, which we will do in the following, see also (1.5).

The reader is reminded that this cost function is non-convex under the assumption that $0 \leq \alpha < 1$.

Constraints

At each node, the sum of power fed into the node must match the local demand, generation, or supply from a substation

$$\sum_{i \in E_j^+} p_i - \sum_{e \in E_j^-} p_e + s_j = D_j - G_j \qquad \forall j \in N \tag{1.3}$$

The first term in the sum denotes the (positive) flows into the node, the second term the (positive) flows out of the node. While it would be more elegant to allow negative flows, the requirement in the cost function that p_e is positive makes this formulation necessary.

In addition, there are some basic constraints on the variables

$$\underline{S}_j \leq s_j \leq \overline{S}_j \qquad \forall j \in N \tag{1.4}$$

$$p_e \geq 0 \qquad \forall e \in E \tag{1.5}$$

Constraint (1.5) ensures that all flows are positive, which is essential for the cost function. In theory, this allows flows in both directions on the same line, however, the cost imposed on flows prevents this from happening.

In any node that is not a substation, $\underline{S}_j$ and $\overline{S}_j$ is equal to zero, and s_j will not take a value.

Feasibility

The current formulation implies the relationship

$$\sum_{j \in N} S_j = \sum_{j \in N} (D_j - G_j) \quad ,$$

stating that total energy supplied by substations is equal to the total net demand. If the substation capacity, limited by (1.4), and the distributed generation in the system is insufficient to cover demand, the problem has no solution. The interpretation is that some energy can not be supplied. To allow for systems with too little capacity in certain hours, the problem could be redefined by adding a slack-variable to Constraint (1.3), associated with a high cost in the objective function. This cost could be interpreted as representing *kvalitetsjusterte inntekts-rammer ved ikke levert energi* (cost of energy not supplied, KILE) costs incurred by a concessionary failing to supply sufficient energy.

2. Reformulation and solution approaches

This chapter describes different solution approaches for the power distance problem stated in Section 1.4. Calculating the minimum power distance for a given, meshed topology is a non-convex problem, and only a brute-force approach testing all possible solutions can guarantee a global optimum. Methods for finding a local optimum perform well in our test systems, but can not guarantee global optimality.

We begin by providing an overview of an overall procedure that can be applied to all solution approaches, proceed by describing the individual steps, and end by giving examples and results highlighting the issues discussed throughout the chapter.

2.1. Overall algorithm

Figure 2.1 provides a high level overview of the overall solution algorithm applied to the power distance problem described in the previous chapter. In this chapter, the different steps in the solution algorithm is described in more detail.

2.2. Pre-processing to reduce complexity

Grid configurations found in practical applications may consist of a significant number of nodes. This can be challenging for any solution approach. However, the properties of the problem and of real world networks allow for some simplification.

The power distance can be directly computed for any radial grid. Challenges concerning the non-convexity of the problem only arise when the grid is meshed. Hence, to reduce the problem size, in a pre-processing stage all radial feeders can be separated from the problem, and solved independently. The steps of the pre-processing procedure is highlighted by a red, dotted box in Figure 2.1, and the algorithm for identifying radial feeders is summarised as a simplified pseudo-code in Al-

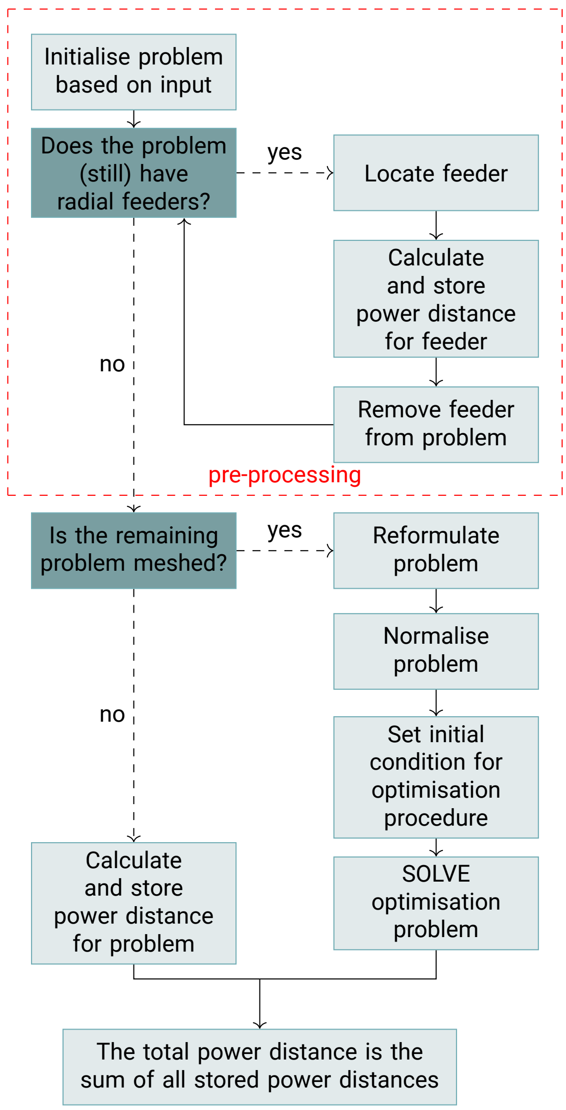


Figure 2.1.: *Flow chart of the overall solution algorithm*

gorithm 1.

Algorithm 1 Identify radial feeders

1: **procedure** IDENTIFY_FEEDERS(L, i)
2: $n \leftarrow 0$
3: $A^0 \leftarrow A$
4: $D \leftarrow \mathcal{D}$
5: **for** $i \leftarrow 1..N$ **do**
6: **if** $d_i = 1\ \&\ \overline{S}_i = 0$ **then** ▷ This is a leaf
7: $k \leftarrow i$
8: j is neighbour of k
9: $n \leftarrow n + 1$ ▷ New subgraph
10: add line k, j to A^n
11: remove line k, j from A^0
12: $d_i \leftarrow 0$
13: **while** $d_j = 2\ \&\ \overline{S}_j = 0$ **do**
14: $d_j \leftarrow 0$
15: $k \leftarrow j$
16: $j \leftarrow$ neighbour of k
17: add line k, j to A^n
18: remove line k, j from A^0
19: **end while**
20: $d_j \leftarrow d_j - 1$
21: Add net demand of A^n to node j
22: **end if**
23: **end for**
24: **return** $A^1, .., A^n$
25: **end procedure**

First, we need a matrix A that has positive elements if there exists a connection between two nodes. A is a logical and symmetric matrix of size NxN, where N is the number of nodes in the grid, and can be created based on the input to the optimisation problem. If there exists a line between node j and i, elements A_{ij} and A_{ji} has the value 1. If there is no line between the two nodes, the elements hold the value 0.

To locate the radial feeders, we need to search for nodes that are at the end of a feeder. To identify these nodes, we start by applying the definition of the Laplacian matrix to A,

$$\mathcal{L} = \mathcal{D} - \mathcal{A} \quad ,$$

Where $\mathcal{D}$ is the degree matrix, and $\mathcal{A}$ is the adjacency matrix here represented by A. The mathematical relationship is used to identify subgraphs. In the adjacency matrix, each element indicates if a pair of nodes is adjacent or not in the graph. The adjacency matrix in this problem definition is a simple graph[1], i.e. the adjacency matrix is a symmetric (0,1)-matrix with zeros on the diagonal. The degree matrix $\mathcal{D}$ is a diagonal matrix which contains information about the degree of each node. The degree is the number of edges connected to one node. I.e. if node i is connected to both node j and k, the degree of node i is 2. d_i refers to the diagonal element i of the degree matrix $\mathcal{D}$. For a graph with 5 nodes as defined in Figure 2.2, we get the following Laplacian matrix, degree matrix, and adjacency matrix.

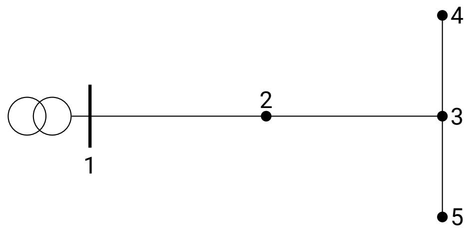


Figure 2.2.: *Simple graph with 5 nodes*

$$\mathcal{L} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

If a node has degree $d_i = 1$ and no substation, it is a leaf and is at the end of a feeder (line 6 in Algorithm 1). The algorithm proceeds along the

[1] undirected and without multiple edges or loops

feeder until a node with more than two neighbours ($d_i > 2$) is found. The identified nodes can be separated as a subgraph representing a radial end of the original graph. This algorithm is applied to each leaf, and all radial feeders are identified and removed from the original graph. The returned graph A^0 holds all substations, and a potentially meshed grid connecting them. The subgraphs A^1 to A^n are purely radial grids, for which the power distance can be computed directly.

Figure 2.3 highlight the radial feeders identified in the meshed test case with one substation. The radial feeders are marked by red or yellow dashed lines. The algorithm moves from node number 1 to 2 and so on, and identifies the first radial feeder 12-13-14. The feeder is removed from the problem, and the algorithm proceeds to feeder 17-16-15. The next radial feeder that is identified is 19-18-15-14-8. At last, the radial feeders 21-20-7 and 22-9 are removed from the problem.

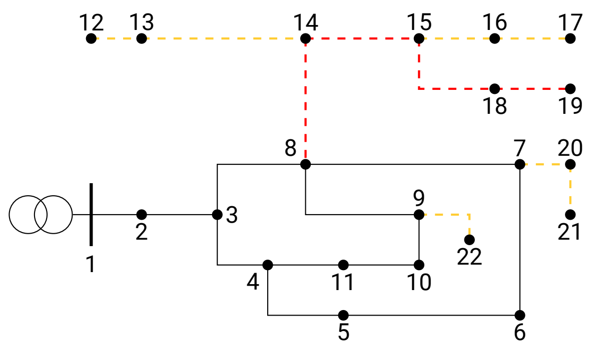


Figure 2.3.: *Simple graph with 5 nodes*

After the pre-processing procedure of removing radial feeders, the remaining part of the grid in Figure 2.3 is marked by black lines. It is still a meshed grid, but with a reduced size.

Further pre-processing

For the potentially meshed graph A^0, further pre-processing steps can be applied.

1. The graph may still hold nodes with degree 2, but without any demand or generation. These nodes add additional states to the optimisation problem, but will not affect the final result. Hence, these nodes can be removed from the graph.

