Using energy carrier price information to understand trade-offs between different configurations of a fully sector-coupled energy system model

Master thesis submitted to Delft University of Technology

in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

in **Complex Systems Engineering and Management**

Faculty of Technology, Policy and Management

by

Tjun Sing Alex Sow

Student number: 4322134

To be defended in public on August 29th 2022

Graduation committee

First Supervisor	: Dr.ir. L.J. de Vries, Section Energy and Industry
Second Supervisor	: Dr. ir. Ö. Okur, Section Systems Engineering
Advisor	: Dr. F. Lombardi, Section Energy and Industry

Preface

This master thesis report is written as part of the CoSEM Master Thesis (SEN2331) at Delft University of Technology.

Readers who are interested in the background information and relevance of this research project can refer to Chapter 1. Readers who are interested in the methodology of this research can refer to Chapter 2. Readers who are interested in the extraction method for energy carrier price information can refer to Chapter 3. Readers who are interested in the fully sector-coupled model can refer to chapter 4. Readers who would like to know more about the results, model limitations, conclusion and recommendations can refer to Chapter 5, 6 and 7 respectively. The links to models and the Python codes used in this project can be found in the Appendix.

I hope you enjoy reading this master thesis report.

T.S.A. Sow
Delft, August 2022

Acknowledgements

Having started my master's program in a lockdown period due to COVID-19 and ending it during the Russian invasion of Ukraine made me realise how fragile and interlinked our world really is. It became clear that no matter whether it's a global pandemic, climate crisis or energy crisis, the consequences affect all of us. Using the knowledge that I have gained during my MSc Complex System Engineering & Management program I wish to contribute to the betterment of this complex world.

I would like to thank Prof.dr.ir. L.J. de Vries for taking up the role of chairperson and first supervisor of my master thesis project as well as Dr. ir. Ö. Okur as my second supervisor. Their guidance and insights during the meetings we had were of great value to me.

I would also like to thank Dr. F. Lombardi for his guidance throughout my entire master thesis project. I would not have been able to complete my master thesis project in time if it were not for his help with Calliope, Python and general day to day guidance. I'm grateful to have been part of the Calliope modelling group.

Finally, I would like to thank my family, my dear girlfriend and my close friends for their support throughout my entire academic journey. Without their continuous support, I would not have had the courage and freedom to achieve what I have achieved in my life so far.

Table of Contents

Preface 2
Acknowledgements 3
List of Figures 8
List of Tables 10
Abbreviations 11
Executive summary 12
Chapter 1 – Introduction 13
1.1 Background 13
1.2 Core concepts 14
1.2.1 Energy system optimisation tools 14
1.2.2 Sector coupling 14
1.2.3 Energy carriers 15
1.3 Research question 16
1.4 Sub questions 17
1.4.1 Research methods and research flow diagram 17
1.4.2 Research flow 18
1.5 Deliverables 18
Chapter 2 – Research methodology 19
2.1 Phase 1: Problem definition 20
2.1.1 Linear programming problem and shadow prices 20
2.1.2 Duality in Linear Programming Problems 21
2.1.2 Calliope 22
2.2 Phase 2: Design and development 23
2.2.1 Working environments 24
2.2.2 Modelling process 25
2.1.3 Phase 3: Design evaluation and communication stage 27
Chapter 3 – Extracting price information from Calliope 28
3.1 Literature review on extraction methods 28
3.2 Mathematical formulation of the Calliope modelling framework 28

3.2.1 Objective function 28
3.2.2 Decision variables 29
3.2.3 Constraints 29
3.3 Extraction method 30
3.4 Testing on simple Calliope models 31
Chapter 4 – Price information in a fully sector-coupled Calliope model 33
4.1 Building the model 33
4.1.1 Europe-Calliope 33
4.1.2 North-Sea Calliope 34
4.2 Running the North-Sea Calliope model 39
4.3 Validation of the 2020 North Sea Calliope model 40
4.3.1 Physical model 40
4.3.2 Dual values for electricity 42
Chapter 5 – Assessing trade-offs within different hydrogen configurations 44
5.1 Experimental setup 44
5.1.1 Base model 44
5.1.2 Weather scenarios 47
5.1.3 Hydrogen configurations 48
5.1.4 Overview of the configuration sets 50
5.2 Analysis of the results 51
5.2.1 Total energy supply 51
5.2.2 Capacity deployment 54
5.2.3 Total cost energy system 57
5.2.4 LCOE of technologies 57
5.2.5 Price stability 59
5.2.6 Price and load duration curves 63
5.2.7 Payback time 65
Chapter 6 – Discussion 67
6.1 Extraction method for shadow prices within the Calliope modelling framework 67
6.1.1 North Sea Calliope 2020 model limitations 67
6.1.2 Best method for the extraction of shadow prices within the Calliope modelling framework 68
6.2 Assessing trade-offs within different hydrogen configurations 68

6.2.1 Weather 68
6.2.2 Capacity deployment in optimization models 68
6.2.3 Price stability and electricity shadow prices 68
Chapter 7 – Conclusions & Recommendations 70
7.1 Sub question 1 70
7.2 Sub question 2 70
7.3 Sub question 3 71
7.4 Sub question 4 71
7.5 Main research question 72
Bibliography 73
Appendices 78
Appendix A – Laptop specifications 78
Appendix B – Calliope models 78
B1 - Simple model 78
B2 - North Sea Calliope model 78
Appendix C – Building the 2020 North Sea Calliope model 79
Appendix D – Description of North Sea Calliope 2020 override files 80
Appendix E – Job script 2020 model 1h resolution 81
Appendix F – Override files for setting the hydrogen shares for the production of synthetic fuel 82
Appendix G – Override files for increasing the fuel demand 82
G.1 Setting fuel demand increase to 30% 82
G.2 Setting fuel demand increase to 50% 83
Appendix H – Average capacity factors wind offshore and onshore 83
Appendix I – Electricity production vs electricity shadow price 84
Appendix S1 – Python scripts for the analysis of the North Sea Calliope 2020 model 86
Loading results from the supercluster 86
Analysing TES for the Netherlands 86
Analyzing electricity duals 87
Appendix S2 – Python scripts for the analysis of the North Sea Calliope 2050 model 88
Loading 2050 models 88
Setting up electricity duals data 90
TES by source in the Netherlands 2050 92

Electricity demand 94
Total electricity production by source 96
Total energy system cost 99
Levelized cost of energy 99
Price stability boxplots 104
Time series electricity duals vs shadow price 105
Duration curves 109
Payback time 114

List of Figures

Figure 1: Research methods, tools and sources 17
Figure 2: Research flow diagram 18
Figure 3: Overview of the design stages and the associated key tasks 19
Figure 4: Simplified overview of the Calliope Modelling Framework 22
Figure 5: Overview of the interactions between different working environments 24
Figure 6: Overview of the modelling process 25
Figure 7: Duals data frame after reformatting 30
Figure 8: Overview of the simple model 32
Figure 9: Assessing power duals against power demand from the simple model 32
Figure 10: Overview transmission and node network from the Euro-Calliope model, figure from GitHub – Calliope project (2022) 34
Figure 11: Overview of the North Sea Calliope Model, figure from Lombardi (2022) 35
Figure 12: 2020 North Sea Calliope model within the Calliope modelling framework 38
Figure 13: TES by source in the Netherlands for the year 2020. Data from IEA (2022) 41
Figure 14: Energy supply by source, comparison between IEA data and model data 41
Figure 15: Electricity duals plotted together with ENTSO-E day-ahead prices 42
Figure 16: Electricity duals plotted together with ENTSO-E day-ahead prices with only power sector Calliope model 43
Figure 17: 2050 North Sea Calliope model within the Calliope modelling framework 46
Figure 18: 2050 North Sea Calliope model with different weather variations 47
Figure 19: Methodology for varying the share of hydrogen in the model 48
Figure 20: Hydrogen share for the production of synthetic fuels 49
Figure 21: Synthetic fuel production share in total energy system 50
Figure 22: Total energy supply by source in the Netherlands in 2050 51
Figure 23: Comparing model data with Infrastructure Outlook 2050 Data on the TES by source 52
Figure 24: TES by source in the Netherlands 2050 with increased fuel demand 53
Figure 25: Comparison electricity demand by source optimal scenario vs +50% fuel demand 53
Figure 26: Total electricity production by source North Sea region 54
Figure 27: Total electricity by source for the North Sea region for different fuel share demands 56
Figure 28: Total electricity production by source in The Netherlands (optimal hydrogen share) 56
Figure 29: Total energy system cost for different hydrogen configurations 57
Figure 30: Levelized cost of energy of technologies for the production of electricity in different hydrogen configurations 58
Figure 31: LCOE for electroysis for different hydrogen configurations 59
Figure 32: Boxplot of electricity duals in NLD for different weather types 59
Figure 33: Comparison of electricity duals between the optimal cost scenario and 80% hydrogen share scenario 61
Figure 34: Electricity production by source vs Electricity shadow prices in the optimal scenario 62
Figure 35: Electricity production by source vs Electricity shadow prices (+50% fuel demand) 62
Figure 36: Price duration curve for different weather scenarios 63

Figure 37: Price duration curve for different weather scenarios with increased fuel demands 64
Figure 38: Load duration curve for different weather scenarios (optimal hydrogen share)........................ 64
Figure 39: Load duration curves for increased fuel demands 65

List of Tables

Table 1: Overview of all model runs for this research project 26
Table 2: Overview of all decision variables within the Calliope model adapted from Calliope (2022) 29
Table 3: Overview of the Python code to extract dual variables from Calliope 31
Table 4: Overview of the steps to convert Euro-Calliope to North Sea Calliope 36
Table 5: Overview of 2020 specific override files 37
Table 6: Overview of the monetary and production variables and its unit 39
Table 7: Overview of DelftBlue supercomputer setup 39
Table 8: Overview of the Gurobi solver options used 40
Table 9: Overview of the different hydrogen configurations 44
Table 10: Hydrogen configurations for the 2050 model 50
Table 11: Demand range within the load duration curves 65
Table 12: Overview payback time energy system for the different configurations 66

Abbreviations

Some of the abbreviations often used during this master thesis project

Abbreviations	
CCGT	*Combined Cycle Gas Turbine*
DHPC	*Delft High Performance Computing Centre*
ENTSO-E	*European association for the cooperation of transmission system operates (TSOs) for electricity*
EV	*Electric vehicle*
IDE	*Integrated Development Environment*
LCOE	*Levelized Cost of Energy*
LP	*Linear Programming*
O&M	*Operations and maintenance*
TES	*Total Energy Supply*
TSO	*Transmission System Operator*
VRES	*Variable renewable energy sources*

Executive summary

Although there is a common understanding that the use of variable renewable energy sources (VRES) is needed in our collective attempt to decarbonise society, the type of technology that we should deploy, and where is not so clear. Stakeholders from real-world projects use the outcomes from optimisation models to aid their decision-making process. One of such outcomes that decision-makers use is the price information of energy carriers in the required temporal and spatial resolution. In the current state, price information is embedded in the form of shadow prices within linear optimisation problems. As a result, price information is relatively easy to extract for conventional, power-sector focussed energy system models. However, when these energy system models are multi-carrier and sector-coupled in which energy carriers undergo several conversion stages, the extraction of price information becomes less trivial. Furthermore, even when price information is extracted in the form of shadow prices, they might not represent real-world price information. This master thesis research aims to develop a generally applicable price information generation method to extend the use of shadow-price based methods in conventional models to more complex multi-carrier fully sector-coupled models. The price information generation method is developed in Python which is tested within the Calliope modelling framework for a multi-carrier fully sector-coupled energy system. It does this by extracting the shadow prices from a linear optimisation problem of the North Sea Euro Calliope model which is adapted from the Euro-Calliope model. The shadow prices are then compared against current real-world prices for the energy carrier electricity. Results show that the shadow prices do not represent real-world prices accurately, however the use of shadow prices can be extended to understand trade-offs between different configurations of fully sector-coupled energy system models. A use case for the shadow prices has been conducted to analyse the price stability of Dutch electricity prices for different hydrogen shares within the energy system for different weather scenarios. Initial results show that the price stability of electricity in the Netherlands could be improved by increasing the share of hydrogen in the energy system. The increase of the hydrogen share within an energy system does not significantly affect the payback time of the energy system and the levelized cost of energy (LCOE) for electricity technologies. This research project shows that shadow prices could be used to understand trade-offs in different configurations of fully sector-coupled energy systems and aid the decision-making for the type and location of technologies to fulfil energy demands in the future. Recommended future research include an improvement of the North Sea Calliope model using a bottom-up approach and the analysis of other sectors within the fully sector-coupled energy system such as hydrogen and heat.

Chapter 1 – Introduction

This chapter introduces the research topic and problem of this master thesis research project. It starts with background information in section 1.1 to motivate the relevance of the problem followed by some of the core concepts that are relevant throughout the research project in section 1.2. In section 1.3 the knowledge gap is described and the main research question is formulated. Subsequently the sub questions are described in section 1.4. Section 1.5 briefly summarises the key deliverables of this master thesis project.

1.1 Background

The limited availability of fossil fuels, our ever-growing appetite for energy, and rising environmental concerns has left modern day societies with a challenging task. The Glasgow Climate Pact has reaffirmed the long-term global goal to hold the increase in the global average temperature to well below 2 ^{o}C above pre-industrial levels and to pursue efforts to limit the temperature increase to 1.5 ^{o}C above pre-industrial levels, recognizing that this would significantly reduce the risks and impacts of climate change. It also recognizes that limiting global warming to 1.5 ^{o}C requires rapid, deep and sustained reductions in global greenhouse gas emissions, including reducing global carbon dioxide emissions by 45 per cent by 2030 relative to the 2010 level and to net zero around mid-century, as well as deep reductions in other greenhouse gases (Glasgow Climate Pact, 2021).

The Netherlands has set a target to reduce greenhouse gas emissions by 49% by the year 2030 and 95% by the year 2050 compared to 1990 levels and 100% renewable electricity production by the year 2050. These legally binding targets have been laid down in the 2019 Climate Act (Klimaatwet, 2020).

Although there is a common understanding that the use of variable renewable energy sources (VRES) is needed in our collective attempt to decarbonise society, the type of technology that we should deploy, and where is not so clear. Optimisation models are being developed and increasingly being used to aid decision-makers in answering this challenging question. Although recent developments have been made to account for higher spatial resolution (Hörsch & Brown, 2017; Pfenninger & Keirstead, 2015; Lombardi et al., 2020) and sector coupling (Brown et al., 2018; Maruf & Islam, 2019; Mangipinto, 2020; Pavičević et al., 2020), state-of-the-art models still focus primarily (often exclusively) on minimising the total cost to society, measured in terms of total investment and operation cost for the deployed infrastructure. As a result, they fail to provide information about how much electricity, heat or hydrogen will cost at the point of consumption, even though these prices will be critical for the political and social acceptability of any energy scenario.

Detailed energy carrier price information of current energy modelling systems are generally only focussed on a single energy carrier, mostly electricity (Luz & Silva, 2021; Tröndle, 2020; Diaz et al., 2017; Laha & Chakraborty, 2021). Detailed price information for other energy carriers such as heat (Lombardi et al., 2019) and hydrogen (Morgenthaler et al., 2020) is available to a lesser extent or not combined within a fully sector-coupled energy system.

This master thesis project aims to bridge this gap by developing a method to extract price information in the form of shadow prices for fully sector-coupled energy systems. Furthermore, the method is then applied to answer a policy related research question to showcase the potential use cases of the energy carrier price information in the form of shadow prices.

1.2 Core concepts

This master thesis project is related to a design in a complex social-technical system. It includes a technical component operating from both the public and private domain. The most important core components are described in this sub section. The information on the core concepts has been gathered through short literature reviews. The main literature used are literature review papers sourced on Google Scholar and papers suggested by supervisors.

1.2.1 Energy system optimisation tools

Energy system optimisation tools are used to generate insights on energy systems on the supply and demand of energy (Pfenninger, 2014). They allow policymakers to explore the impact of their policies on the energy sector as well as assessing the efficacy of reaching certain policy targets by implementing their policy instruments (Lopion et al., 2021). In the past, previous studies are limited to a single sector analysis (mainly the power sector), reduced temporal and spatial resolution and limited time horizon of study (Aryanput et al., 2021). As the use of VRES has been increasing exponentially in the last years and production of energy are becoming more decentralised, higher spatial resolution in energy system optimisation models are needed for the efficient integration of these new resources (Martínez-Gordón et al., 2021).

Pfenninger (2014) distinguishes four model groups: energy system optimization models; energy system simulation models; power systems and electricity market models; qualitative and mixed-methods scenarios. This research will focus on energy system optimization models. Moreover, this research project focuses on the development of a method for the generation of price information. The developed method will then be applied to a fully sector-coupled energy system where the price information is used to analyse trade-offs between different configurations of energy systems.

1.2.2 Sector coupling

According to Maruf (2019, p22), "Sector coupling does not only refer to supply–demand relations but also considers the interlinkage between the consumption sectors like households, commerce, trade, services, industries, transports, etc. While the main objective of sector coupling is to reduce GHG emissions by substituting fossil fuels, the secondary objective is to provide flexibility, network optimization and increased efficiency to the energy systems". The additional flexibility provided by sector coupling proves to play an important role in providing cheaper and more efficient storage solutions, reducing the cost of the energy transition (Pavičević et al., 2020). In current energy optimization models, price information is embedded in the form of shadow prices within linear optimisation problems. As a result, price information is relatively easy to extract for conventional, power-sector focussed energy system models. However, when these energy system models are multi-carrier and sector-coupled in which energy carriers undergo several conversion stages, the extraction of price information becomes less trivial. Furthermore, although energy carrier prices in the form of electricity are represented quite extensively, non-electric carriers are not (Pickering et al., 2022). This research will therefore focus on fully sector-coupled energy system optimization models.

1.2.3 Energy carriers

Energy carriers allow the transport of energy from one place to another. The predominant energy carriers in conventional energy systems are in the form of hydrocarbons such as natural gas, coal and oil products. In the transition towards low-carbon energy systems, electricity, hydrogen and synthetic fuels are becoming more important and predominant as main energy carriers (Foxon et al., 2010; Møller et al, 2017). The key difference between the energy carriers found in conventional energy systems and energy carriers in future energy systems is the low emission of GHGs in the latter. Studies have shown that multi-energy carrier and sector-coupled energy systems such as power-to-heat, power-to-gas, power-to-hydrogen have the potential to increase the efficiency and reduce the overall cost of the whole energy system (Brown et al., 2018; Pavičević et al., 2020).

Power-to-heat

Power-to-heat options include centralised power-to-heat systems such as district heating. Centralised power-to-heat systems convert heat at a central location away from where the actual heat demand is. In contrast, for decentralised power-to-heat systems, use electricity to create heat at the location where the heat demand is. Decentralised power-to-heat options include direct heating systems such as electric heaters. Moreover, thermal energy storage could be present in both centralised and decentralised power-to-heat systems. Thermal energy storage options include heat pumps and hot water storage (Bloes et al., 2018).

Power-to-gas

Power-to-gas refers to the process of using electricity to create hydrogen or natural gas with hydrogen admixture through electrolysis or the creation of natural gas with methane admixture through a process called methanation (Schiebahn et al., 2015).

This research will focus on multi-carrier fully sector-coupled energy systems. Considering the limited time available within this research project, the scope is of this project focusses on hydrogen more than other energy carriers. Hydrogen as an energy carrier and its position in the Dutch vision for future energy systems is elaborated further in the sub sections below.