2. There may be subgraphs connected by only one node. These subgraphs can be separated from the main problem, as the flow through this so called articulation point is fixed. Tarjan's algorithm [2] can be used to identify articulation points.

Operators of the distribution grid usually have the option to open circuit breakers in order to remove unnecessary loops in the grid. This is done for a number of operational reasons. Using this information would allow to further simplify networks, and even to reduce the number of cycles or meshes that are creating the challenges in computing the power distance. However, there are challenges with this approach. First, data about circuit breaker settings is usually not available. Second, it would allow the grid operator to increase the power distance defining his output by setting the breakers into specific positions.

2.3. Reformulation

The problem described in Section 1.4, and specifically the cost function in (1.2) constitute a geometric problem. Geometric problems are not generally convex, but reformulations of the cost function and constraints allow some geometric problems to be transformed into convex problems[1].

The basic idea is to apply a change of variables by taking the logarithm of the variable that has an exponent. Using $\tilde{p} = \log p$ and $\tilde{L} = \log L$ we have

$$P_d = \min \sum_{e \in E} \mathrm{e}^{\tilde{p}_e \alpha + \tilde{L}_e} \quad , \tag{2.1a}$$

[2] Tarjan's Algorithm: An algorithm in graph theory for finding strongly connected components.

s.t. $\forall j \in N$

$$\sum_{i \in E_j^+} e^{\tilde{p}_i} - \sum_{e \in E_j^-} e^{\tilde{p}_e} + s_j = D_j - G_j \tag{2.1b}$$

$$\underline{S} \le s_j \le \overline{S} \tag{2.1c}$$

Note that $\tilde{p}_e$ is unbounded in $\mathbb{R}$. This implies, that p_e is strictly positive as the logarithm is only defined for positive arguments. The change of variables therefore includes a different constraint on p_e than (1.5), namely $p_e > 0$.

The canonical reformulation proceeds by taking the logarithm of this function

$$\tilde{P}_d = \min \log \left(\sum_{\forall e \in E} e^{\tilde{p}_e \alpha + \tilde{L}_e} \right) \quad , \tag{2.2}$$

but in our case this is not possible for the constraints, as the constraints may take a non-positive value and the logarithm is not defined for arguments that are not strictly positive.

The issue arises as we violate one of the conditions for the reformulation, namely that all terms[3] in both the cost function and the constraints have positive coefficients, see (4.41) in [1]. However, as we need to allow for flows in both directions on any line, we either have negative coefficients or negative power flows – either of which violates a pre-condition for the reformulation.

The reformulation does provide a convex cost function, and hence allows application of non-linear solution approaches. However, even after the reformulation the problem is non-convex in the constraints, meaning that there are local optima. Finding the global optimum is np-hard or np-complete, and the solution time grows exponentially with the number of decision variables. In our case the number of decision variables equals the number of lines in the grid.

[3]more precisely, all monomials constituting the geometric problem

2.4. Normalisation

The problem consists of exponential functions. If the arguments $\tilde{p}$ and $\tilde{L}$ are large, the exponential terms may become very large. The numerical algorithms could become inaccurate, or it may not be possible to even represent the terms on a computer. For example, the largest 8-Byte floating point number is typically 1.7×10^{308}. If $(\alpha\, \tilde{p} + \tilde{L})$ is larger than 709, this value would be exceeded. It is therefore recommended to normalise the parameters before solving the problem, using a normalisation factor for power, $\bar{p}$, and for distance, $\bar{L}$.

$$p'_e = p_e/\bar{p} \quad , \qquad s'_j = s_j/\bar{p} \quad , \tag{2.3}$$

$$D'_j = p_e/\bar{p} \quad , \qquad G'_j = G_j/\bar{p} \quad , \tag{2.4}$$

$$L'_e = L_e/\bar{L} \quad , \tag{2.5}$$

where

$$\bar{p} = \max_j D_j \quad , \qquad \bar{L} = \max_e L_e \quad . \tag{2.6}$$

We have chosen the maximum demand at any node and the longest line as normalisation factors, but other factors could be used as well. The transformed variables $\tilde{p}$ and $\tilde{L}$ are similarly normalised

$$\tilde{p}'_e = \log\left(\frac{p_e}{\bar{p}}\right) \quad , \qquad \tilde{L}' = \log\left(\frac{p_e}{\bar{p}}\right) \quad . \tag{2.7}$$

Finally, the power distance is formulated as

$$P_d = \bar{L}\, \bar{p}^{\alpha} \sum_{e \in E} e^{\tilde{p}'_e \alpha + \tilde{L}'_e} \quad , \tag{2.8}$$

The constraints simply use the scaled variables instead of the original variables, i.e.,

$$\sum_{i \in E_j^+} e^{\tilde{p}'_i} - \sum_{e \in E_j^-} e^{\tilde{p}'_e} + s'_j = D'_j - G'_j \tag{2.9}$$

In the remainder of the report we use the original variables to simplify notation. In all test runs, we have scaled the inputs according to the equations above.

2.5. Initial condition and local optima

The problem, as already stated, is inherently non-convex. A non-convex problem will have several locally optimal solutions, but just one globally optimal solution.

2.5.1. Local versus global optima

A simple illustration of local versus global optima for a minimisation problem can be found in Figure 2.4. In Figure 2.4 we have 4 local optima, of which only one of them is the global optimum (number 4). A mathematical solver will check if it can find a better solution by moving in any direction from the current solution (increasing or decreasing the value of x). In a local optimum, the solver will not find a better solution in any direction, and it has no way of knowing if it has found the global optimum.

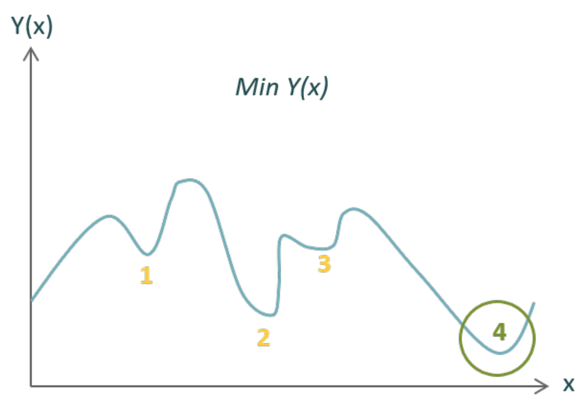


Figure 2.4.: *Local versus global optima for a minimisation problem*

Imagine that you are standing at the bottom of a valley equal to local optimum number 3 in Figure 2.4. You can only see hillsides in all directions, and you can conclude that you are standing at the lowest point in the area. However, there could exist a lower point behind the hills. You could move towards optimum number 2, but you are not guaranteed to find a lower point. Even if you are standing at optimum number 4, which is the global optimum, you have no way of knowing that you are at the lowest point.

Non-convex problems are defined as NP-hard or NP-complete, and there is no known way to find a solution quickly. The only way of knowing if you have found the global optimum for a non-convex optimisation problem is to check every single solution, and the time required to solve the problem increases rapidly as the size of the problem grows.

For problems of the size studied in this report, it is possible to find the globally optimal solution by searching through the entire solution space. However, the problem has bad scalability, and it could be challenging, if not impossible, to find and guarantee the globally optimal power distance for realistically sized problems.

In Section 2.6 we present three ways of solving the electric power distance problem. Two of these approaches can only guarantee a local optimum, but they are expected to be much faster than the brute force approach (approach 3) where we search through the entire solution space. There are however ways to increase the probability that the local optimum found by a solver is a quite good optimum.

2.5.2. The importance of the initial condition

All solvers start with a feasible solution to the problem as an initial condition from where it searches for a better solution. Such an initial condition can be defined by the user, and it could be designed based on known properties of the problem. The initial condition set the basis for the search performed by the solver, and if the initial condition is far from the optimal solution it could end up at a poor local optimum.

In Figure 2.5 there are two red crosses representing different initial conditions for a solver searching for the optimal solution. Starting in cross number 1, the solver will search for a better solution in the direction of local optimum number 1. Once

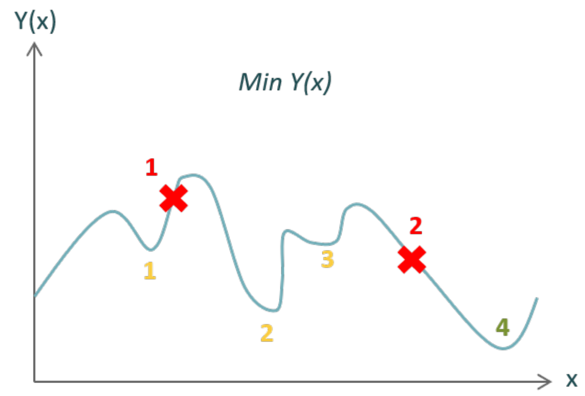


Figure 2.5.: *Different initial conditions*

it reaches the local optimum, it will stop searching even though there are in fact two better solutions to the problem. If we are able to provide the solver with an initial condition starting in cross number 2, it will find the globally optimal solution! All though it could be possible to design initial conditions that provide some certainty that we have reached a good solution, there is no guarantee that this solution is in fact the global optimum.

2.5.3. Designing the initial condition

There are several approaches that could be applied when designing the initial condition. We will test the following approaches:

Power flow Computing the power flow is a problem with well-known solution approaches. The flows from a power flow can be directly used to initialise the decision variables in the power distance problem. In the following, this is referred to as "power flow" initial condition.

Flat power flow Alternatively, one can simply assign the same flow to all lines. As we have separate decision variables for flows in both directions, using such an approach will lead to a zero net flow. We test both a high and a low value for the flat flows. We call these "high flat flow" and "low flat flow".

Flows based on local demand The fourth initial condition is similar to the flat flows. However, we use the average demand of the start and end node of each line as a level for the initial flow. Again, as we set the flow in both directions to the same value, the net flow is zero. We call this condition the "local flow" condition.

Of the four alternative initial conditions, only the power flow condition is a feasible solution. As all other initial conditions have net zero flow on the lines, they cannot cover the demand at the beginning. However, we will see that all four initial conditions lead to feasible solutions. We also tested initial conditions with directed flows (different values per direction), but these were very unstable and we have excluded them from the report.

Results for the different initial conditions are shown in Section 2.7.3.

2.6. Solution approaches

In the following we describe three ways of solving problem (2.1). The first two approaches attempt to find a local optimum based on a good initial condition, while the third is a brute force approach for finding the global optimum.

2.6.1. Approach 1: Approximation with a Sequential Quadratic Program

Traditionally, solvers for Linear Programs (LPs) and Quadratic Programs (QPs) were available, while general nonlinear problems could not directly be solved. The Sequential Quadratic Programming (SQP) approach is a typical method to tackle nonlinear problems by approximating them with a QP, and repeatedly solving until hopefully one converges to a solution. Using SQP allows us to use any conventional QP-solver.

The SQP approach

The cost function of a nonlinear problem can be approximated by a quadratic cost function, and the constraints can be linearised around a starting condition, yielding a QP. The solution to this QP is closer to a local minimum than the starting condition. Repeatedly applying the QP-reformulation will make the solution converge to a minimum.

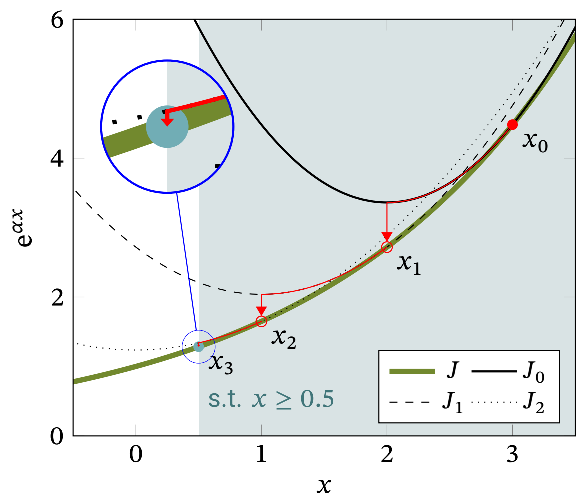


Figure 2.6.: *Simplified depiction of the SQP method for a 1-dimensional exponential cost function. Starting at x_0, repeated quadratisation and minimisation leads to x_3 – the minimal cost within the feasible set.*

The SQP approach is sketched in Figure 2.6. The green line is the cost function J, x_0 the initial starting condition, and the solution is constrained by $x \geq 0.5$. The quadratic approximation of the cost function at x_0 leads to J_0. By minimising J_0, here highlighted by the first red line, one arrives at the minimum of J_0, and with this at x_1. The approach is now repeated for J_1, J_2 – until one arrives at x_3. Any further search will be bounded by the constraint.

The general form of a quadratic cost function J is

$$J_k = \Delta \mathbf{x}_k^\intercal \mathbf{Q} \Delta \mathbf{x}_k + \Delta \mathbf{x}_k^\intercal \mathbf{R} \quad , \tag{2.10}$$

where $\Delta \mathbf{x}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ is the distance from the current solution. In our case, $\mathbf{x}$ are the power flows $\tilde{p}$ across each line, and the current solution is the result of the previous iteration,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x} \quad . \tag{2.11}$$

Of course, this leaves the challenge of choosing the initial starting condition x_0. Also, the user needs to define a stop condition. Usually this is a fixed number of iterations and a condition on the change in the cost between two iterations. If the change in cost is lower than a small bound ϵ, the iteration is stopped even before the maximum number of iterations is reached.

SQP for the power distance

The matrices $R \in \mathbb{R}^n$ and $Q \in \mathbb{R}^{n \times n}$ are the first and second partial derivatives of the original cost function (2.2). The linear terms R_i are computed for each iteration k by

$$R_{i,k} = \frac{\partial}{\partial \tilde{p}_i} \left(\sum_{e \in E} \mathrm{e}^{\tilde{p}_e \alpha + \tilde{L}_e} \right) \Bigg|_{\tilde{p}_{i,k}} = \alpha\, \mathrm{e}^{\tilde{p}_{i,k} \alpha + \tilde{L}_i} \,. \tag{2.12}$$

Note that $\tilde{p}_{i,k}$ is the result of the previous iteration, and hence a fixed parameter. Similarly, we can compute the quadratic cost factors Q

$$Q_{ij,k} = \frac{\partial}{\partial \tilde{p}_j} R_{i,k} = \begin{cases} 0 & \text{if } j \neq i \,, \\ \alpha^2\, \mathrm{e}^{\tilde{p}_{i,k} \alpha + \tilde{L}_i} & \text{if } j = i \,. \end{cases} \tag{2.13}$$

The off-diagonal terms vanish as R_i is not a function of $\tilde{p}_j$ with $i \neq j$. This can be interpreted as that there is no impact of the flow on one line on the cost of another line.

Additionally, the constraints need to be linearised. Again, linearisation uses the partial differential, evaluated at the current operating point x_k. The general form is

$$f(x_k + \Delta x) \approx \Delta x \frac{\partial}{\partial x} f(x) \Big|_{x = x_k} + f(x_k) \,, \tag{2.14}$$

where the decision variable x_{k+1} enters the constraint via $\Delta x = x_{k+1} - x_k$. Constraint (2.1b) in its linearised form is given by

$$\sum_{i \in E_j^+} (\Delta \tilde{p}_{i,k} + 1)\, \mathrm{e}^{\tilde{p}_{i,k}} - \sum_{e \in E_j^-} (\Delta \tilde{p}_{e,k} + 1)\, \mathrm{e}^{\tilde{p}_{e,k}} + s_j = D_j - G_j \quad , \tag{2.15}$$

with $\Delta\tilde{p}_{i,k} = \tilde{p}_{i,k+1} - \tilde{p}_{i,k}$ consisting of the decision variable $\tilde{p}_{i,k+1}$ and the fixed parameter $\tilde{p}_{i,k}$.

2.6.2. Approach 2: Solving with non-linear solver

In recent years, solvers that can efficiently and reliably solve non-linear problems directly have been developed. The reformulated problem (2.1) can be readily transferred to such a solver. One such solver is IPOPT [2], which we will use in the following.

Similar to the SQP-approach, a starting condition for the non-linear solver needs to be set. However, as non-linear solvers can directly use the differentials of functions rather than the linearised version, the solution should be more robust and efficient.

2.6.3. Approach 3: brute-force iteration over reduced networks

Due to the inherent non-convex nature of the problem, we are faced with the challenge of finding the global optimum. Approach1 and 2 are methods for finding a local optimum, and the result depends on the initial condition.

Finding the global optimum is a np-problem, meaning that the computation time grows more than polynomial with the number of variables – in our case the number of power lines. To illustrate why this is not simply a question of computation power, take this example: assume that a problem with ten lines takes one hour to solve, and the computation time grows exponentially. Doubling the number of lines to 20 would increase the computation time to 42 days, adding another ten lines to 119 years, and solving over 53 lines would lead to a computation time of roughly one billion years.

Nevertheless, for the small systems studied in this report, the global optimum can be found. This provides an indication on which starting conditions might be useful, or how far the solutions found by Approach 1 and 2 are from the global optimum in these specific cases.

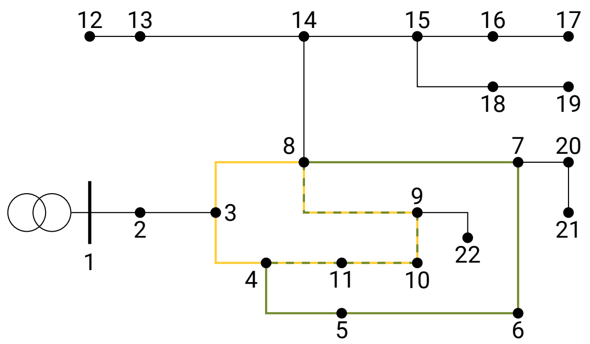


Figure 2.7.: *The meshed system with two simple cycles, marked green and yellow*

Approach for finding the global optimum

We use the fact that each optimal solution – local and global alike – has a tree topology. This means, that any solutions returns a radial topology without cycles. If there are several substations, all possible solutions will consists of one tree per substation, and these trees are not connected.

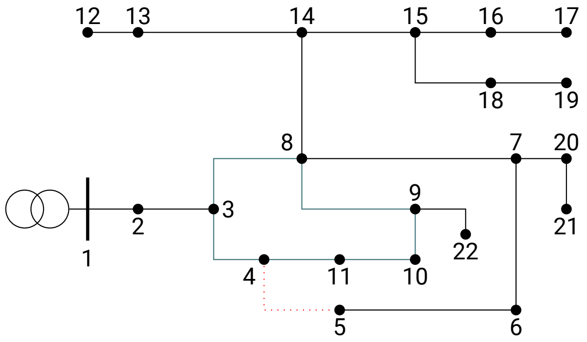
We hence test all possible combinations of removing lines from the topology that fulfil these criteria:

- Each node is connected to exactly one substation
- The substations are no longer connected to each other. This splits the graph into a number of components that is equal to the number of substations
- Every component in the graph (one per substation) is a tree

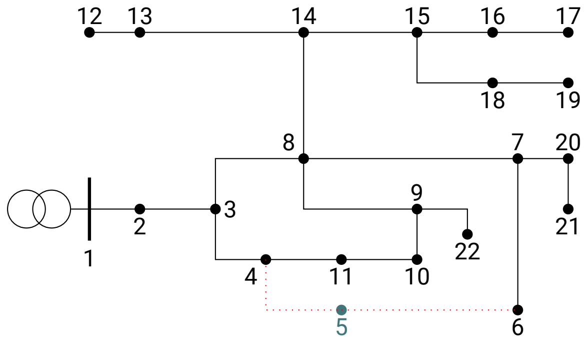
We can then compute the power distance for each combination of removing lines, and selecting the lowest power distance gives us the global optimum.