Hydrogen

Hydrogen as an energy carrier can be stored, transported and it can be used as a fuel or it can be converted to electricity energy using a fuel cell. When hydrogen is used within a fuel cell to generate electricity, the only emissions are in the form of water and warm air (Mazloomi & Gomes, 2012). It is therefore often seen as an important energy carrier in the global decarbonisation efforts (Chapman et al., 2019; Rosen & Koohi-Fayegh, 2016; Møller et al, 2017). While the use of hydrogen is environmentally benign, the production of hydrogen might not always be free of GHG emissions. It is therefore important to note the distinction between green, blue and grey hydrogen. Green hydrogen is produced by fully renewable energy such as solar and wind using process called electrolysis. For the production of blue hydrogen, natural gas is needed where the released carbon dioxide is captured and stored underground. Grey hydrogen uses the same process as blue hydrogen except that the released carbon dioxide is not captured and is released into the atmosphere (TNO, 2022).

Hydrogen in the Netherlands

The Netherlands has strong ambitions on the medium-term (2030) and long-term (2050) to use hydrogen as an energy carrier for a number of key infrastructures within the Dutch energy system. Among others, hydrogen will be used as carbon-free feedstock for the industrial processes with the long-term goal to replace all feedstock in the chemical industry with carbon-free hydrogen. Hydrogen will also be used for high temperature industrial processes exceeding 600 degrees Celsius. Furthermore, the Dutch Climate Accord also states that hydrogen will also be used as a means for long-duration storage and used to decarbonise the mobility sector, particularly in the long distance and heavy transport options (Ministerie van Economische Zaken en Klimaat, 2019; Nationaal Waterstof Programma, 2022). The Netherlands has the ambition to use blue and green hydrogen for the energy transition (TNO, 2022). Furthermore, the Dutch government has ongoing studies to assess the feasibility of implementing the production of green hydrogen in combination with offshore wind farms in the North Sea region (RVO, 2022)

1.3 Research question

The knowledge gap within this research project is a void in modelling framework from which energy carrier prices cannot be extracted in the desired manner from multi-carrier fully-sector coupled energy optimisation models. Therefore, the objective of this research is to develop a method for the generation of energy-carrier prices within multi-carrier, fully sector-coupled energy systems models. In addition, this master's thesis project also aims to provide an answer to how the extracted price information can be used in the policy domain. As this master thesis project focusses on the energy transition in Netherlands which has a strong ambition for the implementation of hydrogen-rich energy scenarios, the policy related part will address hydrogen related scenarios.

Therefore, the main research question of this research project is:

How can energy carrier price information be used to understand the trade-offs between different hydrogen configurations of a fully sector-coupled energy system model.

1.4 Sub questions

In order to answer the main research question, the following sub questions have been formulated:

1. What are the methods to extract energy carrier prices from fully sector-coupled energy systems in existing literature?

2. How can energy carrier prices in the form of shadow prices be extracted within the Calliope framework in fully sector-coupled energy systems?

3. How do the generated price information compare to the real-world price information?

4. What are the price-related trade-offs when varying the share of hydrogen in a fully sector-coupled energy system?

The sub questions have been drafted in such a way that they correspond to the flow of the design stages. These will be described further in detail in chapter 2.

1.4.1 Research methods and research flow diagram

The main resources used for the literary sources and literature review in this master thesis project are literary databases such as Google Scholar, governmental sources such as RVO and research organisations such as TNO. Since the Calliope modelling framework is Python-based, development hubs such as GitHub, Stack Overflow will be used as the main source for learning and developing the Calliope models. For validation processes of electricity prices, market data will be used from ENTSO-E. An overview of the sub questions and the main tools and resources used to answer these sub questions is shown in Figure 1. For the computation of the large complex models, the Delft High Performance Computing Centre (DHPC) is used.

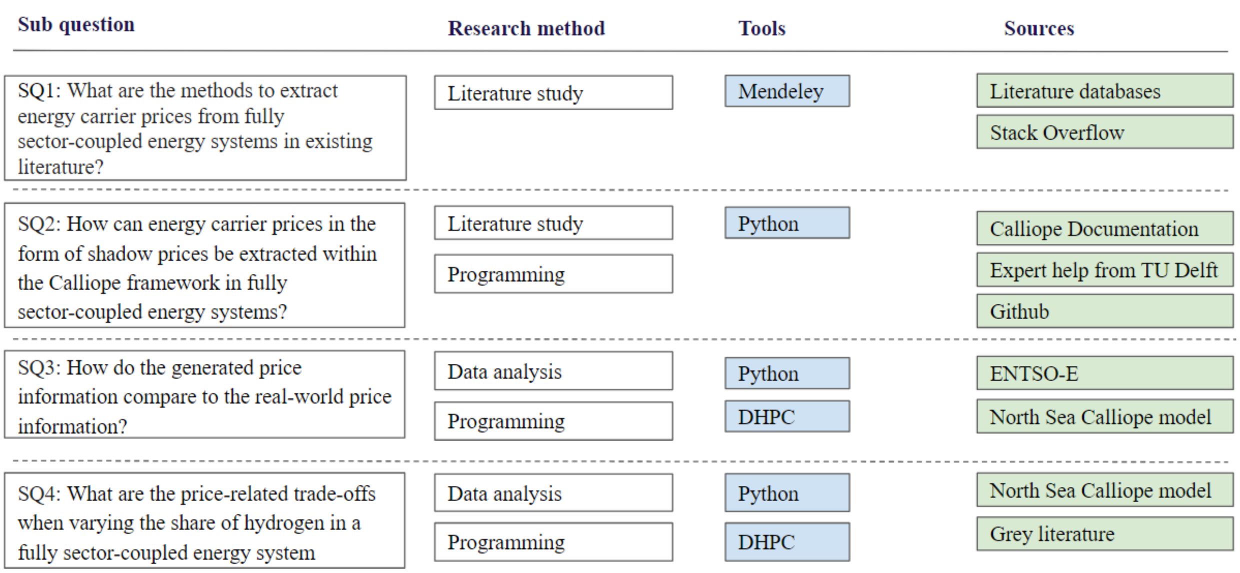


Figure 1: Research methods, tools and sources

1.4.2 Research flow

The overall research flow of this research project in correspondence of the design stages is shown in Figure 2. The outputs of every design stage will be used in the subsequent design stage. At the end of all three stages, the design objective shall be met.

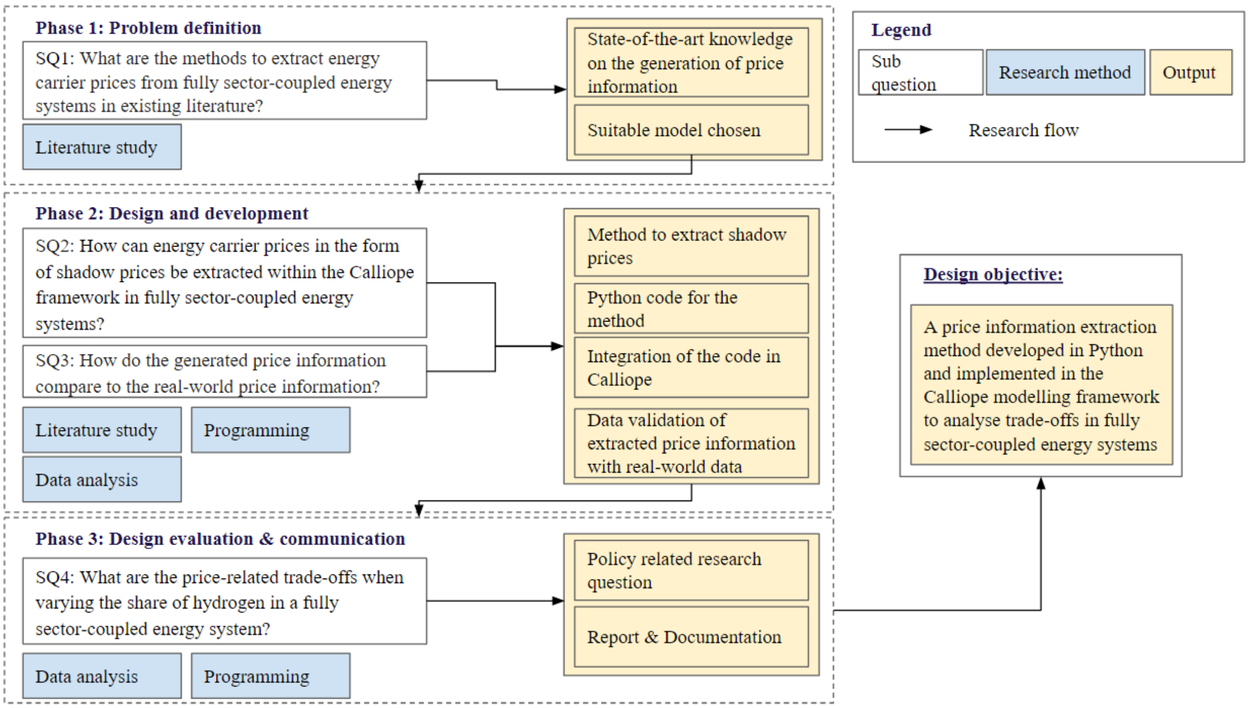


Figure 2: Research flow diagram

1.5 Deliverables

In short, the research project can be divided into three deliverables. First, a literature review on the state-of-the-art knowledge on regarding the generation of price information in multi-carrier and sector-coupled energy models. Second, a state-of-the-art method to extract price information from multi-carrier and sector-coupled energy models within the Calliope framework. Third, a case-study to understand trade-offs within fully sector-coupled models using the price information.

Chapter 2 – Research methodology

In this chapter, the research methodology for this master thesis project is described. In section 2.1, phase 2 of the research project is described. Then in section 2.2, phase 2 of the research project is described. In section 2.3, phase 3 of the research project is described.

Research phases

The project will be carried out using a design approach following the typical three stages of a design cycle. The stages are in sequential order the *problem definition* stage, the *design and development* stage and at last the *design evaluation and communication* stage. In the first stage which is set up to be more exploratory has the goal of answering the first sub question. The second stage is where the actual development starts from which we can answer sub question 2 and sub question 3. The third and final stage aims to answer sub question 4 using the developed method during the second stage. The total allocated time for the research project is 24 weeks.

The process of this research project can be delineated into three logical milestones. The first milestone is reached when the shadow prices can be extracted from a multi-carrier sector-coupled model. As mentioned in chapter 1, even when the extraction of shadow-prices is successful, the obtained numbers might not be representative to real-world data. Therefore, a second milestone will be reached when the extracted shadow prices are evaluated against real-world data. After the second milestone, an additional step will be performed that focusses on the useability of the extracted shadow prices. Therefore, to conclude the third milestone, the extracted price information will be used for a policy related research question to assess the validity and useability of energy carrier price information. The phases of this research project along with the key tasks associated within each phase is presented in Figure 3.

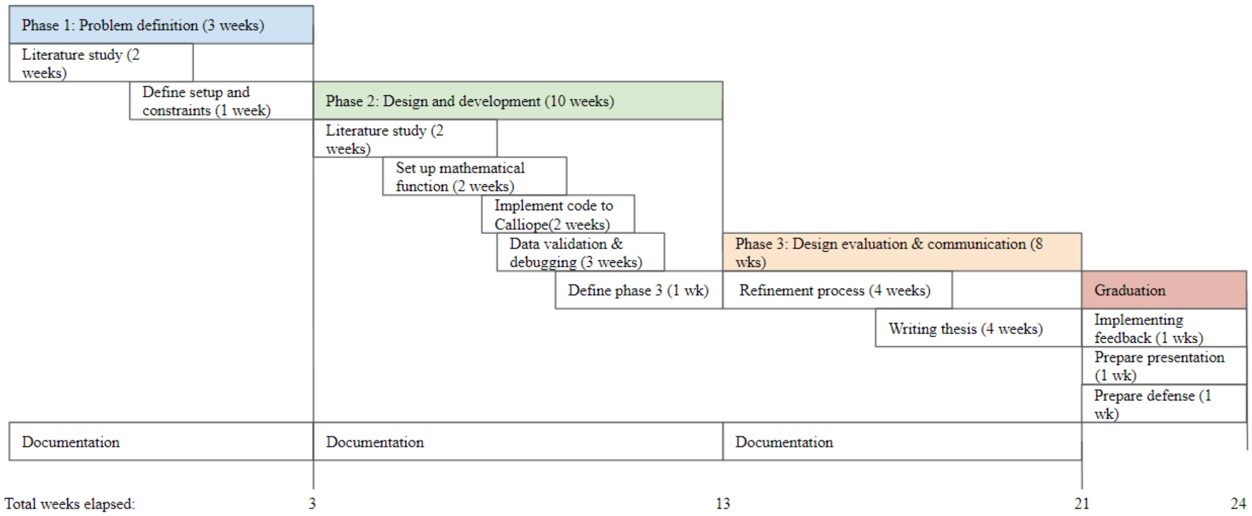


Figure 3: Overview of the design stages and the associated key tasks

2.1 Phase 1: Problem definition

Sub question 1 aims to find state-of-the-art knowledge regarding the generation of price information of complex multi-carrier sector-coupled energy systems through literature study. Potential starting points are the use of shadow prices in multi-carrier sector-coupled models within literary papers, forums such as the Open Energy Modelling Initiative and documentation within the Calliope framework.

2.1.1 Linear programming problem and shadow prices

As mentioned in Section 1.2.1, this research project focusses on energy system optimisation models. Within these optimisation models, different optimisation techniques are available. This research project focusses on the linear programming (LP) method as the Calliope model that is used in this project is defined as a linear programming problem.

A LP problem is a constrained optimisation problem. The objective function is a linear expression for which the maximum or minimum value has to be found given a number of linear constraints. A LP problem typically contains the following three elements:

1. Objective function
2. Decision variables
3. Equality and/or inequality constraints

Where the objective function $Z(x)$ can be formulated as:

$$Max\ Z(x) = \sum_{j=1}^{n} C_j x_j \tag{1}$$

subject to a number of linear constraints of the form:

$$\sum_{j=1}^{n} a_{ij} x_j \ \leq b_i, \qquad i = 1, \dots, m \tag{2}$$

where the non-negative variables x_n are called the decision variables:

$$x_j \geq 0, \qquad j = 1, \dots, n \tag{3}$$

The largest or smallest value possible is the optimal value for the objective function depending on whether the goal is to maximise or minimise. The collection of the decision variables that gives the optimal value is then called the optimal solution.

2.1.2 Duality in Linear Programming Problems

A LP problem possesses the property of duality. The duality principle is a mathematical concept that enables the extraction of information of a mathematical structure using the information of another mathematical structure (Diewert, 1974). In relation to a LP problem, the first mathematical structure is the so-called the primal problem, the second mathematical structure is consequently called the dual problem.

The equations (1), (2) and (3) as such can be seen as the primal problem and subsequently the dual problem can be formulated as:

$$Min\, J(y) = \sum_{i=1}^{n} b_i y_i \tag{4}$$

subject to:

$$\sum_{i=1}^{n} a_{ij} y_i \ \geq C_j, \qquad j = 1, \dots, n \tag{5}$$

$$y_i \geq 0, \qquad i = 1, \dots, m \tag{6}$$

It can be seen that the objective function is inverted. The objective function in the primal problem is to maximise the outcome whereas the objective function of the associated dual problem is to minimise the outcome. Furthermore, the decision variables in the primal problems become the constraints in the dual problem and the constraints in the primal problem become the decision variables in the dual problem.

The dual variables for the primal problem can be interpreted as the marginal cost of the primal problem (Perry & Crellin, 1982) as they represent the change of the optimal value in the objective function to a unit increase of the right-hand side of the associated constraint equation. Consequently, the optimal values of the decision variables of the dual problem are equal to the shadow prices of the primal problem. The opposite also holds true and thus the shadow prices of the dual problem are equal to the solution of the primal problem.

In linear programming, shadow prices are often used to identify the maximum price one should pay to obtain an additional unit of a constraint resource (Perry & Crellin, 1982). From the primal problem described above, it can be said that if constraint b_i is changed by one additional unit, the change in $Z(x)$ gives us the shadow price of the respective constraint.

In the literature (Lee & Zhang, 2012; Wei & Liu, 2013; Althammer & Hille, 2016), shadow prices are often used to assign monetary values to non-marketed resources in economic appraisals. Within the energy sector, it is often used to account for costs in emission related energy carriers such as CO_2 emissions or other greenhouse gas emissions.

2.1.2 Calliope

Calliope is a Python-based open-source multi-energy modelling and linear optimization framework with high temporal and spatial resolution. This master thesis project will therefore be conducted around the Calliope framework due to its high flexibility and high-resolution capabilities. Furthermore, the Calliope framework is currently being used in several EU-funded projects with real-world stakeholders making it a highly relevant framework for research.

In the very basics, the Calliope model is built from YAML files. These YAML files contain definitions and constraints for the technologies and locations within the model. These YAML files combined with demand profiles for all energy carriers form the basis of the model. By running the model, Calliope solves the linear optimisation problem by minimising or maximising the total system cost given the constraints. The full documentation on Calliope can be found on their official website and official publication (Pfenninger & Pickering, 2018).

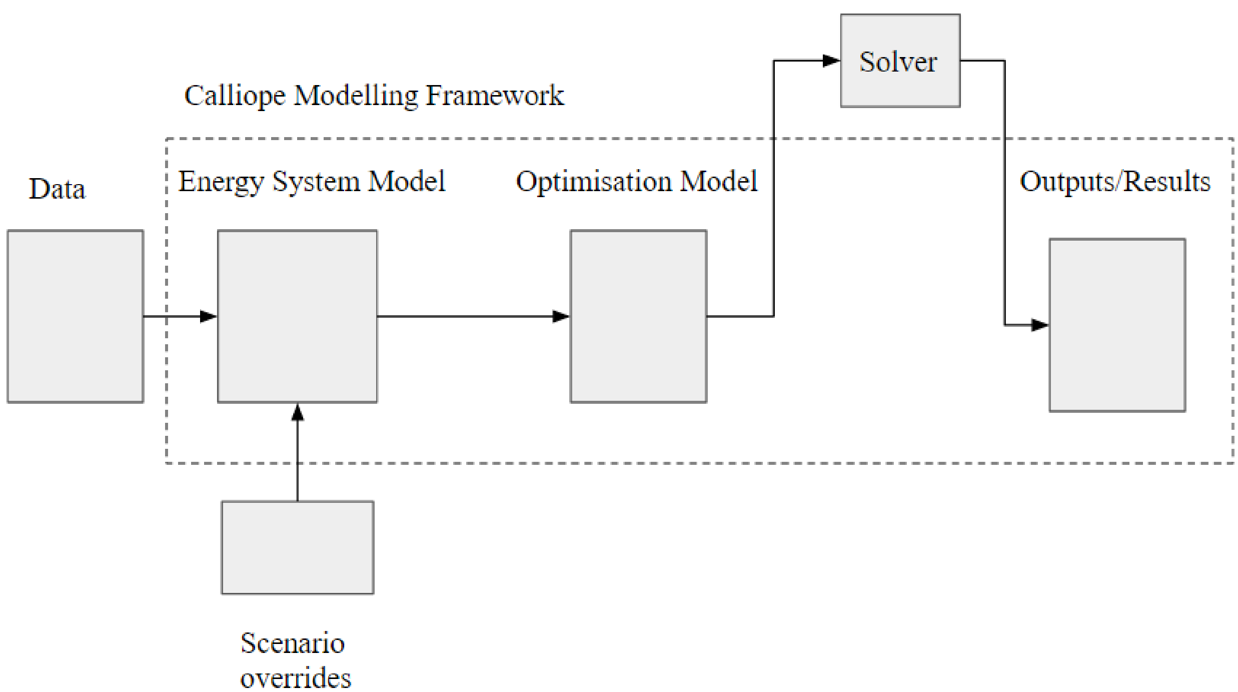


Figure 4: Simplified overview of the Calliope Modelling Framework

Figure 4 represents a simplified overview of the Calliope Modelling Framework adapted from Luz & Silva (2021). The modelling framework indicated by the dashed box contains the following main elements:

1. **Energy System Model**. This includes the geographical representation such as locations of every node within the model; The technologies present within the model such as demand technologies, supply technologies, transmission technologies and conversion technologies; The network representation, such as the links and connections between the nodes.
2. **Optimisation Model.** This is where the objective function is set along with the decision variables, constraints and solver options
3. **Outputs/Results.** This includes timeseries data on carrier flows, resource usage, variable costs, installed capacities, production and consumption profiles.