Algorithm Assuming only one substation, we follow steps 1 to 3:

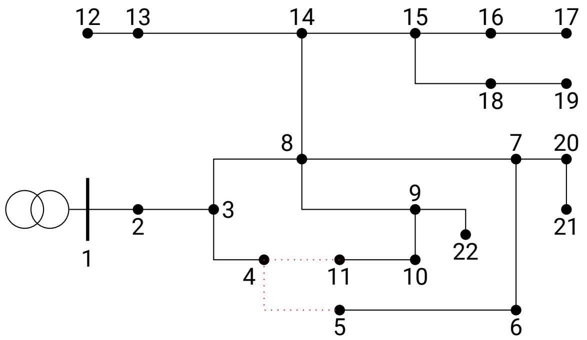
1. Find all lines that are part of a cycle. Lines that are not part of a cycle are called bridges,



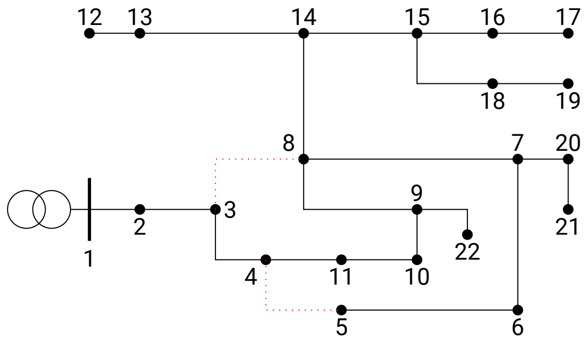
(a) *Iteration 1: removing line 4–5. Still one remaining cycle (3–8–9–10–11–4–3).*



(b) *Iteration 2: removing lines 4–5, 5–6. Node 5 not connected.*



(c) *Iteration 3: removing line 4–5, 4–11. Scaled power distance: 4.876 (with α of 0.9).*



(d) *Iteration 4: removing lines 4–5, 3–8. Scaled power distance: 5.227 (with α of 0.9).*

Figure 2.8.: *Four iterations in the brute force approach*

as removing these lines leads to the graph not being connected

2. Iterate over all combinations of removing lines that are not bridges from the graph
   - If the new graph is not connected, ignore this combination
   - If the new graph is not radial, continue to remove lines
   - If the new graph is radial and connected, store this combination and compute the power distance
3. The lowest power distance found is the global optimum

Example The first step, finding cycles, can be illustrated with Figure 2.7. Here, all lines that are part of a cycle are coloured, and all bridges are black. Removing any of the bridges would lead to one or more nodes no longer being connected to the substation. Algorithms for finding cycles exist, but generally it is difficult to identify cycles. It is easier to find all bridges. For this, a variation of Tarjan's algorithm can be used.[4] Tarjan's algorithm has complexity $\mathcal{O}(V + E)$,[5] meaning that the computation time grows only linearly with the number of nodes V and and edges E.

In the second step, we iterate over any combination of removing lines that are not bridges. Figure 2.8 shows a number of these iterations. The

[4] https://en.wikipedia.org/wiki/Bridge_(graph_theory), https://www.geeksforgeeks.org/bridge-in-a-graph/

[5] The Big-O notation $\mathcal{O}(\cdot)$ is a measure for computational complexity. It means, that the computation time does not grow faster than the argument of $\mathcal{O}$ multiplied by a constant. E.g., if your data set has n elements, and you run an alogrithm with $\mathcal{O}(n^2)$, doubling the number of data points will lead to a computation time four times longer.

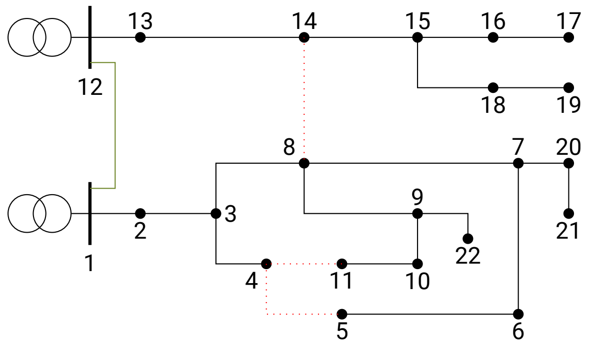


Figure 2.9.: *Ensuring radial topology with multiple substations: Using the additional line 1–12, we can apply the same algorithms and iteration as for the single-substation case (while ignoring line 1–12 in the iteration). Removing lines 4–5, 4–11, 8–14, and dropping the additional line 1–12 leads to two components with one substation each.*

first iteration does not remove all cycles, so we need to remove an additional line. The second iteration leads to a disconnected node, and we discard it. Both checking if the graph is radial and if it is connected can be done by using Depth-First Search (DFS) – effectively traversing the graph and checking if all nodes are reached (then it is connected), and if any node is reached twice (then it has a cycle). DFS has complexity $\mathcal{O}(V + E)$. The third and fourth iteration are both connected and do not have cycles – so we compute the power distance. As we already have a radial system and simply need to add the power distances for each part of the line, we avoid any optimisation for P_d. The complexity of finding P_d in a radial system is also $\mathcal{O}(V + E)$. Finally, of these four iterations, the third has the lowest power distance and we use this as our optimal solution until a better solution is found in another iteration.

Extension to multiple substations In the case of multiple substations, we need to ensure that the network is split in such a way that each node is connected to exactly one substation. At first, this seems complex, as we would need to split the graph into a number of components equal to the number of substation, while at the same time ensuring that exactly one substation remains in each component. However, we can apply a trick to avoid the issue: we can connect all substations to the first substation, but ignore these additional connections in the iterations. As we ensure the network to be radial and connected, these additional connections between the substations automatically lead to the desired topology, see also Figure 2.9.

Scalability

Most operations described above are on the order of $\mathcal{O}(V + E)$, meaning that the computation time grows linearly with the number of nodes V and edges E. This is true for finding all bridges, and for checking if the graph is connected and radial.

However, iterating over all combinations is a combinatorial problem. The number of combinations that need to be checked is on the order of $\mathcal{O}(E^C)$, with number of edges E and number of cycles C. As both the complexity of an iteration grows with $\mathcal{O}(V + E)$ and the number of iteration increases with $\mathcal{O}(E^C)$, the total computational complexity grows with $\mathcal{O}((V + E)E^C)$, or roughly $\mathcal{O}(E^{C+1})$.

We can further tighten this bound by using that not all combinations of removed lines lead to a connected graph. Assuming a problem with two cycles, only combinations of one line from one cycle and one line from the other cycle are valid. Hence we can give an upper bound for the complexity on the order of $\mathcal{O}(E_1E_2)$, with E_1 the number of lines in cycle one, and E_2 the number of cycles in cycle two. Generalised, this can be written as $\mathcal{O}(\prod_i E_i)$, which is always tighter than $\mathcal{O}(E^C)$.

Considering larger problems, the scalability issue lies more with the complexity of the topology in terms of the number of cycles, than with the number of total line elements or line elements per cycle. For larger networks, it is not unreasonable to expect a large number of cycles.[6]

[6] The upper limit for the number of cycles in a graph grows exponentially with the number of nodes – consider the case of a network where every node is connected to every other node. However, such topologies would not be realistic in power systems.

2.6.4. Other solution approaches

To circumvent a full solution of the problem but at the same time increase the chance of finding the global optimum, some alternative approaches can be applied. These however have their own drawbacks, and we have not implemented and tested them.

Genetic algorithms This class of algorithms attempts to mimic evolution. From any starting point, a number of similar solutions are generated, and the best ones are kept for further searches. Depending on whether there was an improvement, the step size or mutation rate may be adjusted. This should enable the algorithm to leave a local optimum if the cost function is sufficiently well behaved.

On the downside, this kind of algorithm does neither have a deterministic outcome nor deterministic computation time, even with a given starting condition. Also, genetic algorithms do not provide any guarantee on the optimality of the result. The increased uncertainty related to the solution is making genetic algorithm less appropriate for regulatory purposes.

Heuristics Knowledge of the problem can be used to reduce the search space, by excluding inefficient solutions from the start or by using experience to guess good solutions. As the problem at hand is quite well specified, it may be possible to find such heuristics.

Splitting the problem into sub-problems As the time to find the global optimum grows exponentially, any reduction in problem size alleviates the issue. We propose some preprocessing steps in Section 2.2. However, there is no guarantee that all networks can be sufficiently reduced to allow for a computation of the global optimum.

2.7. Results

In this section, we present results from applying the algorithms described above to **the meshed test case with one substation**. We selected results in order to highlight the challenges we encountered during the test runs, and have collected the results in three groups. First, we show an example for the non-convexity of the problem, then we discuss Approach 3 for finding the global optimum, and finally we investigate Approaches 1 and 2 which yield local optima.

2.7.1. Example for the non-convexity of the problem

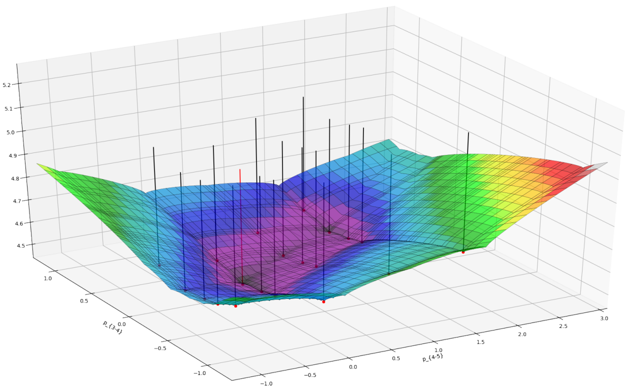
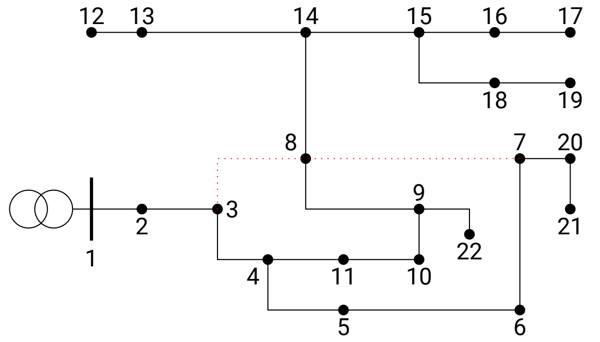


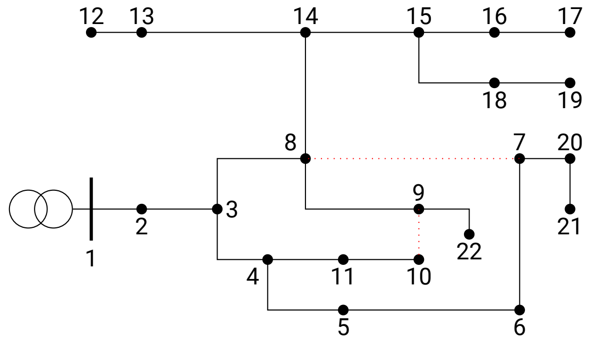
Figure 2.10.: *Non-convexity of the problem: by fixing the flows in the meshed system on lines 3–4 and 4–5, all other flows are determined (the problem has two cycles and hence two degrees of freedom). The plot shows the power distance, which is clearly non-convex as shown by the valleys covering the plane. The black pins identify local optima, the red pin the global optimum.*

As discussed, the problem stays non-convex even after the reformulation. We show this using an example where we fix flows on two lines and report how the cost changes. As there are two cycles, the problem has two degrees of freedom and fixing the right two flows fully determines the power distance. This allows to show the power distance as a function of these two flows.

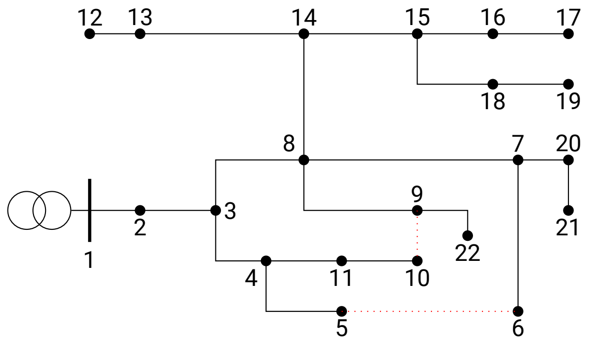
Specifically, we adjust the flows on lines 3–4



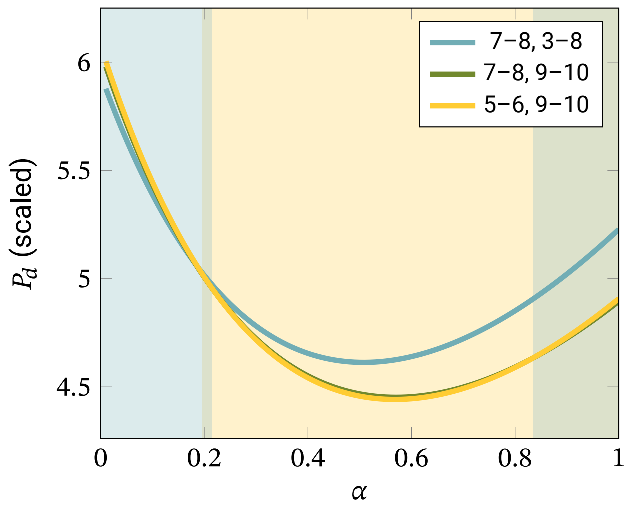
(a) *Removing lines 3–8 and 7–8. This minimises the total grid length (optimal solution for small α).*



(b) *Removing lines 7–8 and 9–10, optimal for large α and α around 0.2.*



(c) *Removing lines 5–6 and 9–10- Optimal for α between 0.21 and 0.83.*



(d) *Global optimum as a function of α. The background colour indicates which solution is optimal.*

Figure 2.11.: *Dependence of the globally optimal solution on α*

and 4–5. Starting from this, all other flows are fixed by the requirement that the inflows and outflows equal local demand and generation.

Figure 2.10 shows how the power distance varies with the flwos on lines 3–4 and 4–5. The axes are normalised and computed as difference to the globally optimal flow. I.e., the global optimum is at (0,0). At (1,1), both flows are twice as large as in the global optimum, etc.

We see a number of valleys crisscrossing the power distance plane. Where two valleys meet, we usually find a local minimum. For example, there are several minima along the line where p_{4-5} is zero (that is, where the flow on 4–5 is the same as in the gloabl optimum). Each of these minima along this valley represents another flow distribution in the remaining lines that is locally optimal. This highlights how difficult it is to find the global optimum: even if the optimal flow on 4–5 would be known, a searching algorithm still might get stuck in a local optimum.

When adjusting α (not shown here), another observation can be made: the valleys become deeper and more pronounced for small α, while for $\alpha \to 1$ the surface becomes smoother and finally converges against a linearly constrained plane. This visualises our initial statement that the problem is linear for $\alpha = 1$.

2.7.2. Computing the global optimum

Depending on the parameter α, different combinations of removing lines lead to the global optimum. For the meshed case with one substation, we need to remove two lines in each iteration, and we find three combinations that are optimal for different values of α, see Figure 2.11. Two of the combinations ({7–8, 9–10} and {5–6, 9–10}) are very close to each other, while the third ({7–8, 3–8}) has higher costs in most cases but lower costs for low α. The third combination is the one that is minimising the total line length, while the other two appear to be more optimal considering the given demand distribution.

Note: while it appears that the power distance is smallest for an α around 0.5, this is an effect of the normalisation we described in Section 2.4. The full power distance is multiplied by $\bar{p}^{\alpha}$, and hence increases monotonously for increasing α.

Scaling of the brute force approach

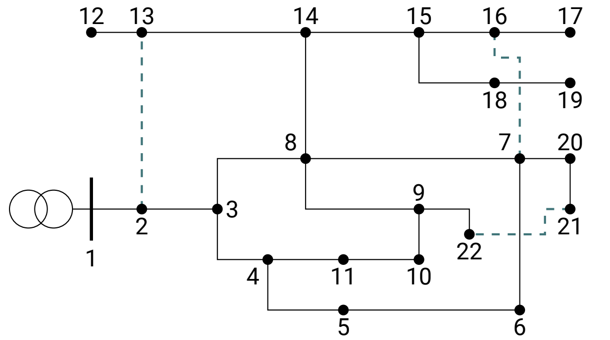
We have defined a theoretical upper bound on how the complexity of finding the global optimum scales with additional cycles in the grid. In this section, we show some results for the meshed system, and how the computation time increases as we add additional cycles.

Table 2.1.: *Different parameters indicating the scaling with additional lines in the meshed system.*

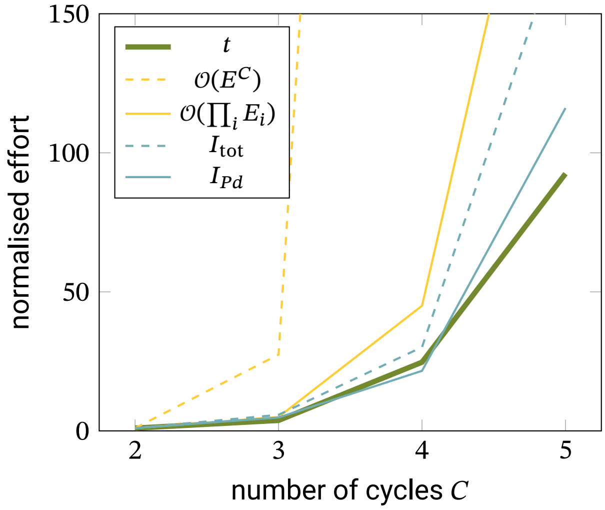
C	E	$\mathcal{O}(E^C)$	$\mathcal{O}(\prod_i E_i)$	t [s]	I_{tot}	I_{Pd}
2	10	100	48	0.044	42	32
3	14	2744	240	0.166	243	154
4	17	83521	2160	1.087	1265	691
5	21	4084101	12960	4.07	7679	3715

Figure 2.12a shows the grid with the new lines. We have added them in the order 7–16, 2–13 and 21–22. Figure 2.12b and Table 2.1 gives some indicators for the complexity – in the figure the indicators are normalised to their value at the base case with two cycles.

The indicators we investigate are the number of cycles C, number of edges in the cycles E, the theoretical bounds on the complexity given by $\mathcal{O}(E^C)$ and the tighter $\mathcal{O}(\prod_i E_i)$, the total time needed for the computation t in seconds, the total number of iterations I_{tot}, and the number of iterations where we actually compute the power distance, denoted by I_{Pd}. The total number of iterations also includes iterations where we identify that the resulting graph is either disconnected or still has a cycle.



(a) *To test the scaling, lines 7–16, 2–13 and 21–22 have been added (in this order).*

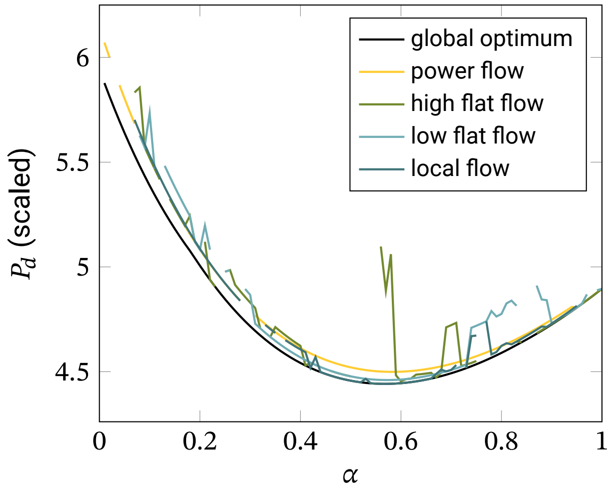


(b) *As expected, the problem scales badly with the number of cycles, however less quickly than predicted.*

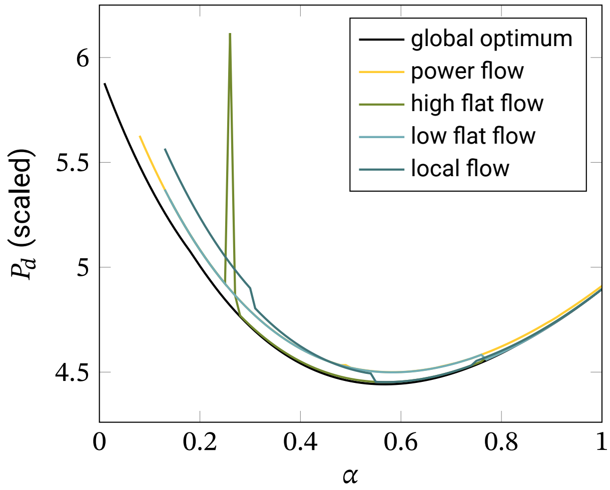
Figure 2.12.: *Scaling of the brute force approach for the meshed test system, with additional cycles introduced as shown in the top figure.*

It turns out, that even the total number of iterations that are effectively performed is growing much slower than the theoretic complexity bound. This is because if we remove a set of lines leading to a disconnected graph, we do not need to go through all other combinations including this set of lines. Hence, we do not reach all combinations in the combinatorial space. For example, in the system with all three new lines, if we remove line 2–3 and line 2–13, the graph is disconnected. We therefore stop searching all other combinations that include the removal of 2–3 and 2–13.

The bound based on the product of the number of line elements per cycle, $\mathcal{O}(\prod_i E_i)$, gives a reasonable prediction of the computation complexity. The remaining losseness of the bound can be explained by the fact that this bound does not consider that some line elements can be part of several cycles, hence overestimating the number of possible combinations. The best proxy we have for the total computational time is the number of iterations in which the power distance is actually computed. This is reasonable, as this is the most complex task in the iteration. It also suggests that the highest potential to improve the efficiency of the algorithm is in the calculation of the power distance.