Outside the Calliope Modelling Framework, external data is used to define the coordinates of the nodes, specifications for the technologies (such as production capacities and costs) and transmission capacities for the links. Scenario overrides are used to define specific scenarios applicable to the model such as different supply and demand profiles for different years. A solver is then used to solve the LP problem given the configuration within the optimisation model from which the results can be extracted within the calliope modelling framework.

In chapter 4, the exact Calliope model used for this master thesis project is further described.

Other fully sector-coupled modelling frameworks

Besides Calliope, there exists also other Python-based energy modelling frameworks suitable for fully sector-coupled modelling such as the Open Energy Modelling framework (oemof) and Python for Power System Analysis (PyPSA) (Ringkjøb, 2018). The reason for choosing the Calliope framework for this research project is due to the fact that method developed for the extraction of price information can also be applied to any other Python-based framework. More importantly, due to expertise in the Calliope framework from this thesis' supervisor, it is possible to get full support with already full-scale models available such as the Euro-Calliope model.

2.2 Phase 2: Design and development

Sub question 2 coincides with the *design and development* stage in which the method to extract energy carrier prices is to be developed. In subsequence of sub question 1, the developed method shall be modified and implemented for the Calliope models within the Calliope modelling framework.

Sub question 3 aims to validate the developed method against real-world price information. A way to assess the accuracy of the developed method is to assess the outcome with known and real-world prices. Therefore, it seems logical to perform this research project for a model which reflects the current energy infrastructure with a good availability of data. The Dutch energy system is therefore a good candidate as they have good open-source data from recent years due to databases available such as on ENTSO-E and EPEX SPOT. Furthermore, defining the model constraints and geographic boundaries will be an important part of this sub question. In order to save time, it is an advantage to find existing models rather than building a model from scratch. The model should be a multi-carrier fully sector-coupled energy system. The base model used is the euro-calliope-2.0

model (GitHub, 2022) which is a fully sector-coupled model including transport, heat and industry sectors with their associated carriers and technologies.

2.2.1 Working environments

Throughout this master thesis project, there are mainly two environments on which the research will be conducted. The workflow and interaction between the working environments is shown in Figure 5.

Since the Calliope modelling framework is developed in Python, the Integrated Development Environment (IDE) Spyder is used to develop, edit and debug the main python scripts to run the desired Calliope models. The same Spyder environment can also be used for data analysis as it has an integrated variable explorer and plots can be created. The Spyder environment alone on a laptop is powerful enough to perform and analyse low resolution runs. The full specifications of the laptop used during this project can be found in Appendix A – Laptop specifications. For higher resolution runs, the Delft High Performance Computing (DHPC) DelftBlue is used. To access the DHPC, the secure remote access software Bitvise SSH client is used to manage the file transfers between the local computer and the DelftBlue supercomputer. A sbatch script is then sent to the DelftBlue supercomputer to run the Calliope model. After successfully run of the models, DelftBlue returns a slurm output along with a netcdf file that contains the optimal values for the simulation as well as the dual variable values stored in a .csv file. These two files are then imported back to the Spyder IDE from which the data analysis is performed.

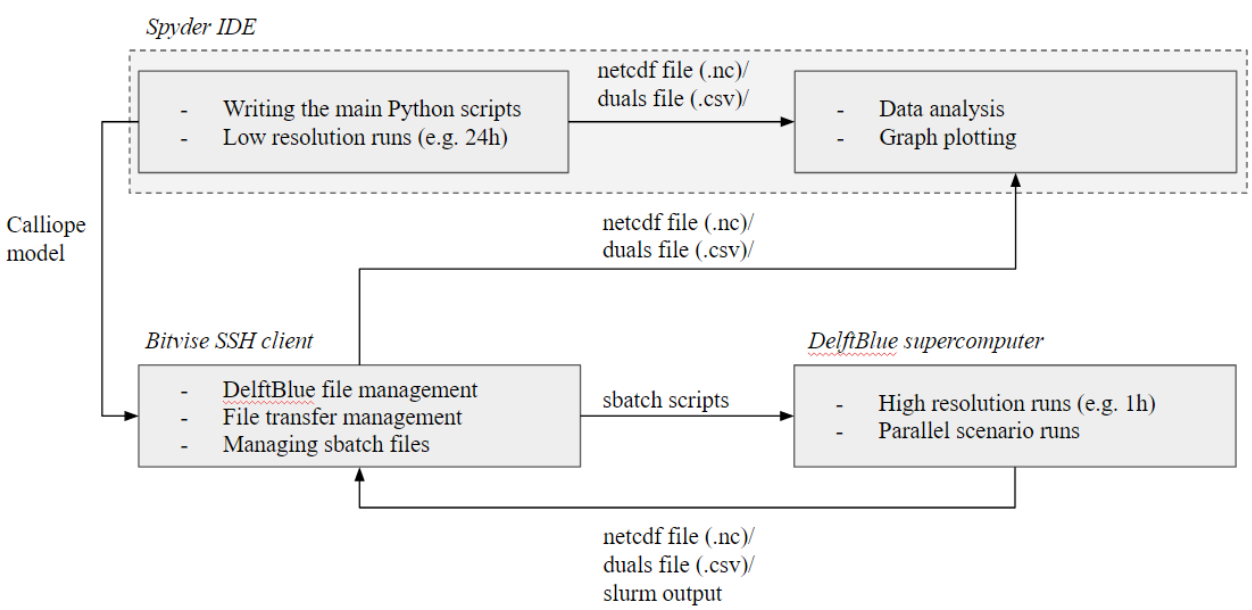


Figure 5: Overview of the interactions between different working environments

Working environment limitations

The DelftBlue supercomputer is limited to a maximum running time of 24 hours for each job with a total storage capacity of 8GB. For high memory jobs the DHCP has six nodes available with 750GB and four nodes with 1.5TB of RAM (GitLab, 2022).

2.2.2 Modelling process

The modelling process can be described the following three distinct phases, *Building the model, Running the model* and *Analysing the results*. In the first stage, all the necessary files such as the YAML files, override files and demand profiles are loaded into the model. YAML files are used to define technologies, locations, constraints and costs variables. Override files are used to define year specific data such as the demand for hydrogen. One can imagine that the demand for hydrogen for example would be a lot more in 2050 than in 2020. After the model is built, the model is then run. During this step the solver is solving the LP optimisation problem and finds time optimal values for all the decision variables for the given objective function. During this same step the dual variables and its values are also extracted. The model results and dual variables are then both stored. The analysis of the results can be divided into two distinct steps. In the first step, the physical flows within the model such as carrier production and carrier consumption are analysed to make sure the model is a feasible and representative model. Once the physical model has been validated, the energy carrier prices are then analysed using the dual variables. Finally, the results are plotted and presented using charts and graphs.

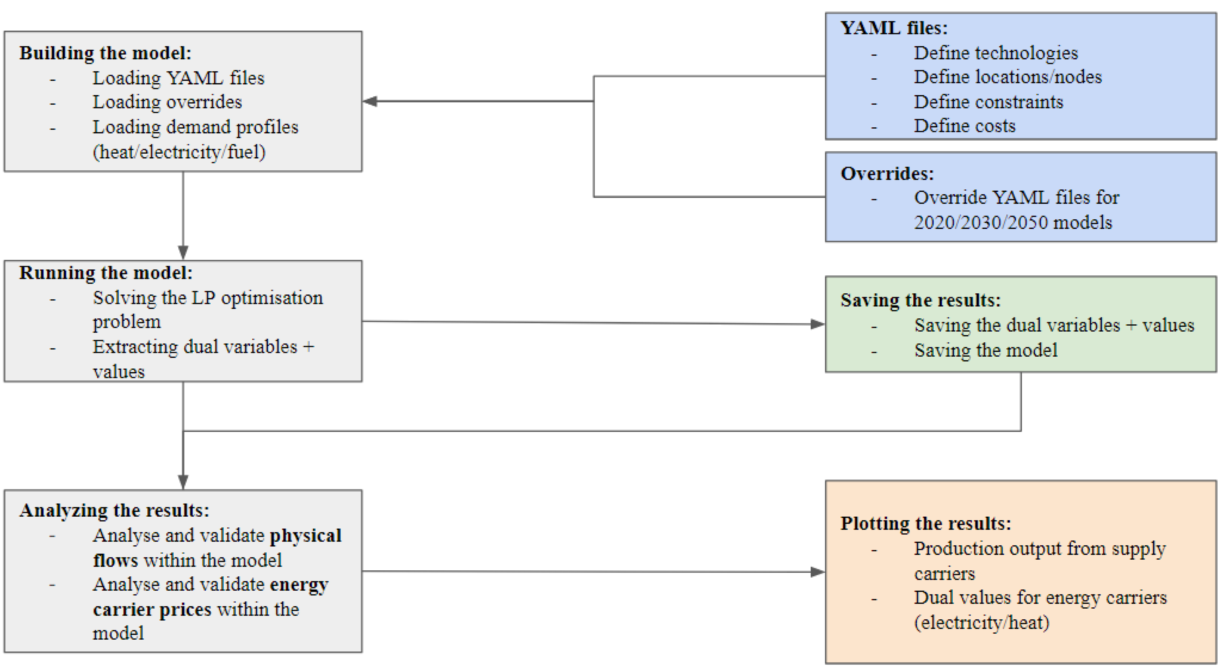


Figure 6: Overview of the modelling process

An overview of all the models run for this research is presented in Table 1. In order to capture the daily and seasonal effects of VRES, the resampling of the time resolution for all the models is set on 1 hour. The *Simple* model has its goal to test the extraction method for energy carrier prices within the Calliope modelling framework. This process is described further in detail in chapter 3. The 2020-1h model will be used in phase 2 for the validation process. The year 2020 has been chosen so that model data can be validated against real-world data. Technology data include technical constraints such as energy capacity, energy efficiencies and lifetime of the respective technologies. Moreover, monetary constraints such as cost initial investment cost and annual operation and management costs are also included. Important to note is that in the 2020 model, hydrogen technologies, synthetic fuel production and batteries are not included as these are not yet widely adopted. In the 2050 models, the models in which the names start with *2050* do have these technologies enables as it is expected that these technologies will be widely adopted in 2050. Both the 2020 model and 2050 models uses technology data from the Danish Energy Agency. Weather data from 2020 is at the moment of research not available and thus weather data for the year 2015 in coherence with the base year of the Euro-Calliope model. Furthermore, both the 2020 and 2050 model are based on the North Sea Calliope model which is built from the Euro-Calliope model. The building of the North Sea Calliope model will be described in chapter 4.

Table 1: Overview of all models runs for this research project

<table>
<tr><th>Phase</th><th>Model name</th><th>Technology data</th><th>Weather data</th><th>Hydrogen technologies</th><th>Synthetic fuel production</th><th>Batteries</th></tr>
<tr><td rowspan="2">2</td><td>Simple</td><td>NA</td><td>NA</td><td>No</td><td>No</td><td>No</td></tr>
<tr><td>2020-1h</td><td>2020</td><td>2015</td><td>No</td><td>No</td><td>No</td></tr>
<tr><td rowspan="9">3</td><td>2050-2010</td><td rowspan="9">2050</td><td rowspan="3">2010</td><td rowspan="9" colspan="3">Yes</td></tr>
<tr><td>2050-2010-8030</td></tr>
<tr><td>2050-2010-8030</td></tr>
<tr><td>2050-2015</td><td rowspan="3">2015</td></tr>
<tr><td>2050-2015-8030</td></tr>
<tr><td>2050-2015-8050</td></tr>
<tr><td>2050-2016</td><td rowspan="3">2016</td></tr>
<tr><td>2050-2016-8030</td></tr>
<tr><td>2050-2016-8050</td></tr>
</table>

The different variations of the 2050 models are to assess trade-offs of different configurations of the energy system which will be part of phase 3 of this research project. This will be elaborated further in the next subsection.

2.1.3 Phase 3: Design evaluation and communication stage

Sub question 4 coincides with the last design stage, the *design evaluation and communication* stage. It includes a refinement process of the developed method for the generation of price information. This refinement process is intended improve the meaningfulness of the extracted data. Therefore, for the third phase, a research scope is introduced to address the policy relevant research question. To cope with the interest of the Dutch energy transition and its high ambitions for a hydrogen-based energy system in the future, the North Sea region has been selected with an analysis of the supply and demand of energy in the Netherlands. This reduces the computational load and allows this research project to fit within the time limit of this master thesis project. A more detailed description of the North Sea region model can be found in chapter 4. In chapter 5, the trade-offs of different configurations of a fully sector-coupled energy system is analysed through varying the share of hydrogen within the energy system. Within this analysis, technical parameters such as infrastructure and capacity deployment are analysed. Economic parameters such as levelized cost of energy, cost recovery and price deviation of electricity prices in the different configurations are also analysed. The model results are then compared to literature and existing plans for offshore wind energy and hydrogen production in the North Sea. The comparison of model results and literature are discussed in chapter 6. The conclusion and recommendations are then finally described in chapter 7. Furthermore, the writing of the full thesis report and the preparation of presentation as part of the communication stage is also an important part of phase 3.

Chapter 3 – Extracting price information from Calliope

This chapter will provide the results from sub questions 1 and 2 regarding the extraction of price information from Calliope. It includes a description of how the method works and its application is tested on the tutorial models from Calliope.

3.1 Literature review on extraction methods

Since this research project is conducted around the Calliope framework, the scope of the literature review is reduced to the finding of existing shadow price extraction methods suitable for the Calliope framework. Moreover, the developers from the Calliope modelling framework recommend the use of Gurobi or CPLEX as these are significantly faster than the open-source solvers GLPK and CBC (Calliope, 2022). The Gurobi solver has been chosen over the CPLEX solver due to the already existing integration of Gurobi in the sector-coupled Calliope model of this research project. The Gurobi license is free for academic purposes. Therefore, the scope of the literature review on extraction methods is further reduced to finding existing shadow price extraction methods for the Gurobi solver.

The idea of extracting shadow prices from LP problems is not novel. A search in Stack Overflow, a public platform where developers share programming codes, the term "get dual problem python" returns a total of 80 results. However, the extractions of shadow prices in Calliope returned a total of 0 results. This is due to the relatively limited research being performed using the Calliope modelling framework. From Stack Overflow, a reference to the Pyomo could be found which is a class within the Python language for optimisation modelling. The documentation of Pyomo then shows the Python script through which the dual values of a LP problem could be accessed (Pyomo, 2022). The Python script can then be applied to the Calliope Modelling framework. Other optimization modelling libraries have also been found such as SciPy and PuLP. The Calliope modelling framework is built using the Pyomo optimization library, therefore the use of other modelling libraries have not been researched extensively in this research project.

3.2 Mathematical formulation of the Calliope modelling framework

As mentioned in section 1.2.4. Every LP problem contains the main following main elements, the *Objective function*, *Decision variables* and *Constraints*. The formulation of these elements are further described in this subsection. It is important to note that only the most relevant formulations for this research project are elaborated upon. The full mathematical formulation of Calliope can be found in the documentation on the Calliope website.

3.2.1 Objective function

The objective function in Calliope is to minimise total system cost for specified cost class or set of cost classes and can be formulated as follows:

$$min: z = \sum_{loc::tech_{cost,k}} (cost(loc :: tech, cost - cost_k) * weight_k + \sum_{loc::carrier,timestep} (unmet\_demand(loc :: carrier, timestep) * bigM) \quad (7)$$

where k denotes the cost class.

3.2.2 Decision variables

Table 2 presents an overview of all the decision variables in Calliope. The decision variable *carrier_prod* represent the total energy production from all energy carriers within an energy system. It is therefore the most important decision variable for this research project as the goal is to analyse the marginal price of energy carriers of the energy system. The energy carriers are available within the dimension *carriers_prod* of *carrier_prod*.

Table 2: Overview of all decision variables within the Calliope model adapted from Calliope (2022)

Decision variable	Variable name in Calliope	Dimensions
Energy capacity	energy_cap	loc_techs
Carrier production	carrier_prod	loc_tech_carriers_prod, timesteps
Carrier consumption	carrier_con	loc_tech_carriers_con, timesteps
Cost	cost	costs, loc_techs_cost
Resource area	resource_area	loc_techs_area,
Storage capacity	storage_cap	loc_techs_store
Storage	storage	loc_techs_store, timesteps
Resource consumption	resource_con	loc_techs_supply_plus, timesteps
Resource capacity	resource_cap	loc_techs_supply_plus
Carrier export	carrier_export	loc_tech_carriers_export, timesteps
Variable cost	cost_var	costs, loc_techs_om_cost, timesteps
Investment cost	cost_investment	costs, loc_techs_investment_cost
Purchased	purchased	loc_techs_purchase
Units	units	loc_techs_milp
Operating units	operating_units	loc_techs_milp, timesteps
Unmet demand	unmet_demand	loc_carriers, timesteps
Unused supply	unused_supply	loc_carriers, timesteps

3.2.3 Constraints

The most important constraint for the analysis of shadow prices of energy carriers is the energy balance constraint. The energy balance ensures that, within each location, the production and consumption of each carrier is balanced. After all, the shadow price of an energy carrier is the marginal increase of the objective function through the marginal increase of the production of the associated energy carrier.

The energy balance is formulated as:

$$\sum_{loc::tech::carrier_{prod} \in loc:carrier} \boldsymbol{carrier_{prod}}(loc :: tech :: carrier, timestep) + \sum_{loc::tech::carrier_{con} \in loc:carrier} \boldsymbol{carrier_{con}}(loc :: tech :: carrier, timestep) + \sum_{loc::tech::carrier_{export} \in loc::carrier} \boldsymbol{carrier_{export}}(loc :: tech :: carrier, timestep) \tag{8}$$

$$\forall loc :: carrier \in loc :: carriers, \forall timestep \in timesteps$$

3.3 Extraction method

This section describes the process for the extraction method. When running the Calliope model, the solver is solving the primal and dual problem simultaneously, however the dual variables are not stored. In order to store the dual variable, the following steps are needed:

1. Build the Calliope model without running it
2. Create a dictionary variable to store the dual variables
3. Load and display all dual variables

After the third step, all dual variables are stored in a dictionary variable called *duals* containing data frames with the dual variables for all carriers. As described in section 3.1, the relevant dual variables for this research are the dual variables in the energy balance constraint which has the name *system_balance_constraint*. These dual variables are then stored it in a variable called *system_balance_duals*.

However, the data in the *system_balance_duals* data frame were not yet presented in the right format as the energy carriers could not be analysed separately from each other. A new data frame is created with the columns *region, carrier, timestep, dual-value*. After the reformatting, the data frame has the format as presented in Figure 7.