(a) *Using the non-linear solver IPOPT.*



(b) *Using the SQP approach.*

Figure 2.13.: *Results with different initial conditions*

2.7.3. Computing local optima

Approach 1, using SQP, and Approach 2, using the non-linear solver IPOPT, both depend on the initial condition. In this section, we will discuss the properties of the different initial conditions proposed, and more generally the properties of the two solution approaches.

Figure 2.13 shows the difference between the global optimum (black lines) and the nonlinear and SQP approaches for the meshed system. The different initial conditions lead to the different solutions, shown with the coloured lines. We highlight a number of findings:

Robustness Both approaches do not always find a solution, even for feasible starting conditions. In the figures, this can be seen by gaps in the coloured lines. However, the SQP approach seem to be somewhat more robust, at least for larger α. The robustness materialises in two ways. First, there are few α values larger than 0.2 for which the SQP does not find a solution. Second, changing alpha usually leads to a smooth change in the power distance computed by SQP, while for the nonlinear solver, small changes in α can lead to a different local optimum with a different associated power distance.

Of course, in a regulatory setting, robustness is essential to avoid uncertainty and discussions.

Effect of the initial condition We have tested four different initial conditions; the power flow, a high flat flow, a low flat flow, and initial flows based on local demand.

Of these four initial conditions, the power flow leads to the most stable solution, however often not to the best solution found by the two approaches. Rather, the power flow as initial condition seems to nearly always lead to a specific solution, in our case removing lines 7–8 and 8–9.

The other conditions are less robust, but often outperform the power flow as initial condition. SQP seems to work well with the high flat flow initialisation (except for one outlier), while IPOPT seems to prefer the local flow initialisation.

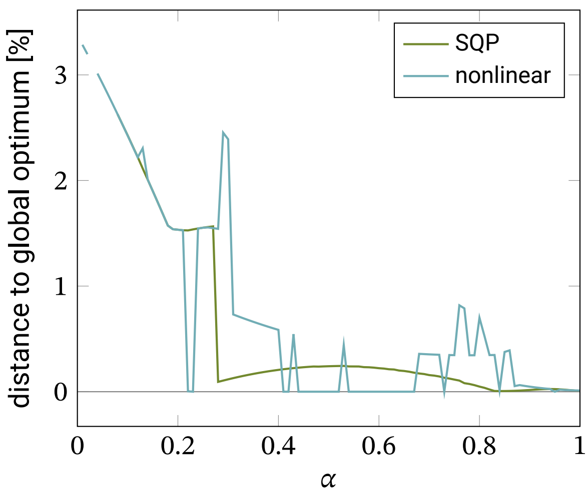


Figure 2.14.: *Difference in percent between the best solution found with either optimisation approach, and the global optimum.*

Relative difference between the local and global optima While we do not have a theoretical bound on the distance between any local optimum and the global optimum, we can check how far off we are for our test system.

To this end, we compare the best solution found with either approach and with any of the four initial conditions to the global optimum. Figure 2.14 shows the results. Again, we see that the SQP approach is more robust, while the nonlinear optimisation sometimes is better, and sometimes

Table 2.2.: *Comparison of computation time and solution quality between SQP, nonlinear (NL) and brute force (BF) approaches. α is 0.7.*

cycles		2	3	4	5
SQP	time [s]	0.66	3.03	1.72	1.82
	iterations	4	24	13	13
	P_d	4.538	4.545	4.187	4.122
	Delta [%]	1.02	1.18	1.43	1.28
NL	time	0.12	0.14	0.17	0.14
	P_d	4.537	4.561	4.244	4.179
	Delta [%]	1.00	1.54	2.81	2.68
BF	time [s]	0.044	0.166	1.087	4.07
	iterations	42	243	1265	7679
	P_d	4.492	4.492	4.128	4.07

worse than the SQP approach. While the results generally look promising with an error less than 1 % for the relevant area of α, we remind the reader that it is not clear if the problem behave equally well for larger systems.

Solution time and scaling To test the scaling of Approaches 1 and 2, we would need to test them on a number of differently sized systems – which we do not have in the scope of this project. However, we can highlight the difference between finding the global optimum and finding local optima, by comparing how all approaches behave when faced with networks of different complexity. For this, we use the meshed system and the same set of added lines as in Figure 2.12. In addition, we choose the power flow as initial condition, and set α to 0.7.

Table 2.2 shows some metrics to compare the three approaches. We see some interesting effects. The computation time for the SQP approach is generally significantly higher than for the nonlinear approach. This is due to the fact, that the problem needs to be solved repetitively. The number of iterations in the SQP approach are given as well, and computation time scales linearly with iterations.[7] However, the computation time does

[7] As all code is implemented in Python – a scripting language – all computation times are only indicative. Keep also in mind, that for the non-linear solver we only use minimal own code and mostly pre-compiled libraries, while for the

not seem to depend much on the complexity of the grid, certainly not for the nonlinear approach.

We also see that the local optima found generally are further away from the global optimum as the system becomes more meshed. This indicates that for large systems, using the SQP or nonlinear approach might lead to significantly worse results.

SQP approach most time is spent within non-optimised scripts.

3. Power and energy distance as inputs to the DEA

In this chapter, we will discuss how the power distance can enter the DEA benchmarking used for the income regulation of DSOs in Norway. We describe challenges with aggregating the power distance into a yearly energy distance, highlight data requirements and discuss incentives set by the power distance.

3.1. Energy distance: from hourly values to a yearly parameter

The power distance is computed for a specific hour. However, for the DEA approach, we need one parameter representing the whole year. In this section, we will discuss different options on how to move from hourly "power distance" to yearly "energy distance".

Designing a good measure for the energy distance is challenging. In fact, it is a similar challenge as for the power distance: the power distance aims to take different spatial distributions of demand into account, while the energy distance should consider temporal distribution of demand. Effectively, the method in which the energy distance is computed from the power distance decides how DSOs with different demand profiles are compared against each other.

We discuss four approaches to moving from power distance to energy distance; aggregating hourly power distances, selecting representative hours, averaging demand over several hours, and formulating an optimisation problem directly for the energy distance. For all discussions we assume that data to compute the power distance in each hour of the year is available.

3.1.1. Aggregating hourly power distances

The simplest approach to arrive at an energy distance would be to use some sort of summation, or in generalised terms a norm over all hourly values. Typical norms would be:

1-norm This is the sum over the absolute values.[1]

2-norm This is the square root of the sum over the squares.

∞-norm The infinity norm, or largest absolute value in the set.

These three norms have different interpretations. The 1-norm considers the total energy, but is independent of the distribution over time. The demand profile might be flat or uneven, but the 1-norm would be the same[2]. As the DSO has to provide a grid covering all hours, his task is more complicated if the demand is uneven. Applying a 1-norm could be interpreted as an incentive or expectation on the grid operator to promote even consumption in his supply area.

2-norms are typically used if outliers should have a higher impact than values closer to zero. Hence, the 2-norm somewhat addresses the problem of uneven demand, but does not have an immediate interpretation. In any case, using the 2-norm, hours with a high power distance value (high

[1] As all power distances are positive, it is the same as the sum of hourly power distances. If one divides by the number of hours, it is the mean.

[2] Strictly, this holds only for $alpha = 1$, however, the core of the argument also holds when considering $\alpha < 1$.

demand) are weighted more than those with a low power distance.

The ∞-norm only looks at the largest power distance in the year. This is meaningful in the sense that this is actually the largest power distance on which the DSO has to dimension his grid. But at the same time it would give the DSO an incentive to promote increasing the peak demand, in order to have a higher output in the DEA.

When aggregating hourly power distances, we acknowledge that there is a trade-off between representing the task of the DSO as closely as possible, and giving the DSO an incentive to promote reducing peak demand.

3.1.2. Selecting representative hours

To simplify any computation, one could aim at using representative hours rather than all 8760 hours of the year. For example, one could use the hour with the highest net demand (not necessarily the same as the largest power distance), or representative hours such as winter peak, summer valley or just the first Wednesday at 15:00 of each month. Representative hours can make the task of two DSOs less comparable, as one DSO might be faced with high demand in the relevant hours, while the other by chance might have a low demand in the representative hours. It would also not be possible to control whether DSOs take measures to affect demand and hence power distance in those hours.

Alternatively, one could attempt to identify specific demand situations and compute one representative hour for each situation, potentially weighted by the number of hours with a certain situation. Such identification would rely either on manual identification or on clustering algorithms.

Manual identification Manual identification draws on the knowledge of the grid owner or some other expert knowledge. However, in a regulatory setting, the lack of comparability and determinism is prohibitive.

Clustering algorithms Algorithmic clustering, that is using a program to identify similar demand patterns, has similar drawbacks. Each node would be one dimension, and the demand distribution in each hour a data point in this high-dimensional space. The clustering algorithm would try to identify demand patterns in which the demand points are close to each other. Finding an optimal clustering is tricky. First of all, what is optimal in a clustering sense would depend on user defined parameters, such as the number of clusters to be found. Second, while efficient algorithms exist, these are heuristic algorithms, which are challenging in a regulatory environment where deterministic outcomes are necessary. In the end, it is not clear if such an approach would deliver any net benefit to the problem. We have therefore not attempted to implement it, and recommend against it.

3.1.3. Averaging demand over several hours

Instead of computing the power distance for each hour and then averaging, one can also first average the demand over a set of hours and then compute the power distance. This has a number of advantages. As computing the power distance is challenging for meshed networks, we profit from computing it less often. It also makes the approach more robust against outliers.

The reduced computational effort makes it easier to take more or all hours of the year into account, and hence makes the approach more robust against the DSO implementing demand or generation management in representative hours that is affecting the power or energy distance.

The drawback is that extreme demand peaks may be reduced by the averaging, and the challenge a DSO is actually facing from a very uneven demand distribution would not be recognised appropriately. This could be helped by averaging groups of hours that have similar characteristics, for instance all hours between 06:00 and 09:00 on weekdays for a given season.

3.1.4. Combined computation of the power and energy distance for a full year

Lets assume that the DSO cannot change its topology during the year, and that the energy distance should be computed as the minimal grid that covers demand at all hours. It is possible to give a precise formulation of the energy distance as an optimisation problem.

Energy distance problem formulation

The approach is to compute the power distance for all hours in parallel. As cost, we do not take the individual power distances $\tilde{p}_e$, but the upper bound on each line needed during the year, $\tilde{e}_e$. In a way, this is similar to the ∞-norm. However, instead of only taking the supremum of the individual power distances, we co-optimise all power distances in parallel. Hence, it might be that we find a topology that leads to a lower overall grid requirement, but a higher grid requirement in individual hours.

The problem can be formulated as follows

$$E_d = \min \sum_{e \in E} \mathrm{e}^{\alpha\, \tilde{e}_e + \tilde{L}_e} \tag{3.1a}$$

s.t. $\forall k \in T, \forall e \in E, \forall j \in N$

$$\tilde{p}_{e,k} \leq \tilde{e}_e \quad , \tag{3.1b}$$

$$D_{j,k} - G_{j,k} = \sum_{i \in E_j^+} \mathrm{e}^{\tilde{p}_{i,k}} - \sum_{h \in E_j^-} \mathrm{e}^{\tilde{p}_{h,k}} + s_{j,k}, \tag{3.1c}$$

$$\underline{S}_k \leq s_{j,k} \leq \overline{S}_k \quad . \tag{3.1d}$$

k in T are all time steps k under consideration in the period T, usually one year. The cost function now uses the "energy flow" on each line segment, $\tilde{e}_e$, rather than the power flow $\tilde{p}_e$. The new constraint (3.1b) is the core of the approach: here the energy flow – one variable per line segment over the year – is connected to the power flow, which may change in each hour. (3.1b) ensures that in all hours sufficient line capacity is available.

To highlight how this works, consider this toy example: let the power distance be minimal with flows on line A in most hours, but in a specific hour the power distance would be lower when using line B. The optimisation would have the discretion to route power over line A even in that specific hour. As the cost of the line is already accounted for by the usage in the other hours, doing so would not increase the energy distance, while adding line B would. Hence we minimise the energy distance, even though in one hour we would have a higher power distance than the optimal power distance for that hour.

Computational complexity and solution approaches

Elementary, the same solution approaches as for the power distance can also be used for the energy distance problem. That is, we can attempt to solve the non-linear problem using a non-linear solver or SQP approach, with the limitation that this gives no guarantee if we found a local or global optimum. Alternatively, we can use a method to find the global optimum, such as the brute force approach iterating over alternative topologies.

Given the high-dimensional space, the SQP and non-linear approach are bound to get stuck in local optima. We therefore see these approaches as not too promising for this larger and more complex task.

Interestingly, using the brute force approach might be feasible. Since we assume that we use the same topology in every hour of the year, the *number* of iterations does not increase compared to solving the power distance for a single hour. The execution time of each iteration, however, will increase linearly by the number of hours under consideration. In other words, we use the same combinatorial approach as in the power distance brute force approach, but instead of computing the power distance for one hour for each potential topology, we compute it 8760 times. If we can apply the brute force algorithm to Problem (2.1), it should be possible to apply it to Problem (3.1) as well.

In the end, this will depend on the complexity of the real-life grids found in Norway.

Extension to KILE costs

We can further extend the optimisation to include KILE costs in the optimisation. For this end, we append to cost function

$$E_d = \min \sum_{e \in E} e^{\alpha \tilde{e}_e + \tilde{L}_e} + \sum_{j \in N} \sum_{k \in T} K_{j,k} \tag{3.2}$$

and constraint (3.1c)

$$D_{j,k} - G_{j,k} - K_{j,k} = \sum_{i \in E_j^+} e^{\tilde{p}_{i,k}} - \sum_{h \in E_j^-} e^{\tilde{p}_{h,k}} + s_{j,k} \quad . \tag{3.3}$$

In case there are a few extreme demand situations over the year, the optimisation now can decide if it is cheaper to pay the KILE costs or to upgrade the grid to cover the demand 100 % of the time.

3.2. Data availability and quality as a precondition

If the power distance and energy distance is to be calculated based on the problem definition used in the previous chapters, each system operator will have to provide a large amount of very detailed data. Some of this data is expected to be readily available through ElHub, while other data could be more challenging to retrieve. In short, the necessary data for calculating the power and energy distance is

- Hourly demand data at each node/customer
- Capacity of all substations
- Hourly generation data from distributed generation
- Grid topology, including length of all lines and preferably coordinates of nodes/customers

Demand and distributed generation

To calculate the power distance, it is necessary to know the demand of each customer served by a given system operator. If a customer produces and feeds in electricity to the grid itself, that has to be accounted for as well. We expect demand data to be readily available through ElHub, and that the format of the data will be similar across all system operators. The same quality and data of input is key to implement efficient algorithms for calculating the power distance.

It is however necessary to define at which level the data should be aggregated: that is, should demand be accounted per customer or per node, and how is a node defined. In other words, the voltage level that is studied needs to be specified. Similarly, the regulator needs to specify from which time frame demand data is used, e.g., demand data of the last year, the last three years, corrected for temperature, or an average of the demand over the last N years.

Capacity of substations and grid topology

Regarding capacity of substations and information about grid topology (length of lines and coordinates of nodes/customers), there exists no formally defined platform gathering this data. As a result, the format and quality could have great variations across the different system operators. The regulator should expect challenges in achieving a good and homogeneous data quality for substations and grid topology, unless a common database similar to the ElHub or at least a common standard is mandated by the regulator.

Data quality and missing data

There are two categories of challenges that occurs in collecting and handling large amounts of data. Both challenges would need to be addressed if the goal is to implement an efficient calculation of the power and energy distance. The first challenge is the general quality and comparability of data from different sources, as discussed for substations and grid topology. The second challenge is missing or unavailable data. When working with large amount of data with hourly resolution, it is bound to exist missing data points and the solution algorithm would need a procedure for how to handle missing data.

3.3. Improved exogeneity but disincentives to invest

The goal of the power distance measure is to provide a better variable for measuring the tasks that each DSO faces, and to exogenise incentives. It should also provide DSOs with a sufficient incentive to invest, while not incentivising overinvestments. Finally, the power distance should not discourage mergers.

We find that the power distance increases exogeneity, tends to provide investment disincentives rather than incentives to overinvest, and might in some cases slightly penalise mergers.

Exogeneity Making the incentives exogenous means, that the parameter used as output in the DEA should to the largest extend possible depend on parameters exogenous to the DSO, such as the demand distribution, and not on parameters endogenous to the DSO, such as investment decisions taken by the DSO.

The power distance relies on the existing topology of the grid. This limits to what extend it can be considered as fully exogenous. In addition, the demand distribution may be to an extend under the control of the DSO. Hence DSOs could attempt to affect the power distance in two ways, in the short term by managing demand and in the long term by investment decisions.

The extend to which DSOs can affect their output by demand side management depends on their ability to influence demand for capacity and/ or energy and on the detailed design of the energy distance as discussed in Section 3.1). Using a definition of energy distance that addresses this theoretical risk should be an effective measure to prevent this from happening.

We discuss the investment incentives of the power distance in more detail in the next two paragraphs. Here we compare it to the current regulation: the impact of investments on the own output is already present in the parameters used today, namely number of substations and line lengths. As the power distance considers the actual demand as an exogenous element, the measure is likely to provide more accurate incentives for future grid development than the current output parameters.

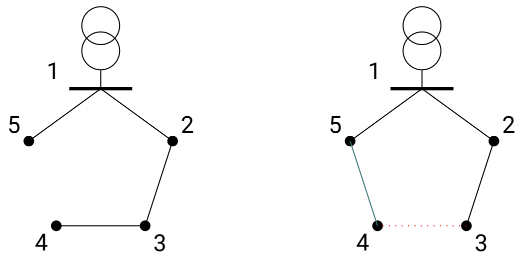
Incentive to overinvest The second issue concerns incentives to overinvest, in order to achieve a higher power distance. In theory, a DSO would profit from building fewer but longer lines, and from building substations as far away from large consumers as possible. However, the incentives are weaker than in today's outputs and overinvestment should also be curbed by local planning authorities. Overinvestment is therefore not a major argument against the power distance measure, but rather something that NVE should keep in mind.

Disincentive to invest Third, the power distance favours radial grids over meshed grids compared to the current model. As the power distance solution always will result in a radial flow, the redundancy of meshed grids is not reflected. In principle, this can lead to lower investments and lower security of supply than the optimal level from an economic welfare perspective.

It is easy to construct an example where a DSO has a disincentive to make a sensible investment improving security of supply. Consider Figure 3.1: on the left is a grid with a radial structure. Adding line 4–5 would close a cycle and improve security of supply, as each node now would be connected to the substation via two paths. However, the power distance parameter *decreases* by around 10 %. While this might be an extreme example, it is not unusual for DSOs to take the decision to add lines in order to create a more meshed net in order to improve security of supply.

Incidentally, we have shown the effect already

(a) Before investment (b) After investment



$P_d = 5.782$ $P_d = 5.249(-9.2\,\%)$

Figure 3.1.: *Example for a disincentive to invest: adding a line between nodes 4 and 5 leads to a reduction in power distance output of nearly 10 %. The dotted red line is considered superfluous by the power distance measure.*

in the results section in 2.7.2. When we discussed the scaling of the brute force approach, we tested the meshed grid and successively added lines. While adding lines, the power distance decreases, see the results in Table 2.2.

However, the regulatory model also includes cost of energy not supplied as a key parameter, which helps balance the incentives. To the extent that the general investment incentives are weakened, this could also be mitigated through other measures, e.g. the regulatory WACC or the calibration mechanism for the cost norms. To alleviate the issue further, one could consider a bonus for nodes that have two connections to the next substation, effectively giving an incentive for meshes – however, such an adjustment would in turn need to be checked for new loop holes or wind fall profits.

Impact on mergers Finally, the measure can have an impact on the incentives for mergers. The power distance measure is not additive or linear, in contrast to the number of substations and line length. If two adjacent nets are merged, the total power distance will be either the same (if the "new" connection is not used in the optimal grid) or lower (if the "new" connection is part of the optimal grid), but cannot become higher than the sum of power distances of the individual grids.[3] This could lead to a loss in DEA score for an unchanged cost level.

NVE already compensates companies for reduced DEA scores due to the general characteristics of the DEA model (the harmony effect). The power distance measure may necessitate additional compensation mechanisms in order to preserve the incentives for mergers. A possible area for further work could therefore be to look at the merger incentives with specific designs of the power distance measure and actual network data when it becomes available.

[3] This also raises the question how neighbouring, connected grids by different owners will be handled in the power distance computation: are they considered disconnected or connected?