Index	region	carrier	timestep	dual-value
0	SWE	electricity	2015-01-01 00:00:00	0.00260523

Figure 7: Duals data frame after reformatting

The new data frame now allows for the analysis of each energy carrier separate from each other. An overview of all the functions and code needed to extract price information from Calliope can be found in Table 3.

Table 3: Overview of the Python code to extract dual variables from Calliope

Function	Code
Building the model without solving it	model.run(build_only=True)
Creating a dictionary variable to store the dual variables	model._backend_model.dual = pyo.Suffix(direction=pyo.Suffix.IMPORT)

Display all dual variables	duals = {}
	for c in model._backend_model.component_objects(pyo.Constraint, active=True):
	duals[("{} Constraint".format(c))] = []
	for index in c:
	duals["{} Constraint".format(c)].append(("{}".format(index), model._backend_model.dual[c[index]]))
	duals["{} Constraint".format(c)] = pd.DataFrame(duals["{} Constraint".format(c)])
	system_balance_duals = duals['system_balance_constraint Constraint']

Isolating system balance duals	system_balance_duals = duals['system_balance_constraint Constraint']
Reformatting the data frame of the system balance duals	def process_system_balance_duals(system_balance_duals): column=system_balance_duals[0] info = column.str.split("[(::)]") info = pd.DataFrame(info.tolist(), index= info.index) info.drop([0,2,5],axis=1,inplace=True) car_and_time = info[3].str.split(",") info[['car','time']] = pd.DataFrame(car_and_time.tolist(), index= info.index) info.drop([3,4],axis=1,inplace=True) info.columns = ['region','carrier','timestep'] info.region = pd.DataFrame(info.region.str.split("[']").to_list(), index=info.index)[1] info.carrier = pd.DataFrame(info.carrier.str.split("[']").to_list(), index=info.index)[0] info.timestep = pd.DataFrame(info.timestep.str.split("[']").to_list(), index=info.index)[1] info.timestep = info.timestep + ':00:00' info.timestep = pd.to_datetime(info.timestep) info['dual-value'] = system_balance_duals[1] return (info)

3.4 Testing on simple Calliope models

The developed method is first tested on smaller models. To keep things simple the method is applied on a single-region, non-sector coupled model with only the power sector available. The following simple model shown in Figure 8 has been created.

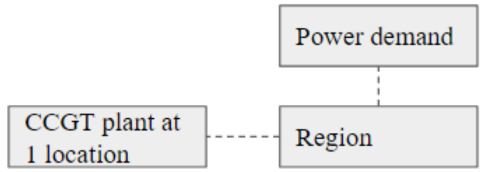


Figure 8: Overview of the simple model

The simple model contains a region with a certain power demand profile and 1 combined cycle gas turbine (CCGT) plant at 1 location within the region. The full simple model can be found in Appendix B1 - Simple model. The CCGT has a maximum production capacity of 36kW and the power demand has at times a demand of great than 36kW. This is to analyse the effects of what would happen in case of unmet power demand. As there are no competing power production plants, it is expected that the dual value will be equal to the variable costs of the power plant. Furthermore, it is expected that the dual value would go to infinity when there is unmet demand as there is no power production capacity to fulfil the power demand and the value of power would skyrocket.

The shadow price for power, power demand and power production of the CCGT are then plotted in Figure 9. It can be seen that as the power production from the CCGT plant is unable to fulfil the demand, the shadow price goes to 1e6, which is the maximum value possible within objective value range. For all other timesteps, the dual value remain constant at 10 €/kWh which matches exactly the set operational cost for the CCGT plant.

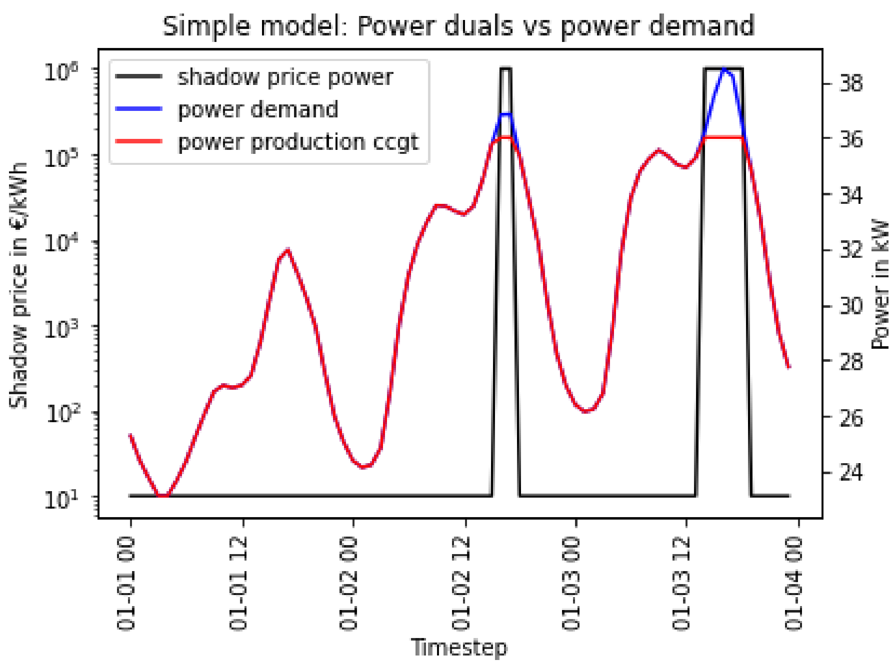


Figure 9: Assessing power duals against power demand from the simple model

Chapter 4 – Price information in a fully sector-coupled Calliope model

This chapter describes how the method developed in chapter 3 is now applied to a fully sector-coupled energy system model. Section 4.1 described the process of building a suitable Calliope model for this master thesis project. Section 4.2 Describes the running process and section 4.3 concludes with the validation process of the model. The Python scripts used for the analysis for the 2020 model can be found in Appendix S1.

4.1 Building the model

The developed method described in chapter 3 is applied to a multi-carrier and fully sector-coupled model. This section describes how the multi-carrier and fully sector-coupled Euro-Calliope model forms the basis of the North-Sea Calliope model which will ultimately become the main model for the validation and analysis aspects of this research project.

4.1.1 Europe-Calliope

The first European model in Calliope was developed by Tröndle et al. (2020) to analyse the European power sector. However, due to its focus on fully renewable electricity, the model did not have a realistic transmission topology. Pickering et al. (2022) have upgraded this European model to include all energy sectors with a realistic transmission topology. It includes all established and already commercially available supply and demand technologies for different sectors and regions. The 35 countries are represented by 98 nodes through which different energy carrier flows such as power, heat, hydrogen, synthetic hydrocarbons and biofuels can be analysed on an hourly resolution for an arbitrary year. The Euro-Calliope model contains 13 distinct carriers. From Pickering et al. (2022, p.15), these are "electricity, hydrogen, CO_2, liquid and gaseous hydrocarbons (kerosene, methanol, diesel and methane), solids (residual biofuel and municipal waste), low-temperature heat (combined space heat and hot water, and cooking heat), and vehicle distance (heavy-and light-duty road vehicles)." These carriers can either be consumed by demand, transport or transformation technologies; produced by heat, legacy or renewable technologies; stored by storage technologies. It is also important to mention that trading of energy carriers outside the Euro-Calliope region is not enabled. The demand is fully met by the supply within the region. The YAML files containing the definitions and specifications of these technologies can be found in Appendix B2 - North Sea Calliope model. An overview for the transmission and node network for the Euro-Calliope model is presented in Figure 10.

The demand data for the Euro-Calliope are sourced from Eurostat, JRC-IDEES and Open Power System Data databases (Pickering et al., 2022). The supply data such as technology costs are sourced from the Danish Energy agency technology catalog (Pickering et al., 2022). The multi-carrier and fully sector-coupled Euro-Calliope model forms the ideal basis for this master thesis project. However, it is important to note that the focus of this master thesis project is not the actual building and development of the Euro-Calliope model but using the Euro-Calliope model that is already presented to the public. Therefore, the in-depth details and methodology on the sourcing of data are therefore left out in this report, but they can be found in the paper published by Pickering et al. (2022) and the Euro-Calliope GitHub (GitHub, 2022).

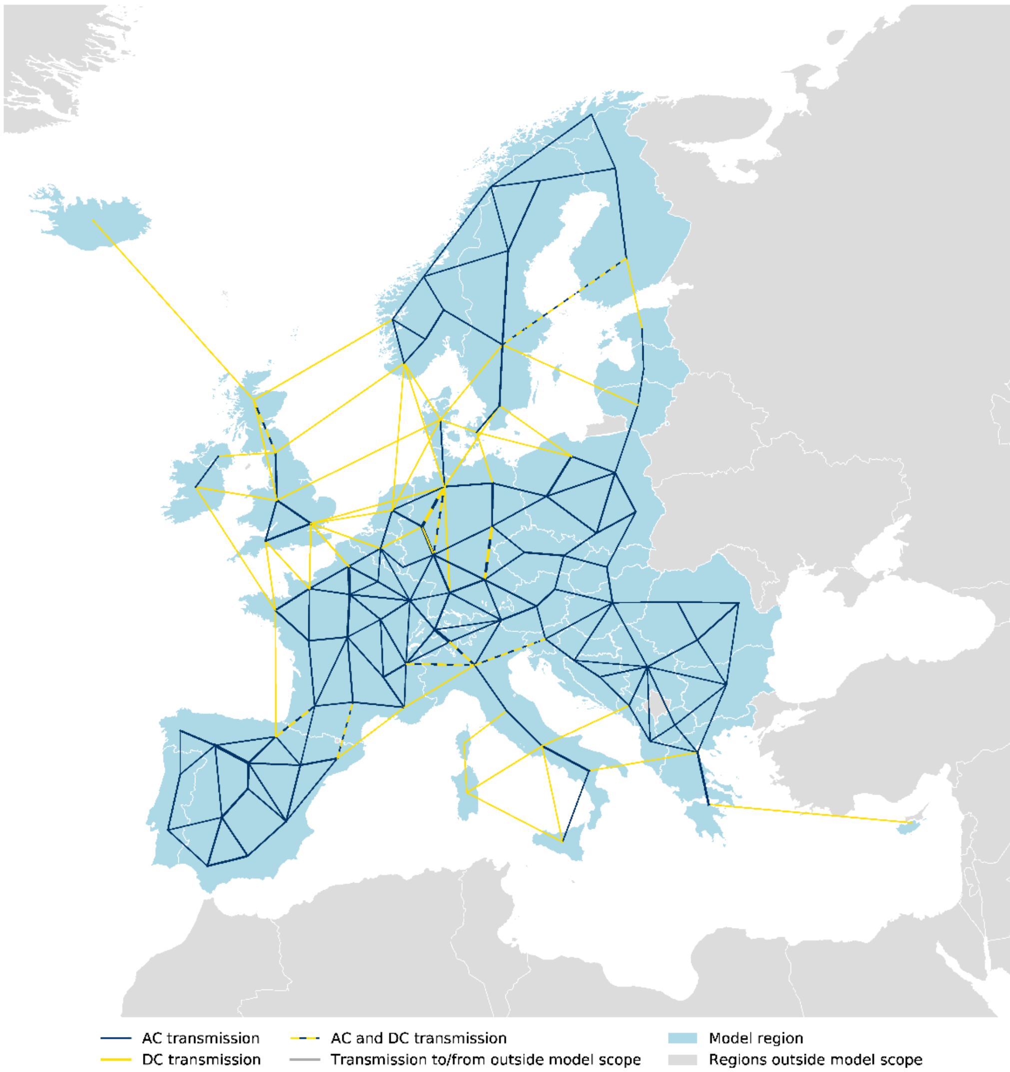


Figure 10: Overview transmission and node network from the Euro-Calliope model, figure from GitHub – Calliope project (2022)

4.1.2 North-Sea Calliope

One can imagine that running such a large-scale model is computationally burdensome. The European scale Calliope model has therefore been downscaled to the North Sea region covering all the major stakeholders within the North Sea projects, such as the North Sea Wind Power Hub Programme (NSWPHP, 2022) and the North Sea Energy organisation (North Sea Energy, 2022).

This North Sea Calliope model subsets the problem to what is most relevant for the Dutch energy transition. The North Sea Calliope consist of 10 nodes, one node for each country, covering all major stakeholders in North Sea projects. The included countries are Belgium, Germany, Denmark, France, Great Britain, Ireland, Luxembourg, the Netherlands, Norway and Sweden. Although Great Britain is not involved in some of the North Sea projects, it is still considered an important stakeholder due to its important presence in the region. An overview of the nodes and links of the North Sea Calliope can be found in Figure 11.

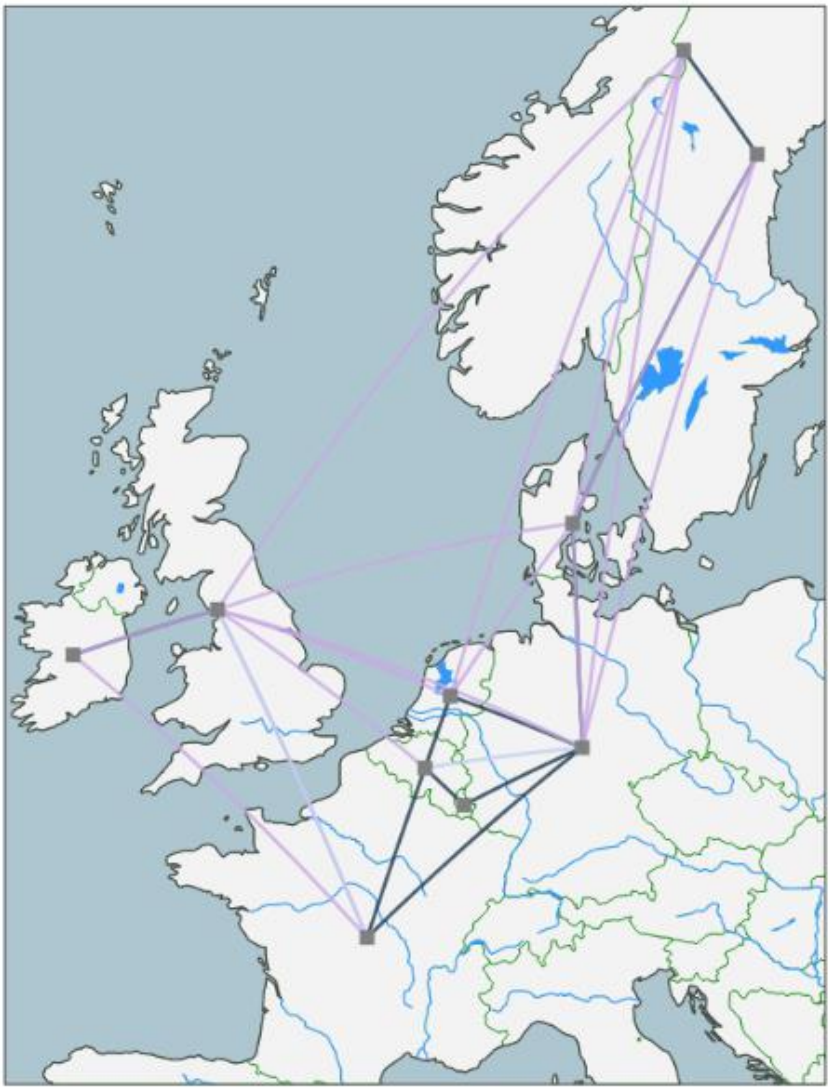


Figure 11: Overview of the North Sea Calliope Model, figure from Lombardi (2022)

The implications of the reduction in nodes for most countries mean that most energy carrier flows are reduced to a single flow to and from each country. Since the Netherlands was already represented by 1 single node in the larger European scale Calliope model, it is not expected that this would affect the carrier flow analysis. The internal transmission capacity of the Netherlands is among one of the best in the world it can therefore be assumed that there are no internal bottlenecks (Statista, 2022). However, the transmission operator (TSO) in the Netherlands has indicated that the net capacity is reaching its limit in multiple provinces (TenneT, 2021). Within the North Sea Calliope model, inter-country bottlenecks can still be analysed through the 1 node representation of countries. Similar to the Euro-Calliope, the trading of energy outside the North Sea region is disabled. The demand is fully met by the supply from within the North Sea region. Ultimately, this North Sea configuration is designed so that it can be run on a laptop with 16GB of RAM at a 24-hour resolution within reasonable computing time. Note that a 24-hour resolution might not be accurate enough for the validation process, but it is good enough to gain insight on the feasibility of the model (e.g., whether the output of the objective function is a non-negative value). For the validation process, a 1-hour resolution run is conducted which will be further described in Section 4.2.

Building the North Sea Calliope model

The steps along with the associated override files needed to downscale the Euro-Calliope model to the North Sea Calliope model are presented in Table 4. The {} represent the specific year for which the North Sea Calliope model is build. The most important step is to remove all non-North Sea countries from the Euro-Calliope model while maintaining the operational feasibility of the model.

Table 4: Overview of the steps to convert Euro-Calliope to North Sea Calliope

Step	Override file
Removing all non-North Sea locations from the Euro-Calliope model	north_sea_overrides.yaml
Removing all non-North Sea locations from bio-fuel supply	biofuel-supply-{}.yaml
Removing all non-North Sea locations from fuel group constraints	fuel_group_constraints_{}.yaml
Removing all non-North Sea locations from heat group constraints	heat_group_constraints_{}.yaml
Removing all non-North Sea locations from vehicle group constraints	vehicle_group_constraints_{}.yaml

North Sea Calliope 2020 model

The next step is to validate the newly build North Sea Calliope model. Within the North Sea Calliope model, technology data for the year 2020 is used using technology data from the Danish Energy Agency (Energistyrelsen, 2021). It is important to note that the weather conditions for this model are taken from the year 2015. This is due to the fact that the index year for the models are set for the year 2015. In order to make the North Sea Calliope model with the all the definitions and characteristic from the year 2020, additional override files are needed. An overview of 2020 specific override files are presented in Table 5. The key changes are the removal of non-widely available technologies such as hydrogen-based technologies, electric vehicles, the production of synthetic fuels and the use of large-scale energy storage systems. Furthermore, the demand and supply profiles are set to the year 2020.

Table 5: Overview of 2020 specific override files

Step	Override file
Remove all future technologies such as hydrogen, large-scale synthetic fuels and energy storage	kill-fancy-techs.yaml
Demand share of technologies specific to the year 2020	Fix-demand-share_min.yaml
Production capacities of technologies specific to the year 2020	fix-current-national-capacities.yaml
Technology specifications specific to the year 2020	heat-techs.yaml, renewable-techs.yaml, storage-techs.yaml, transformation-techs.yaml
Importing the 2020 specific yaml files	model-2015.yaml

The so-called scenario string to build the 2020 North Sea Calliope model is presented in Snippet 1. Thereafter, the model is built using Snippet 2. The YAML files can be found in the main Calliope folder which can be accessed through Appendix B2 - North Sea Calliope model. A brief description of all the override files can be found in Appendix D – Description of North Sea Calliope 2020 override files. The full North Sea Calliope model within the Calliope modelling frame is presented in Figure 12.

```
scenario_string['2020'] =
"industry_fuel,transport,heat,config_overrides,gas_storage,link_cap_1x,"\
                        "freeze-hydro-capacities,heat_techs_2020,"\
                        "renewable_techs_2020,transformation_techs_2020,"\
                        "fossil-fuel-supply,res_1h,"\
                        "add-biofuel,coal_supply,north_sea,"\
                        "kill-fancy-techs,fix-generation-capacities,"\
                        "demand_share_fuel_current_min"
```

Snippet 1: Define override scenarios

```
model_input = create_input.build_model(path_to_model_yaml,
scenario_string[selected_scenario], path_to_netcdf_of_model_inputs)
```

Snippet 2: Generating and saving the model inputs

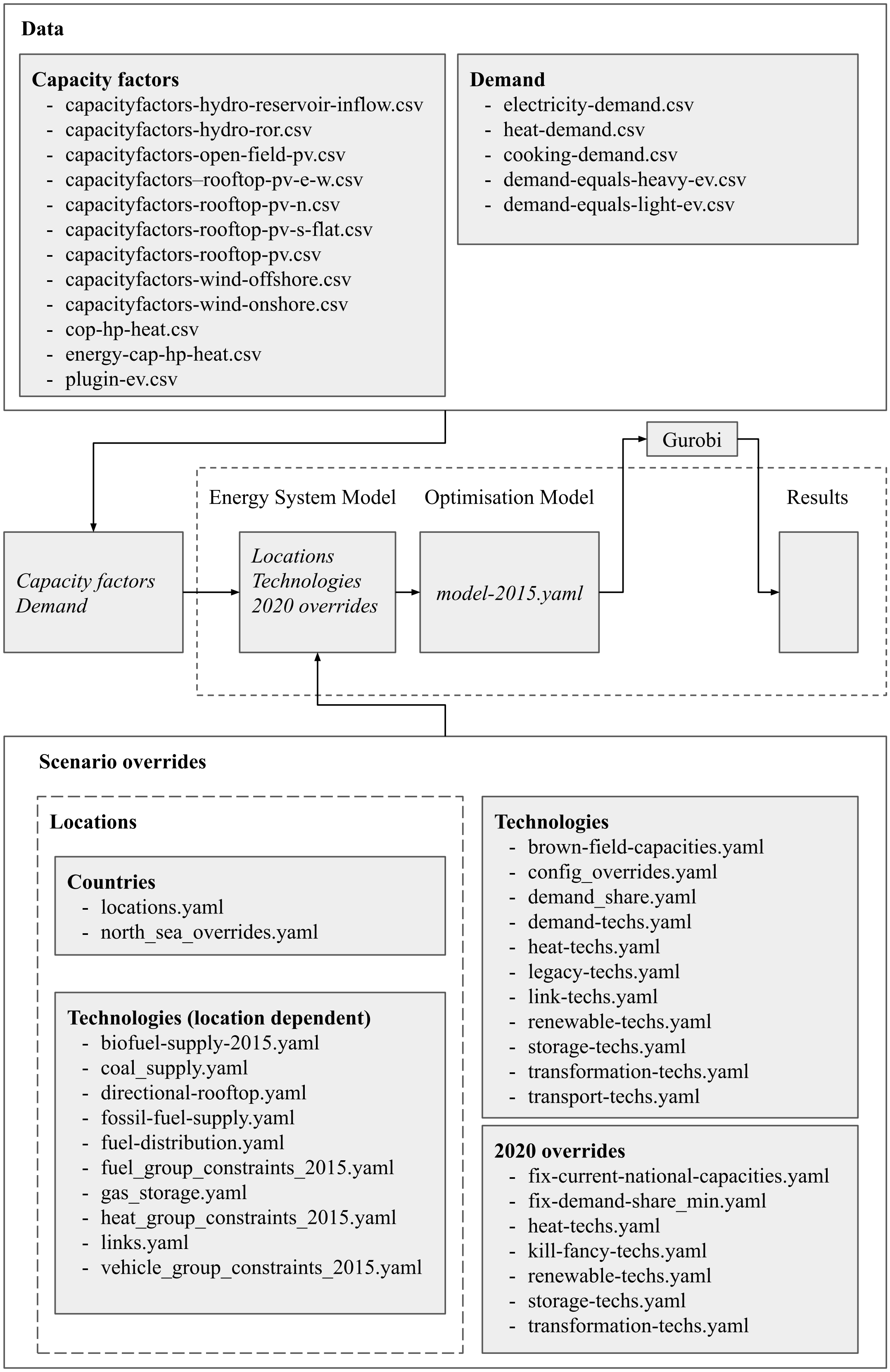


Figure 12: 2020 North Sea Calliope model within the Calliope modelling framework

Units within the North Sea Calliope model

An overview of the base units for the North Sea Calliope model is presented in Table 6. These units will be relevant when analysing the physical and duals values of the model.

Table 6: Overview of the monetary and production variables and its unit

Definition	Variable	Unit
Monetary cost for installed capacity	cost.monetary.energy_cap	10.000 EUR/MW
Monetary cost for O&M annually	cost.monetary.om_annual	10.000 EUR/MW
Monetary cost for O&M per production unit	cost.monetary.om_prod	10.000 EUR.MWh
Production capacity	constraints.energy_cap	100.000 MW

From the equation (7) and the units given in Table 6, it can be calculated that the unit for the objective function is in Billion EUR (10.000EUR/MW * 100.000 MW) The shadow prices, therefore equal to Billion EUR per extra unit of carrier. A unit of carrier in the model equals to 100.000 MWh as the production capacity is multiplied by the timestep in the post-process step. So, the base unit for shadow price is given in 10.000 EUR/MWh. The monetary value in the model has the base year in 2015.

4.2 Running the North-Sea Calliope model

In order to run the model, the Python code presented in Snippet 3 is used.

```
model_run, duals = run.run_model(path_to_netcdf_of_model_inputs,
path_to_netcdf_of_results)
```

Snippet 3: Running the model and extracting the duals

For the validation of electricity duals, the hourly day-ahead-price for the Netherlands in the year 2020 from ENTO-E is used. To run the model at an one-hour resolution, the DelftBlue Supercomputer (2022) was used. The job specifications are presented in Table 7. The specific job script can be found in Appendix E – Job script 2020 model 1h resolution.

Table 7: Overview of DelftBlue supercomputer setup

Definition	Variable	Value
Type of job	partition	Compute
Total run time for the job	time	24:00:00
Number of tasks	ntasks	1
Number of CPUs per task	cpus-per-task	16
Memory per CPU	mem-per-cpu	10G

Solver settings

Within the Calliope Modelling Framework, different solvers are available. For this research project, the Gurobi solver is used for its superior computing power compared to the default cbc solver. A downside for the Gurobi solver is that it is only commercially or academically available. While setting up the solver options, a good balance needed to be found between stability and speed. A high stability run would require more run time, and due to the hard limitation of 24 hours per run, this is not always possible. However, if the stability is not high enough, a numerical error would occur and the model would render itself infeasible. The solver settings used for the 2020 North Sea Calliope model is presented in Table 8. The full documentation of the Gurobi solver options can be found in the documentation on their website (Gurobi, 2019).

Table 8: Overview of the Gurobi solver options used

Definition	**Variable**	**Value**
Thread count	Threads	6
Algorithm used to solve continuous models	Method	-1
Crossover basis construction strategy	Crossover	-1
Primal feasibility tolerance	FeasibilityTol:	1e-3
Dual feasibility tolerance	OptimatilyTol:	1e-4
Barrier convergence tolerance	BarConvTol:	1e-4
Barrier homogeneous algorithm	BarHomogeneous	1

4.3 Validation of the 2020 North Sea Calliope model

The validation process requires two steps. In the first step, the physical model is analysed to check on the feasibility of the model. The second step is to check the monetary model through the use of day-ahead prices and dual values for electricity.

4.3.1 Physical model

In order to validate the physical model, the total production from supply technologies from the Netherlands from the North Sea Calliope model will be evaluated against the total energy supply data from the IEA. The total energy supply data by source is presented in Figure 13. The notation for the carriers are a little bit different for the IEA data and the model data. Within the Calliope model, the VRES are grouped under the carrier name *electricity*. *Methane* in the model is equivalent to *Natural gas* in the IEA data. After synchronization, the supply shares are then compared to each other and presented in Figure 14. It is important to note that the IEA data and model data are not 1-on-1 identical to each other due to the simplifications made in the Calliope model. Furthermore, the model data doesn’t include import and export data from outside the region, whereas the IEA data does include import and export data. The difference for between IEA data and model data could therefore be explained by the fact the Netherlands is an important trade hub for natural gas. The difference in the total energy supply for biofuel could be explained by the deliberate choice of removing synthetic fuel production in the model and being replaced by the traditional fossil fuels, coal, oil and natural gas. Nevertheless, the similarity in the share of supply is acceptable.

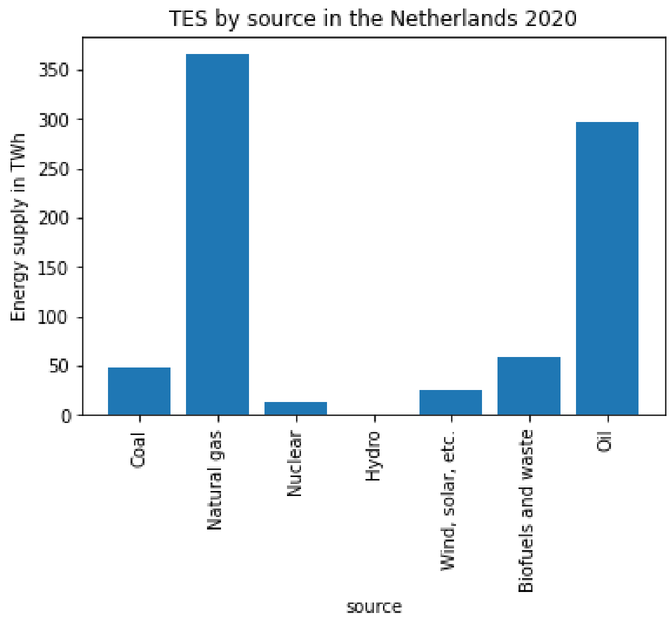


Figure 13: TES by source in the Netherlands for the year 2020. Data from IEA (2022)

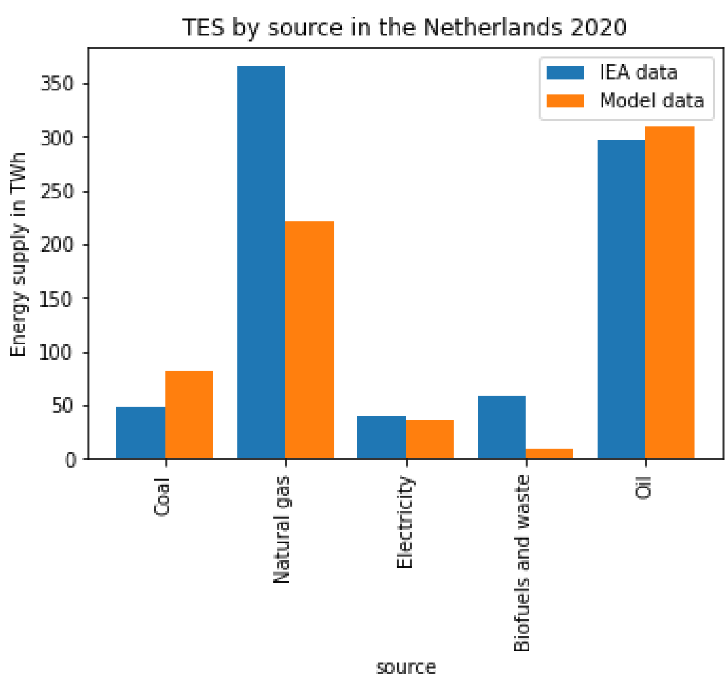


Figure 14: Energy supply by source, comparison between IEA data and model data

4.3.2 Dual values for electricity

As mentioned in section 4.1, the hourly day-ahead-price for the Netherlands in the year 2020 from ENTSO-E will be used to compare the dual values for electricity from the year 2015, because the weather data from the 2020 model is also from the year 2015.

The electricity duals for the 2020 North Sea Calliope model has been plotted for the month March in Figure 15. The month March has been chosen arbitrarily. The electricity duals from the model are less dynamic than the real-world prices. It can be observed that the shadow price for electricity is more or less flat in the given timeframe. This implies that the demand for electricity is always sufficiently met and the price can therefore stay at the price of the variable costs of electricity production.

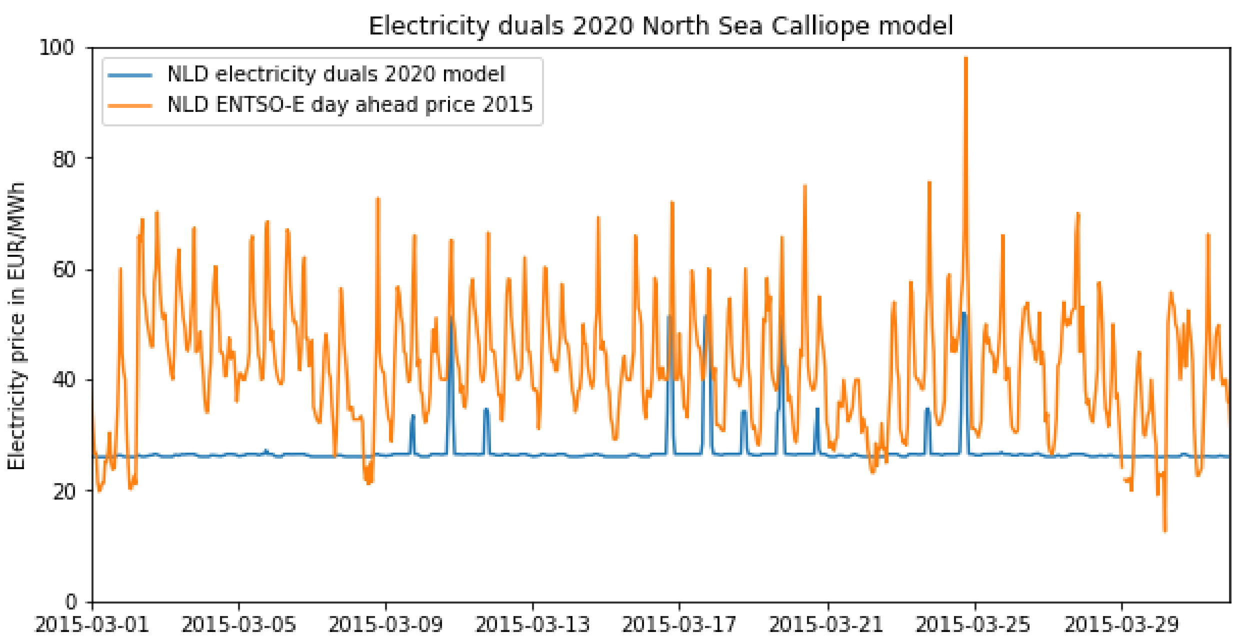


Figure 15: Electricity duals plotted together with ENTSO-E day-ahead prices

To test this hypothesis, a model that includes only the power sector is created. The electricity duals are again plotted in Figure 16. It can be observed that the price dynamics are still relative flat compared to the real-world prices. It does show however, that when the dependency on electricity increases, the prices are also more dynamic.

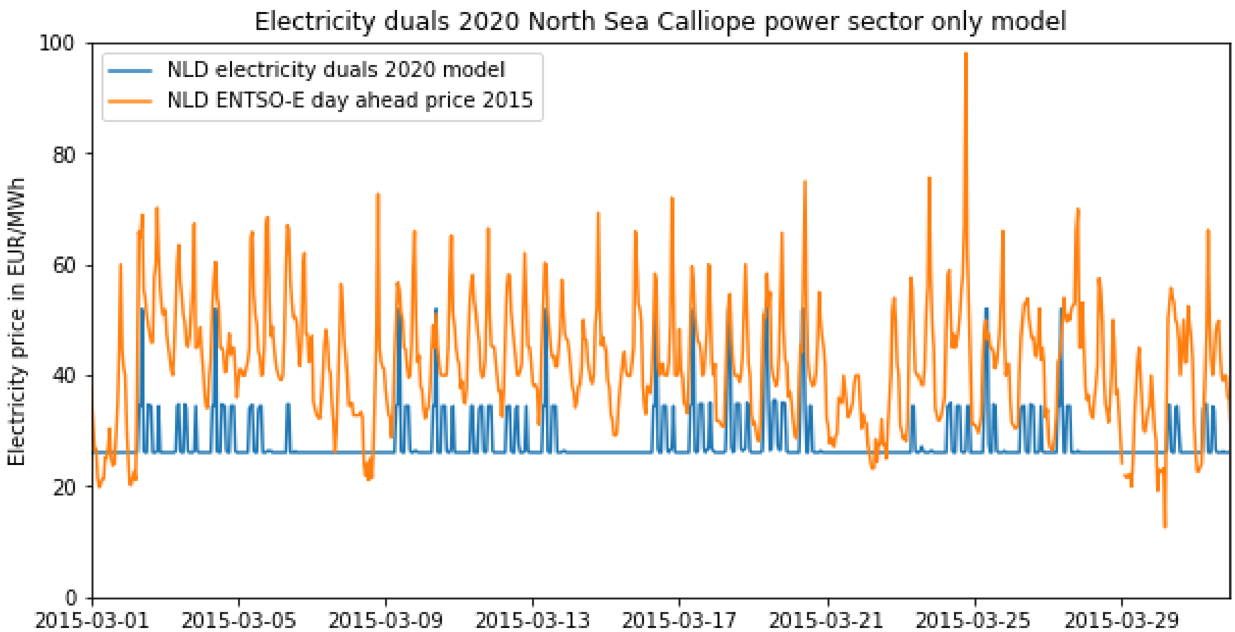


Figure 16: Electricity duals plotted together with ENTSO-E day-ahead prices with only power sector Calliope model

It can be seen from Figure 15 and Figure 16 that shadow prices do not represent real-word prices perfectly. This is in some way expected as the model is not a perfect representation of the real world. The model is merely a simplified version without many variables such as ramping costs and shutdown costs of the production units. Furthermore, there are no transmission bottlenecks and grid congestions within the model. Moreover, as the demand is always met by the supply, the scarcity of electricity is never an issue and as a result, there is no reason for electricity prices to increase.

Although the shadow prices do not reflect the real-world prices, the shadow prices can be used to analyse trade-offs within a fully sector-coupled energy system such as the influence of different hydrogen shares and its ability to absorb shocks in different scenario configurations. This unique use of price information is tested in chapter 5, where shadow prices are used to further explore the trade-offs in different hydrogen configurations.

Limitations for LP problems in fully sector-coupled models

The largest limitation in the validation of individual energy carrier prices is due to the nature of LP problems. A LP problems does not optimise each decision variable independently, rather it searches for an optimal objective value for the whole problem. Consequently, fully sector-coupled energy models do not focus on optimizing each energy carrier independently, but it finds a sector-wide equilibrium. This implies that while some energy carrier prices might make sense, others may not.

Chapter 5 – Assessing trade-offs within different hydrogen configurations

This chapter describes how the price information can be used to understand trade-offs between different hydrogen configurations of a fully sector-coupled energy system model. Section 5.1 describes the experimental setup. Section 5.2 describes the results. The Python scripts used for the analysis for the 2025 model can be found in Appendix S2.

5.1 Experimental setup

In order to analyse trade-offs between different hydrogen configurations, first a set of hydrogen configurations is created. This is done in three consecutive steps. First, a base Calliope model is chosen from which the runs are performed. Second, by using the available weather data sets, a optimal hydrogen share configuration is found for every weather year. From these optimal hydrogen share configurations, it is possible to identify the years in which weather was bad, normal or good. Finally, these three optimal hydrogen share weather models form the basis for the different hydrogen configurations. For each weather scenario, the share of hydrogen is then increased incrementally to assess the effect of different shares of hydrogen within the energy system for each weather year. This will ultimately give a total of 9 different hydrogen configurations for the fully sector-coupled energy system from which the trade-offs can be analysed. The analysis includes technical parameters such as deployed capacity and economical parameters such as LCOE, cost recovery and price deviation across the different configurations. An overview of these 9 different hydrogen configurations are shown in Table 9.