4. Alternatives to the Power Distance

The Power Distance has some quite specific properties. For example, it will always prefer radial networks and therefore ignores the benefits of meshed topologies for security of supply. The Power Distance as defined in this report also respects the existing topology, and therefore cannot highlight more efficient grid configurations. On the other hand, it provides the regulator with a good indication of a lower bound for the costs of transferring energy, and informs whether the existing grid is excessively oversized or not.

As discussed in the beginning of this report, the power distance does constitute an improved measure for the task of each DSO compared with the existing outputs such as length of high voltage lines. Nevertheless, it is worth investigating if other measures might give a better comparison of the DSOs tasks. For example:

Power Flow Compute power flow either with existing line parameters or fixed line parameters per length, but without power constraints . Then compute cost including the alpha parameter "ex-post". This approach will result in the same flow independent of the value of alpha, but the actual cost will vary with alpha.

Optimal grids One could also imagine a parameter computing the optimal grid, independent of the given topology. However, this would be challenging for to reasons. First of all, a lot of information would be needed. Second, a DSO with a sub-optimal grid configuration will not realistically be able to improve on the existing grid. The resulting measure may therefore be very hard on some DSOs.

In the next section we test how a power flow based power distance compares with optimal power distance.

4.1. Power-flow based power distance versus optimal power distance

The cost of transporting power for the power flow based power distance is calculated in the same manner as the optimal power distance, but it is simply a sum of parameters rather than a minimisation problem.

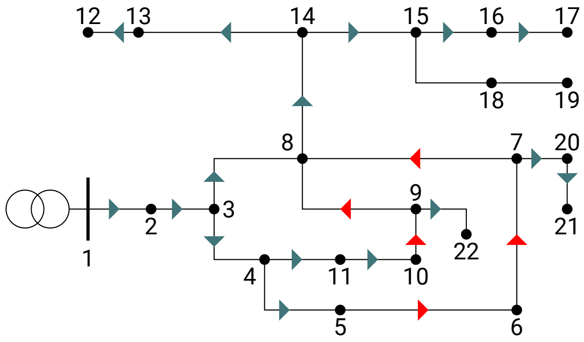
$$P_{\mathrm{d}} = \sum_{e \in E} L_e \, |P_e|^{\alpha} \tag{4.1}$$

Figure 4.1a show how the power will flow on the lines in the meshed test case if we perform a DC power flow with fixed line parameters per line length. In contrast to the solutions found when calculating the power distance, the power flow approach results in loops in the grid, i.e. node 8 is receiving power from more than one node; nodes 3, 9 and 7.

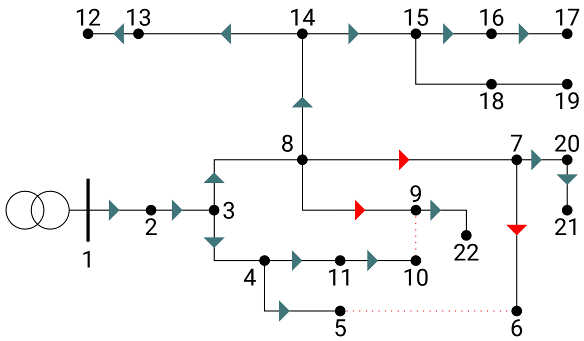
The direction of flow given a brute force calculation of the optimal power distance in the same grid is illustrated in Figure 4.1b. The main differences are the lack of loops in the optimal power distance solution, and the direction of flow on lines 6–7, 7–8, and 8–9. The magnitude of flows on the different grid elements is otherwise often quite similar. The exact flows for both cases can be found in Table 4.1.

The power distance based on a power flow has some interesting differences to the optimal power distance. First, the power flow based approach allows for meshed solutions, that is, the power flow based approach will keep all lines.[1] Also, the power

[1] This is related to the power flow being easier to compute than the optimal power distance.



(a) *Power flows.*



(b) *"Flows" in the global optimum for the power distance problem.*

Figure 4.1.: *Direction of power flows when computed directly, and power flows leading to the globally optimal power distance. Red arrows highlight difference in direction between the two lines, dotted lines are not used. α is 0.7.*

Table 4.1.: *Power flow and flows in globally optimal solution.*

From	To	power flow	global optimum
1	2	10214	10214
2	3	10214	10214
3	4	4923.8	1965
4	5	2027.5	728
5	6	1299.5	-
7	8	118.5	-1181
8	9	-1017.3	642
9	10	-1659.3	-
10	11	-2134.3	-475
3	8	5014.2	7973
12	13	-250	-250
13	14	-284	-284
6	7	751.5	-548
11	4	-2464.3	-805
14	8	-5563	-5563
14	15	5070	5070
15	16	225	225
16	17	225	225
15	18	0	0
18	19	0	0
7	20	556	556
20	21	331	331
9	22	68	68

flow itself is only subject to the physical properties of the grid, and is independent of the value of α.

Figure 4.2 is a plot of the power distance for different values of α based on power flows directly, based on power flow as initial condition for the SQP approach and the nonlinear solver, as well as the globally optimal solution found with brute force calculation. For all values of α, the power flow based power distance is the furthest away from the globally optimal power distance. The solutions found with the power flow as initial conditions are always better when a solution is found, but there are values of α at which the solver has not been able to find a solution at all.

Two of the main benefits of the power flow based power distance is that the quality of the solution will be consistent and transparent for all DSOs, and that there always will exist a solution to the problem. For the optimal power distance, unless we are able to guarantee a globally optimal solution, one DSO could be measured against a solution very close to the optimal power distance, while another DSO might be measured against a power distance far from the optimal solution. As we have seen in Figure 4.2, a solver might not even be able to find a local optimum to describe the DSOs task.

In terms of using power distance as an output describing the task of the DSOs, it is necessary to guarantee that the companies are being evaluated on the same basis. The power flow based power distance does guarantee consistency and transparency, while accounting for the distribution of load, but it will not reward better designed grids in the same manner as the optimal power distance

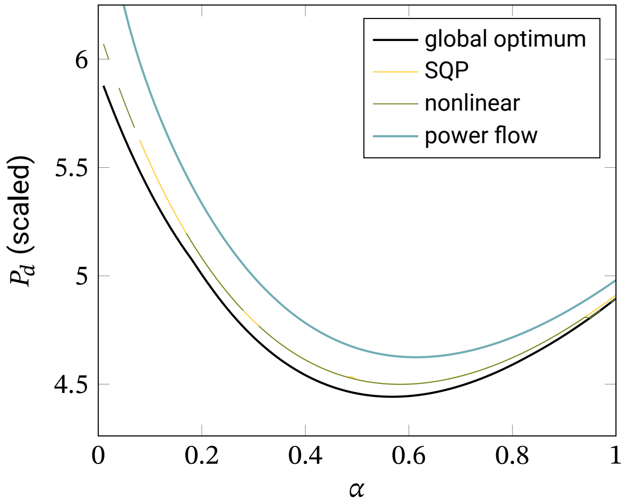


Figure 4.2.: *Power distance using the power flows directly, and the global optimum.*

can. On the other hand, all though we have shown that with the power flow as initial condition the local optima are quite robust for the test cases studied in this report, it could be hard to guarantee that companies are being evaluated on the same basis when using the optimal power distance as a measure.

5. Conclusions and next steps

Computing the optimal power distance proved to be a challenging problem. The main findings, as well as some suggested next steps, are summarised in the following.

Computational complexity Computing the optimal power distance is a non-convex problem, which cannot be reformulated to a convex problem. Different approaches for finding solutions were proposed, two for finding local optima and one brute force approach for finding the global optimum. However, the former have no guarantees either on convergence nor bounds on the optimality gap, while the computation time of the latter grows with the power of the number of cycles in the grid. Whether or not it is possible to determine the globally optimal power distance will depend on the complexity of actual Norwegian distribution grids.

Data requirements include demand data and grid topology Any parameter taking the demand distribution as an exogenous indicator for the task of the DSO depends on the availability of demand data, ideally in good spatial and temporal resolution. The power distance in addition needs the grid topology in a form that not only contains individual line elements, but also how lines are connected to each other.

Power distance in the DEA The power distance aims to be an exogenous parameter, and is certainly an improvement over parameters such as the line length or number of substations. However, the DSO can to some extent still take investment decisions that affect their own output. It would be very interesting to study both investment and merger incentives of the power distance in the DEA, applying specific investment decisions and mergers to actual network data.

From power distance to energy distance A selection of approaches to computing a yearly parameter (energy distance) were discussed, including direct computation of the energy distance as an optimisation problem. If one attempts to reduce the number of hours for which the power distance is to be computed, it is certainly beneficial to first average the demand over a number of hours, and then compute the power distance. One could define several outputs for the DEA based on the power and energy distance, with the objective of representing different characteristics of each grid.

Alternatives to the power distance To avoid the computational challenges, one could use a power flow on an unconstrained grid with normalised line parameters. This would still inform about lines that are oversized, but not of lines that are superfluous. The advantage is a mire intuitive formulation, a higher degree of consistency and transparency, and a significantly faster or more robust solution. It would however require further studies on a larger selection of test cases to conclude on a comparison of the power distance based on power flow with the optimal power distance.

Open questions and further research In this project, we have only tested and discussed approaches on smaller test cases that are not actual examples from Norwegian DSOs. It would be interesting to see how the solution approaches perform with real-life grid topologies. If data from neighbouring grid companies were available, one could also discuss the influence on mergers on the benchmarking of the companies. Finally, one could investigate if there are investments that the DSO could make or avoid in order to increase their own output when measured against the optimal power distance as defined in this report.

A. Acronyms

DEA Data Envelopment Analysis

DFS Depth-First Search

DSO Distribution System Operator

KILE *kvalitetsjusterte inntektsrammer ved ikke levert energi* (cost of energy not supplied)

LP Linear Program

NVE Norges Vassdrags- og Energidirektorat

QP Quadratic Program

SQP Sequential Quadratic Programming

B. References

[1] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. 2009.

[2] A. Wächter and L. T. Biegler. "On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming". In: *Mathematical Programming* 106 (1 2006), pp. 25–27.

Disclaimer

About THEMA:
THEMA Consulting Group is a consulting firm focused on electricity and energy issues, and specializing in market analysis, market design and business strategy.

THEMA Consulting Group

Øvre Vollgate 6
0158 Oslo, Norwegen

support@thema.no
https://www.thema.no/

Berlin office

Friedrichstrasse 68
10117 Berlin, Deutschland

NVE

Norges vassdrags- og energidirektorat

MIDDELTHUNSGATE 29
POSTBOKS 5091 MAJORSTUEN
0301 OSLO
TELEFON: (+47) 22 95 95 95