Table 9: Overview of the different hydrogen configurations

	Bad weather	**Normal weather**	**Good weather**
Optimal hydrogen share	**Configuration 1:** Optimal hydrogen share in bad weather	**Configuration 2:** Optimal hydrogen share in normal weather	**Configuration 3:** Optimal hydrogen share in good weather
Slightly increased hydrogen share	**Configuration 4:** Slightly increased hydrogen share in bad weather	**Configuration 5:** Slightly increased hydrogen share in normal weather	**Configuration 6:** Slightly increased hydrogen share in good weather
Increased hydrogen share	**Configuration 7:** Increased hydrogen share in bad weather	**Configuration 8:** Increased hydrogen share in normal weather	**Configuration 9:** Increased hydrogen share in good weather

5.1.1 Base model

The base model for assessing trade-offs within different hydrogen scenarios will be the North Sea Calliope model for the year 2050. The 2050 model, unlike the 2020 model has the ability to produce synthetic hydrocarbons such as methane, methanol, diesel and kerosene. Furthermore, in the 2050 model, the model is totally free from the use of traditional fossil fuels such as coal, natural gas and oil. Instead, the use of hydrogen, electric vehicles and large-scale energy storage system are widely adopted. This makes the 2050 model the ideal base model for the experimental setup. The building of the 2050 North Sea Calliope model is similar to the 2020 North Sea Calliope model. The main difference is that the 2020 override files that were used to set technology data to 2020 values is not implemented for the 2050 model. Furthermore, for the demand data, annual demand and annual industry energy demand data is used for the year 2050. These demand profiles can be found in Appendix C, under the 2050 folder. The 2050 North Sea Calliope model within the Calliope modelling framework is presented in Figure 17. The key different YAML files compared to the 2020 North Sea Calliope model are presented in bold.

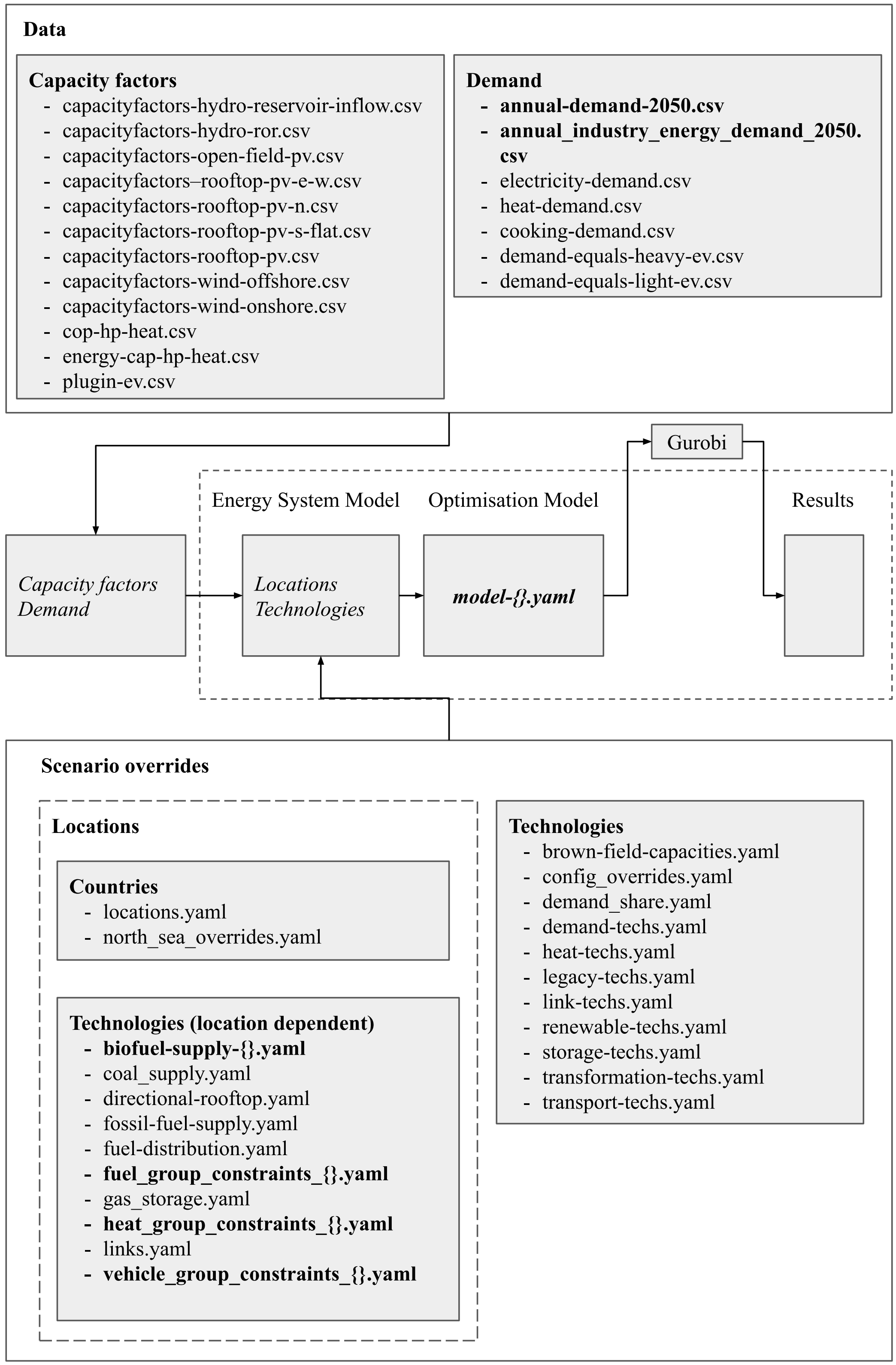


Figure 17: 2050 North Sea Calliope model within the Calliope modelling framework

5.1.2 Weather scenarios

Energy output from VRES are dependent on weather (Pfenninger et al, 2014). Within the Calliope model, weather has therefore an effect on the capacity factors of hydro, PV (open-field and rooftop) and wind (offshore and onshore) power. Furthermore, it also has an effect on the demand for electricity, cooking, electric vehicles (heavy transport, light transport and plugin EV) and heat. Note that some of the YAML files in Figure 17 are in bold and contain *{}*. The *{}* represent the respective weather year for the 2050 model. For this master thesis, weather data from 2010 to 2018 is available. The capacity factor time series for solar and wind power are from Tröndle et al. (2022). The respective YAML files and data files can be found in Appendix C under the folder 2050/model/national. The methodology to acquire demand data is described in Pickering et al. (2022). In order to assess the stability of price dynamics in different hydrogen scenarios, three different weather types will be selected from the available data set. The weather types are classified as 'good weather', 'bad weather' and 'normal weather'. One cost-optimal solution will be found for each of the weather years for the 2050 model. The version with the lowest objective value which equates to the lowest cost value is then considered a 'good weather' year. Naturally, the version with the highest objective value is considered as a 'bad weather' year. For the 'normal weather' year, the version closest to the median objective value is chosen. An overview for the objective value with the associated weather year is shown in Figure 18. It can be observed in Figure 18 that the weather year 2010 has the highest objective value from all the different weather years, therefore the year 2010 is classified as a 'bad weather' year. The weather year 2015 has the lowest objective value from all the different weather years, therefore the year 2015, is classified as a 'good weather' year. The weather year associated with the median objective value is the weather year 2016, therefore, the weather year 2016 is classified as a 'normal weather' year.

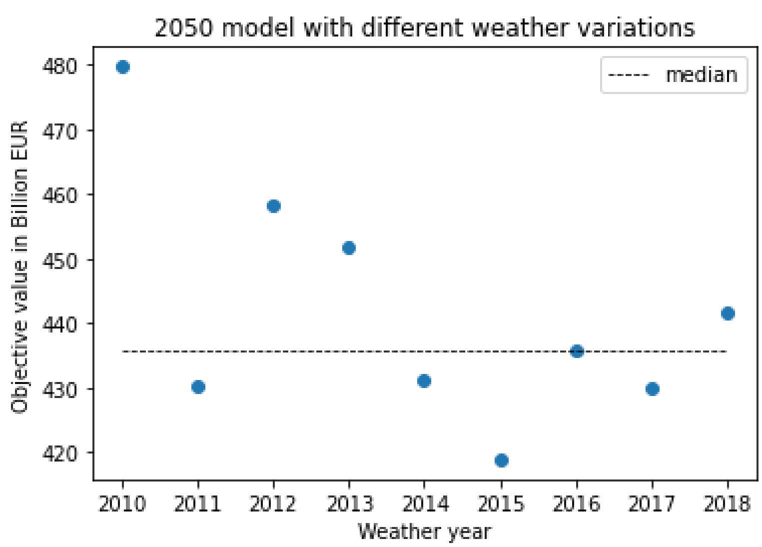


Figure 18: 2050 North Sea Calliope model with different weather variations

5.1.3 Hydrogen configurations

The goal is to create different configurations where the share of hydrogen is different in the total energy supply. To do this two separate steps are performed. The first step is related to the relative share of hydrogen compared to biofuels in the production of synthetic fuels. Within the model, hydrogen is mainly used for the production of the synthetic hydrocarbon fuels diesel, methane, methanol and kerosene. These synthetic hydrocarbons are then used in other demand sectors such as heat, transport and storage. Biofuel is also used for the production of the same synthetic hydrocarbons. This means that biofuel is a direct competitor for the production of synthetic hydrocarbon fuels. To vary the relative share of hydrogen in different configurations, it is therefore necessary to set the share of hydrogen in the production of synthetic fuels relative to biofuels.

The second step is related to the relative share of synthetic fuels in the total energy supply. To make sure the that the share of hydrogen is actually increased or decreased when varying the share of hydrogen in the production of synthetic fuels, the fuel share demand for the synthetic fuels needs to be set. An overview of this methodology is presented in Figure 19.

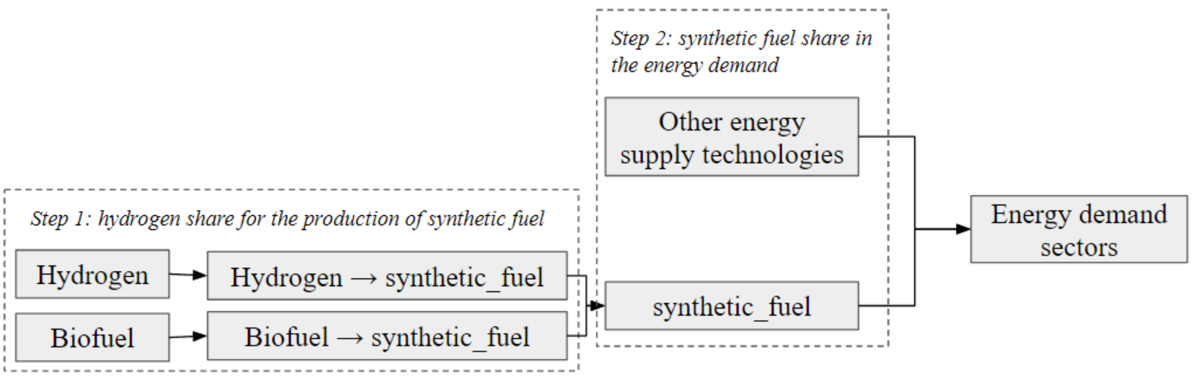


Figure 19: Methodology for varying the share of hydrogen in the model

Hydrogen share for synthetic fuel production

Figure 20 presents the hydrogen share for the production of synthetic fuel for the optimal hydrogen share configurations for the three defined weather scenarios. It can be observed that the share of hydrogen for the production of synthetic kerosene and synthetic methane is always 100% regardless of the weather type. This is due to the high cost for biofuel to convert to kerosene and methane when compared with hydrogen. The average share of hydrogen for the production of synthetic fuels are 90.7%, 87.8% and 92.4% for a bad, good and normal weather scenarios respectively. Since the average share of hydrogen in the production of synthetic fuel is already quite high, setting the inequality constraint to a minimum of less than 80% would not result into any significant changes compared to the optimal hydrogen share configuration. Moreover, in order to reduce computing time and ensure feasibility of the model, an inequality constraint is used instead of an equality constraint. The share of hydrogen in the production of synthetic fuel is therefore set to a minimum of 80%. Furthermore, since synthetic fuels can only be produced by either hydrogen or biofuels, it can be observed that hydrogen is predominantly used compared to biofuels for the production of synthetic fuels. It is important to note that the supply of synthetic fuels is not constant for all weather years as can be seen in Figure 22, therefore a different share of hydrogen for the synthetic fuel production might not necessarily mean that the absolute value for the use of hydrogen is different. The hydrogen share for the production of synthetic fuels relative to biofuels is set using an override file which is is presented in Appendix E.

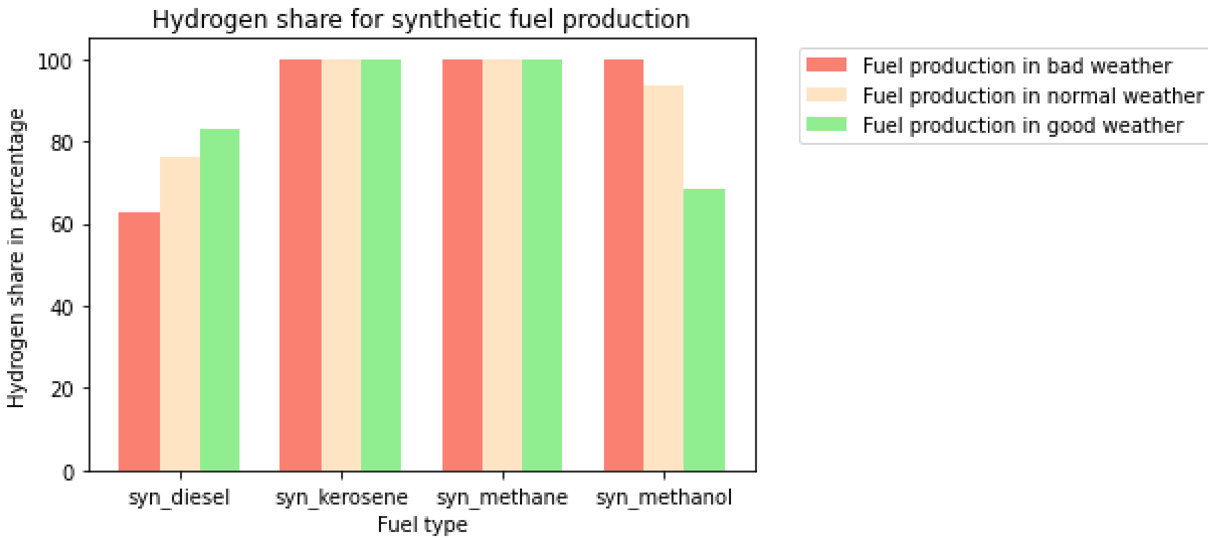


Figure 20: Hydrogen share for the production of synthetic fuels

Synthetic fuel share in the total energy supply

Figure 21 presents the synthetic fuel share in the total energy supply. It can be observed that the synthetic fuel share within the energy system ranges between roughly 2 and 10% for synthetic methane and synthetic methanol respectively. The combined share of all synthetic fuels will be set to a pre-determined value as the total share of synthetic fuels relative to the total energy demand. These pre-determined values are arbitrarily chosen as no prior information is available but should large enough to show significant changes relative to the optimal hydrogen share configurations. The values chosen are therefore an increase of 30% in the fuel share demand and an increase of 50% in the fuel share demand. The fuel demand in all 2050 models is supplied by synthetic fuels, therefore the increase in the fuel share demand is equivalent to the increase in the use of hydrogen. The increase of fuel demand for synthetic fuels in the total energy supply is set using the demand share override file which can be found in Appendix G. Appendix G.1 presents the override file to increase the fuel demand share by 30%. Appendix G.2 presents the override file to increase the fuel demand share by 50%.

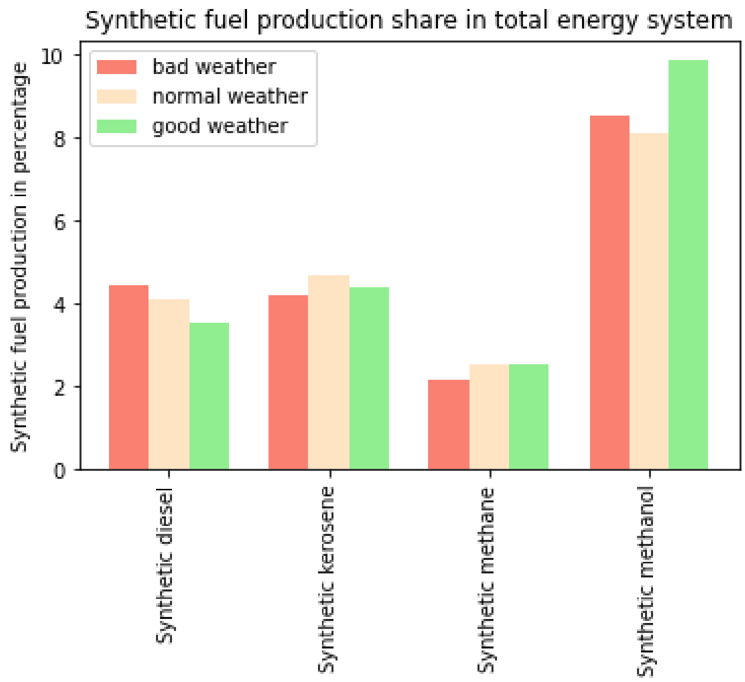


Figure 21: Synthetic fuel production share in total energy system

5.1.4 Overview of the configuration sets

The different hydrogen configurations are now defined and are presented in Table 10. The three scenarios chosen are therefore an optimal hydrogen share scenario, a scenario with a minimum of 80% hydrogen share for the synthetic fuel production combined with a 30% fuel demand increase and a scenario with a minimum of 80% hydrogen share for synthetic fuel production combined with a 50% fuel demand increase. This would allow for the analysis of price dynamics relative to the increase of the hydrogen share in the total energy system.

Table 10: Hydrogen configurations for the 2050 model

	Bad weather		**Good weather**		**Normal weather**	
	H_2 share for synthetic fuel production [%]	Fuel demand increase [%]	H_2 share for synthetic fuel production [%]	Fuel demand increase [%]	H_2 share for synthetic fuel production [%]	Fuel demand increase [%]
Optimal hydrogen share	90.7	0	87.8	0	92.4	0
Slightly increased hydrogen share	80	30	80	30	80	30
Increased hydrogen share	80	50	80	50	80	50

5.2 Analysis of the results

In this section, the results for different configurations of a fully sector-coupled energy system is described. It starts with an analysis of technical parameters such as total energy supply and capacity deployment within the energy system in section 5.2.1 through 5.2.3. Then From section 5.2.4 through 5.27 economic parameters such as total energy system cost, levelized cost of energy, price stability and payback time are analysed.

5.2.1 Total energy supply

The total energy supply (TES) by source in the Netherlands for the 2050 model for different weather variations is presented in Figure 22. Some observations can be made when looking at the total energy supply for different weather types. In the optimal hydrogen share configuration, electricity supply is linearly dependent on the weather type. During good weather, the electricity supply is the highest, due to the high availability of VRES such as solar and wind and during bad weather the electricity supply decreases. Total energy supply in the Netherlands is 733, 762 and 717 TWh for a bad, good, and normal weather scenario respectively.

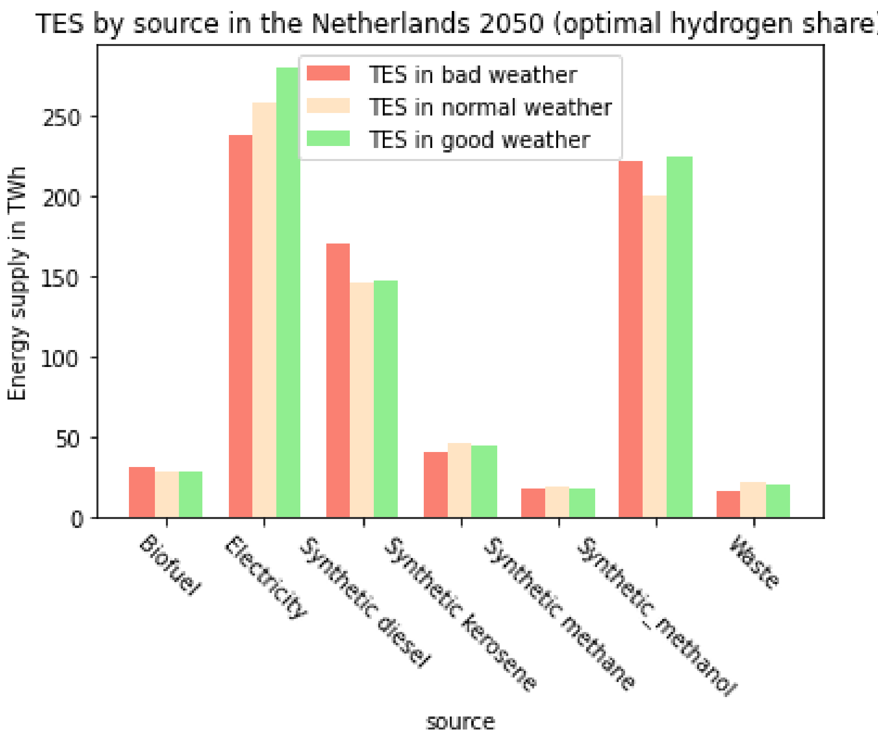


Figure 22: Total energy supply by source in the Netherlands in 2050

The model data on the TES in the Netherlands for the year 2050 is compared against the Infrastructure Outlook 2050 from TenneT (TenneT, 2019). The comparison on the supply share within the energy system is presented in Figure 23. It can be observed that the supply share of electricity in both data sets is the highest, followed by hydrogen. This indicates that both in the North Sea Calliope 2050 model and Infrastructure Outlook 2050 report expect that these two sources of supply will be dominant in the future energy infrastructure in the Netherlands. However, when looking at the other supply sources, the differences are much larger. Liquid fuels in the North Sea Calliope 2050 model for example presents a much higher percentage in the supply share compared to the Infrastructure Outlook 2050 data. Moreover, methane and other sources of energy supply which include biofuels and waste are lower when compared to the Infrastructure Outlook 2050 data. Aside from the supply share within the energy system, the absolute value of energy supply is also compared. The Infrastructure Outlook 2050 reports a TES of 417 TWh whereas the model data has a TES of an average of 737 TWh across the three weather scenarios. The model data thus has an increase of almost 77% relative to the Infrastructure Outlook 2050 data.

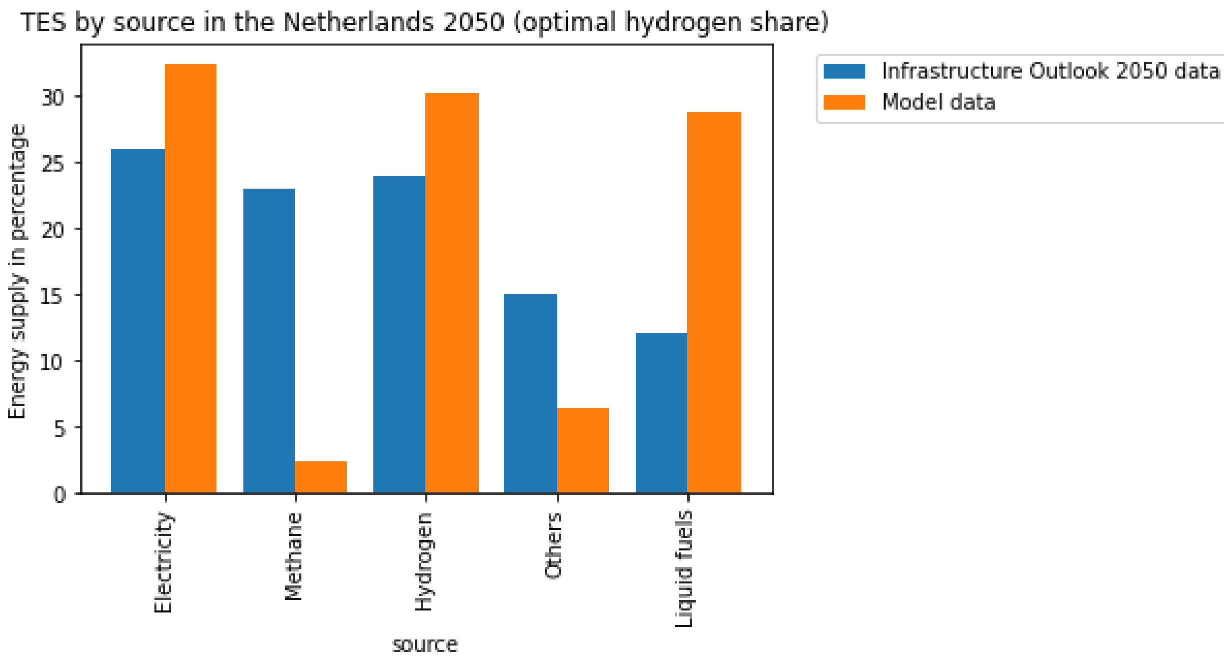


Figure 23: Comparing model data with Infrastructure Outlook 2050 Data on the TES by source

When increasing the fuel share within the energy system, the energy supply share seems to be predominantly electricity as can be seen in Figure 24 where the TES by source in the Netherlands for 2050 is presented for increased fuel demand scenarios. The increased fuel demand seems to be fulfilled almost entirely by electricity. However, as a result, the energy supply shares within the energy system do not seem realistic nor in line with other 2050 scenario reports such as the Infrastructure Outlook 2050.

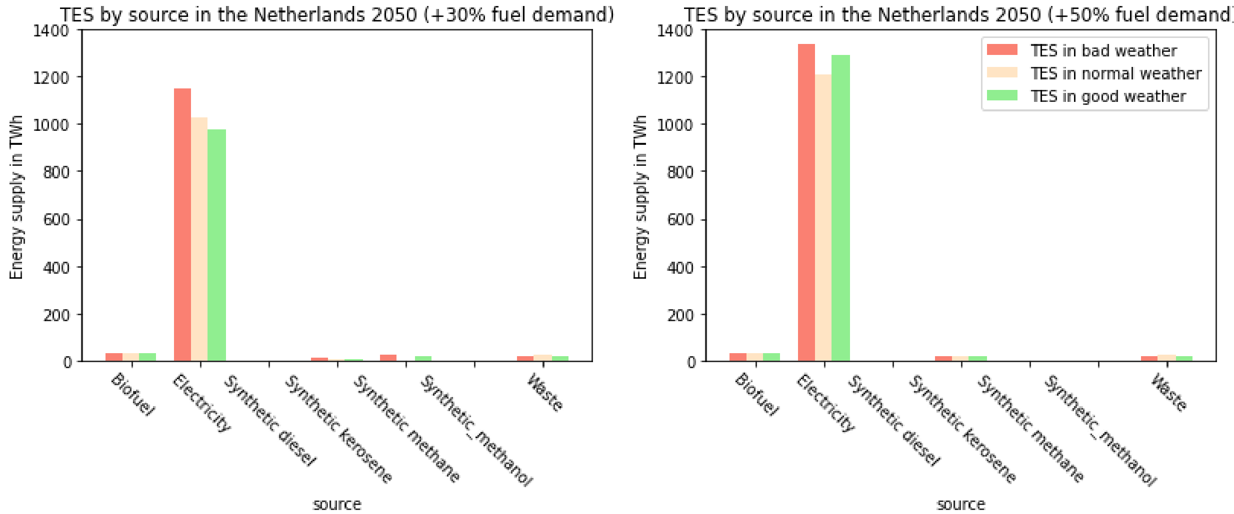


Figure 24: TES by source in the Netherlands 2050 with increased fuel demand

The increase in of energy supply through electricity can be explained by the increase in electricity demand from electrolysis as can be seen in Figure 25. This is a direct result from the increase of hydrogen within the energy system. It can also be observed that the conversion of hydrogen to liquid fuels has also increased significantly compared to the optimal hydrogen share configuration to a total demand of 1108 TWh which is more than twice as high than the highest estimates from TNO (Detz et al, 2019). Furthermore, it can also be observed that the electricity demand for batteries has reduced. This could indicate that the conversion to hydrogen using excess electricity is more preferable compared to storing it in batteries.

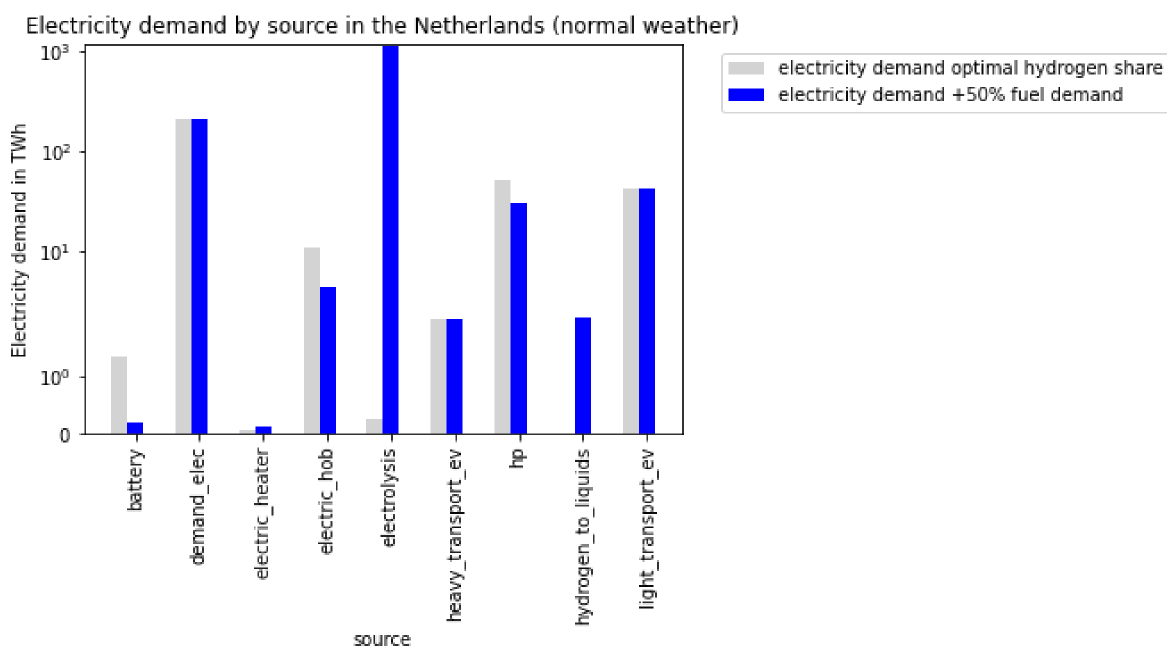


Figure 25: Comparison electricity demand by source optimal scenario vs +50% fuel demand

5.2.2 Capacity deployment

Figure 26 presents the total electricity production by source from the whole North Sea region across the different weather scenarios. It can be observed that onshore wind is the main source of electricity production for the whole North Sea region followed by open field PV. This indicates that the energy production within the energy system is predominantly from VRES. Furthermore, it seems that onshore wind production is higher for both bad and good weather scenarios compared to a normal weather scenario. Open field PV is having the lowest and highest electricity production for bad weather and good weather respectively. Electricity production from wind will be analysed further in detail in the following subsection since it is the predominant source of electricity.

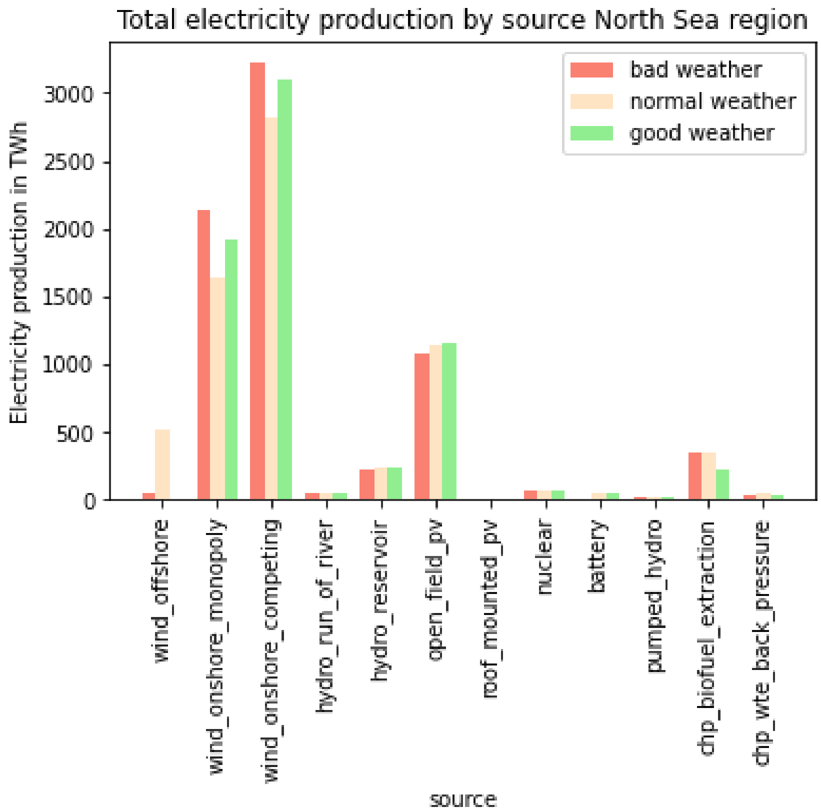


Figure 26: Total electricity production by source North Sea region

Wind: North Sea region

The installed capacity for wind electricity production is calculated using the following formula:

$$C_{installed} = \frac{E_{generated}}{CF * h} \qquad (9)$$

Where $C_{installed}$ is the installed nominal capacity in megawatt, $E_{generated}$ is the total energy generated in megawatt hour, CF is the average capacity factor and h the time period in hours. The capacity factors from each country and average capacity factor across the entire North Sea region for both offshore and onshore wind energy can be found in Appendix H – Average capacity factors wind offshore and onshore.

Applying equation (9) for the normal weather scenario then gives an installed capacity of 130 GW and 1125 GW for offshore and onshore production respectively. Although there is a consensus that onshore electricity production will be greater than offshore electricity production in 2050 (WindEurope, 2022; TenneT, 2019), the absolute values for installed capacity are different. WindEurope (2019) reports that 750GW of onshore wind production is needed for the entire European Union in 2050. Furthermore, WindEurope (2020) reports a total 380 GW of offshore wind production in the Northern seas and for the entire European Union, a total of 450 GW of offshore wind energy is installed by 2050. The actual difference between the installed capacities is even more as the WindEurope report includes offshore wind capacity from Poland, Finland, Lithuania, Latvia and Estonia as part of the Northern seas. The preference for onshore wind farms over offshore wind farms in the model can be attributed to the lower cost of onshore wind farms compared to offshore wind farms. The O&M for the production cost for onshore windfarms within the model is €1.22/MWh compared to the €2.40/MWh for offshore windfarms. Therefore, whenever onshore wind production is available, this will always be dispatched first whereas in real world scenarios the dispatch order might be influenced by factors other than cost alone. Figure 27 present the total electricity production by source for the North Sea region for different fuel share demand increases. The nominal installed capacities in the normal weather scenario for onshore and offshore wind production for a 30% increase in fuel share demand are 2101 GW and 236 GW respectively. For a 50% increase in fuel share demand the installed capacities for onshore and offshore wind production are 2420 GW and 362 GW respectively. These nominal installed capacities exceed the currently planned infrastructure outlooks for 2050. However, they are within the total potential for onshore and offshore wind energy of 52TW and 8.6 TW across Europe (Caglayan et al., 2019; Enevoldsen et al., 2019).

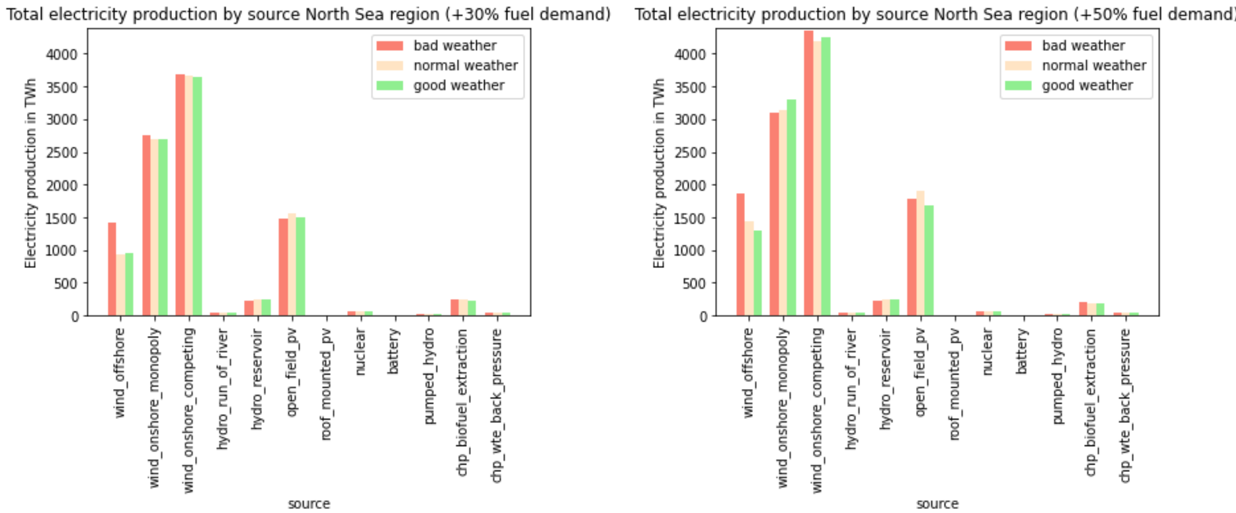


Figure 27: Total electricity by source for the North Sea region for different fuel share demands

Wind: The Netherlands

The total electricity production by source in the Netherlands is also predominantly from wind energy followed by open field PV as can be seen in Figure 28. This is similar to the entire North Sea region.

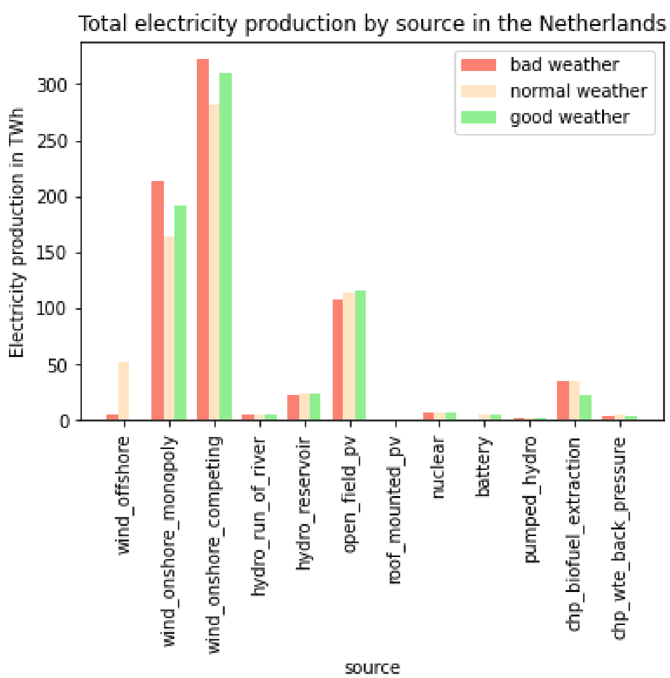


Figure 28: Total electricity production by source in The Netherlands (optimal hydrogen share)

By using equation (9) and the capacity factors for the Netherlands, the nominal installed capacities for the Netherlands are also calculated. For a normal weather scenario and optimal hydrogen share the nominal installed capacities for onshore and offshore wind energy production are 140 GW and

11.1 GW respectively. Both these nominal installed capacities are higher than the reported capacities from the Infrastructure Outlook 2050 from TenneT (2019) which state capacities of 14 GW and 53 GW for onshore and offshore wind respectively. Moreover, the Infrastructure Outlook 2050 report a higher generation for offshore wind compared to onshore wind in the Netherlands.

5.2.3 Total cost energy system

In this sub section the total cost of the energy system is analysed. An overview of the change in total system cost is presented in Figure 29 for the different hydrogen configurations. The total system cost is equal to the objective value of the optimisation problem as the objective is to minimise to total system cost. It is expected that the system is higher for scenarios with a higher fuel demand as higher fuel demand requires more energy supply and therefore increases the total costs. Similar to the optimal hydrogen share configurations, the total cost for bad weather scenarios remain the highest and the total cost for good weather scenarios remain the lowest. It is interesting to note that the relative increase in total energy system cost is the lowest for bad weather scenarios, from 479 billion euros to 742 billion euros and the highest for normal weather scenarios, from 436 billion euros to 694 billion euros. This equates to a relative change of 54.6% and 59.4% for a bad and normal weather respectively. This implies that the relative cost to improve price stability through an increase of hydrogen share within the energy system is lower for bad weather scenarios compared to good and normal weather scenarios. The relative cost increase for good weather scenarios compared to bad weather scenarios, however, is quite minimal with a relative increase of just 0.3%.

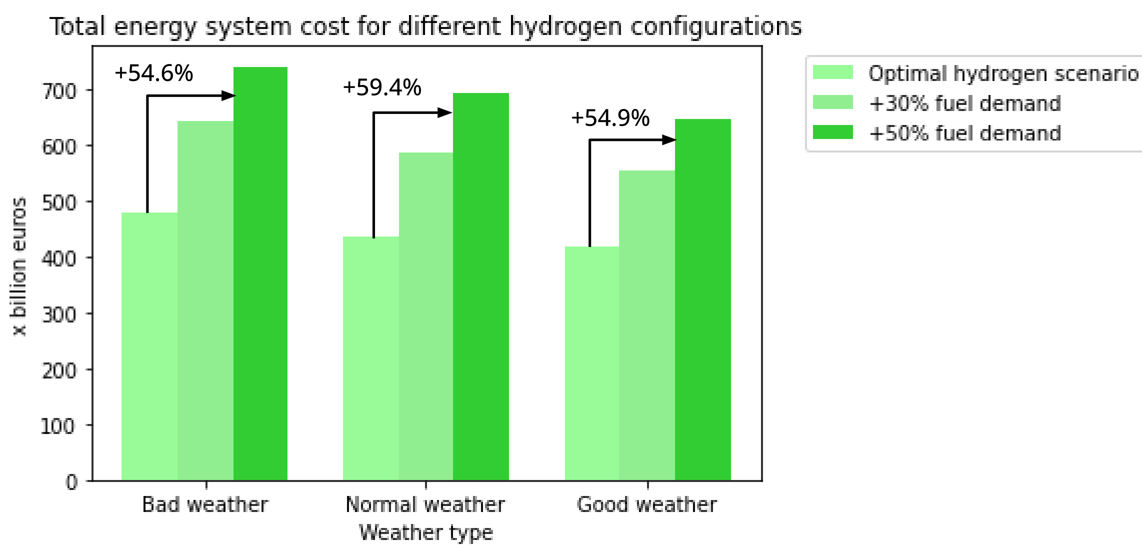


Figure 29: Total energy system cost for different hydrogen configurations

5.2.4 LCOE of technologies

The levelized cost of energy (LCOE) of technologies for the production of electricity in different hydrogen configurations in a normal weather scenario has been calculated and presented in Figure 30. The LCOE is a metric to measure the average cost to produce a unit of energy during its entire lifetime. The LCOE for offshore windfarms are €47.05/MWh, €45.88/MWh and €45.78/MWh for the optimal, slightly increased and increased fuel demand respectively. For onshore windfarms, the LCOE are €28.84/MWh, €30.55/MWh and €31.60/MWh respectively and are thus cheaper than offshore windfarms. The LCOE for technologies do not differ much when increasing the fuel demand within the energy system except for batteries. From Figure 30 it can be seen that the LCOE for batteries decrease with increased fuel demand. Figure 25 shows that electricity demand from batteries is decreasing with increased fuel demands. A possible explanation for the reduction of LCOE for batteries could be an increased use of batteries for the same number of installed capacities. As electricity surpluses are more frequent, batteries are used more frequently as well. As a result, more energy is discharged during its lifetime lowering the LCOE. The energy flows for batteries should therefore be analysed in future studies to validate this hypothesis.

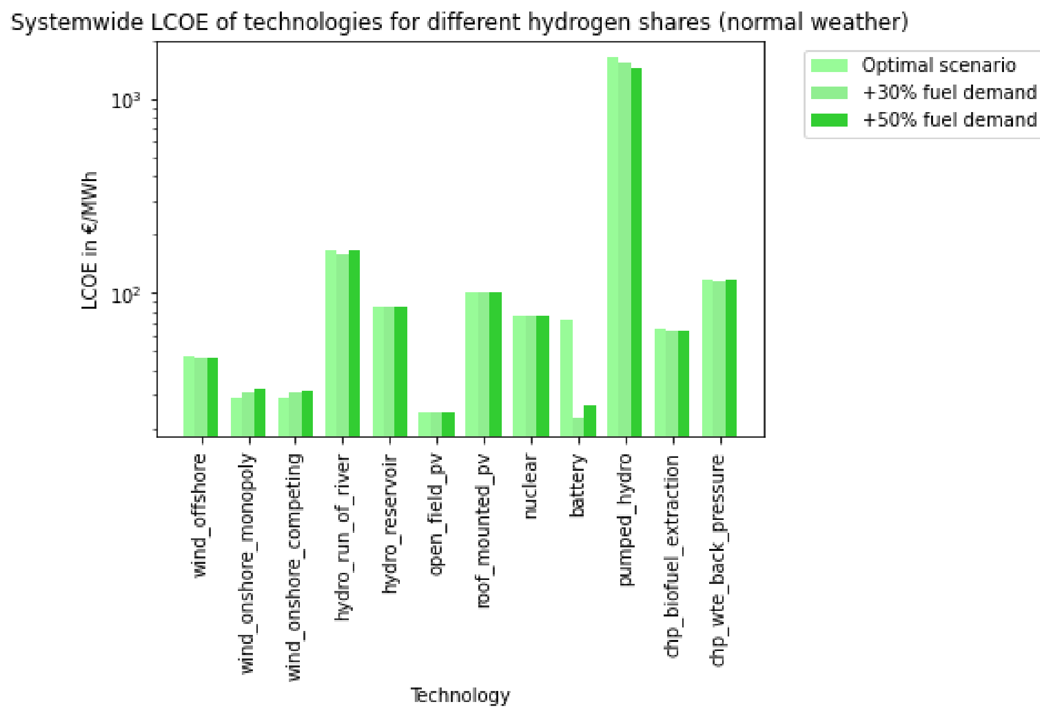


Figure 30: Levelized cost of energy of technologies for the production of electricity in different hydrogen configurations

The LCOE for electrolysis remain mostly stable for all configurations as can be observed in Figure 31. The largest difference is observed in the normal weather configuration where the LCOE changes from €13.73/MWh to €15.01/MWh from the optimal hydrogen configuration to 50% fuel demand increase configuration respectively. For all other configurations the LCOE ranges from €14.13/MWh to €14.72/MWh. This indicates a relative stable capacity deployment of the installed capacity for electrolysis assuming that the installed capacity increases with increased demand.

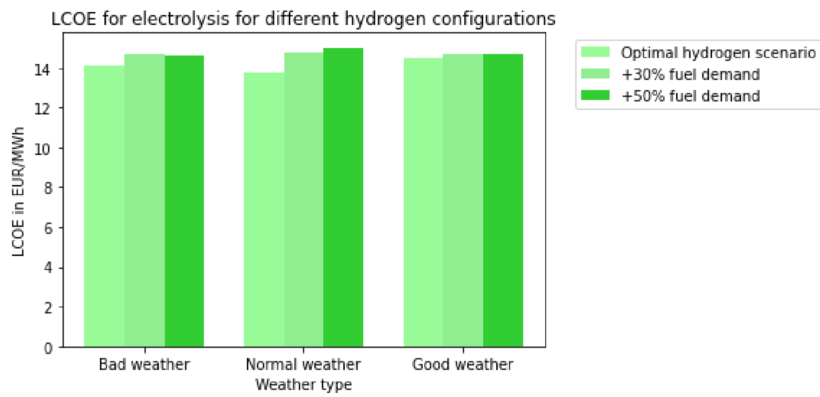


Figure 31: LCOE for electroysis for different hydrogen configurations

5.2.5 Price stability

A boxplot is used to analyse the stability of electricity price in the Netherlands for the three weather scenarios. The median is represented by the green line, the mean is represented by the green triangle. It is possible to observe the change in the spread for shadow prices for electricity in boxplots and therefore give an indication of the stability of the price dynamics. From Figure 32, it can be observed that the price is the most stable for a normal weather condition with minimal spread. The average and median shadow prices for electricity are the lowest for good weather scenarios and the highest for bad weather scenarios. It is interesting to note that for a normal weather scenario, the shadow prices do not go to zero whereas for bad weather and good weather scenarios the shadow prices can be found more often to be zero. This could indicate that the surpluses of electricity in both bad weather and good weather scenarios is more frequent causing the shadow prices to drop. Moreover, in bad weather the shortage of electricity is more frequent compared to good weather and normal weather scenarios causing more frequent price peaks.

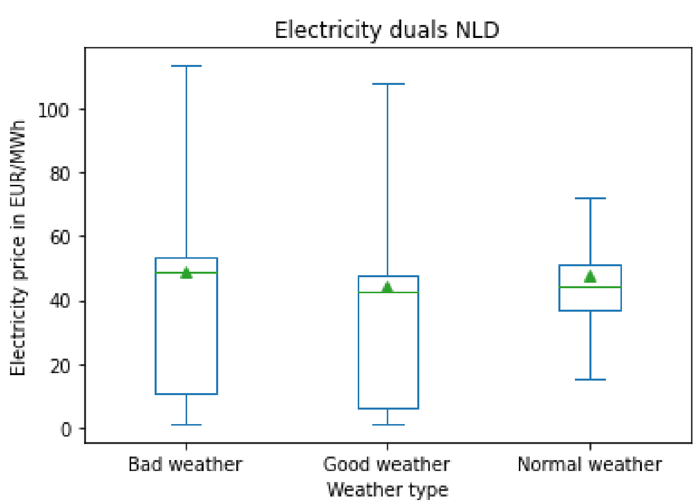


Figure 32: Boxplot of electricity duals in NLD for different weather types

The boxplots for the full configuration set are presented in Figure 33. It can be generally observed that the spread of shadow prices decreases as the fuel demand increases. This implies that the increase of hydrogen share within an energy system contributes to a higher price stability meaning that the occurrence of extreme price peaks both towards the upside and downside is reduced. This improvement in price stability is more significant in bad weather scenarios relative to and normal weather scenarios. This can be seen by the relatively more occurrences of shadow prices above 80 euros per MWh in the optimal scenario during bad weather compared to normal weather conditions. These extreme price peaks seemed to have disappeared when the fuel share demand is increased by 30% and 50%. Furthermore, the interquartile range for the bad weather scenario with 50% increase in fuel share demand seems to be even the smallest amongst all data sets. Although all configurations seem to have improved price stability from the increase in hydrogen share within the energy system. The extreme weather cases such as the bad weather and good weather scenarios seems to benefit more relatively.

Regarding the average shadow prices, it can be observed that the average prices are the highest and lowest for bad weather scenarios and good weather scenarios respectively. Furthermore, it can also be observed that the average shadow price is lower in all scenarios with an increased share of hydrogen compared to the optimal hydrogen configuration. This implies that although the total system cost might be higher for configurations with a higher hydrogen share, the average cost for specific energy carriers, electricity in this case, does not necessarily increase. On the contrary, the average shadow price for electricity decreases. More specially, the average shadow prices for electricity decrease by 1.9%, 3.2% and 2.7% for bad, good and normal weather scenarios respectively when comparing the optimal scenario against the scenario with 50% fuel demand increase.

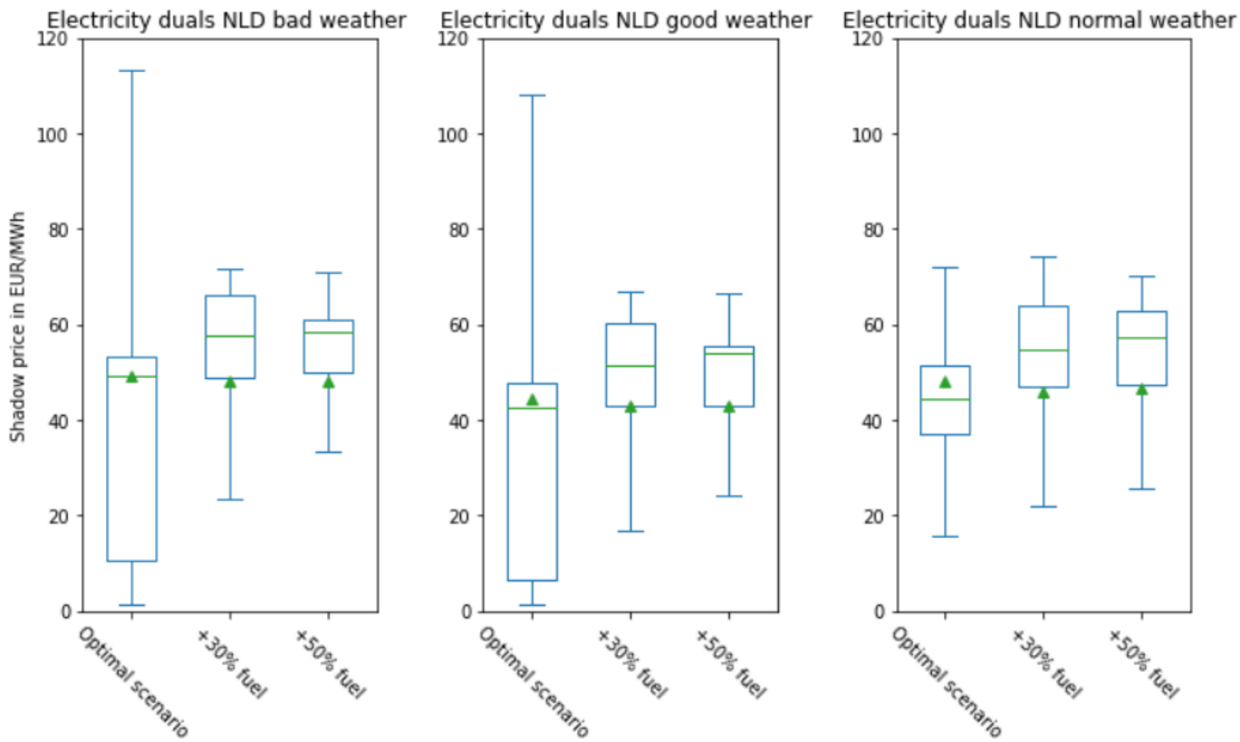


Figure 33: Comparison of electricity duals between the optimal cost scenario and 80% hydrogen share scenario

A closer look at peak electricity shadow prices

To understand the nature of peak electricity shadow prices, the time-series data is plotted against the electricity production by the different sources. This time-series data for the whole year for a optimal hydrogen share configuration in normal weather scenario is presented in Appendix I – Electricity production vs electricity shadow price.

It can be observed that the peak electricity shadow prices only occur during the winter months November through February. Furthermore, it can also be observed that during the summer months, the electricity shadow prices drops to near-zero values, but never negative. This indicates that during winter months, electricity supply is scarce resulting in higher electricity shadow prices and that electricity supply is in surplus during the summer months causing the electricity shadow prices drop. Such price behaviour is typical for VRES such as solar and wind. When looking at the electricity production, it can indeed be observed that wind is the dominant power source for electricity followed by solar.

Electricity shadow prices with increased fuel demands

To understand what happened to the peak electricity prices when fuel demands are increased, the time-series data is plotted against the electricity production by different sources for both the optimal hydrogen share configuration and +50% fuel demand configuration. The analysis is done for the normal weather scenario in both cases. The graphs are presented in Figure 34 for the optimal hydrogen share scenario and in Figure 35 for the 50% increased fuel demand. The time-series graph are zoomed in on the month February where the occurrence of high electricity shadow price peaks are more frequent.

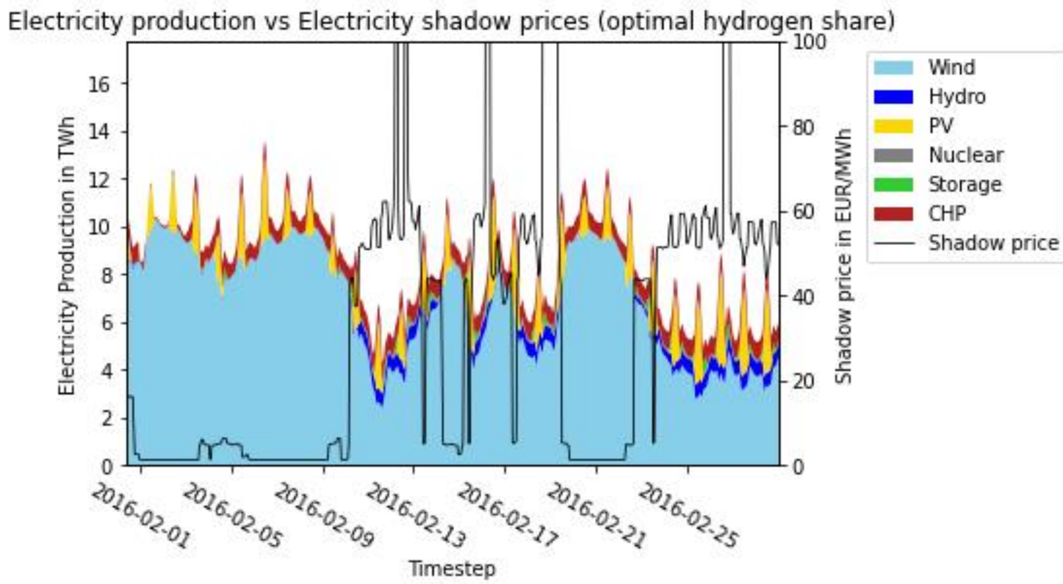


Figure 34: Electricity production by source vs Electricity shadow prices in the optimal scenario

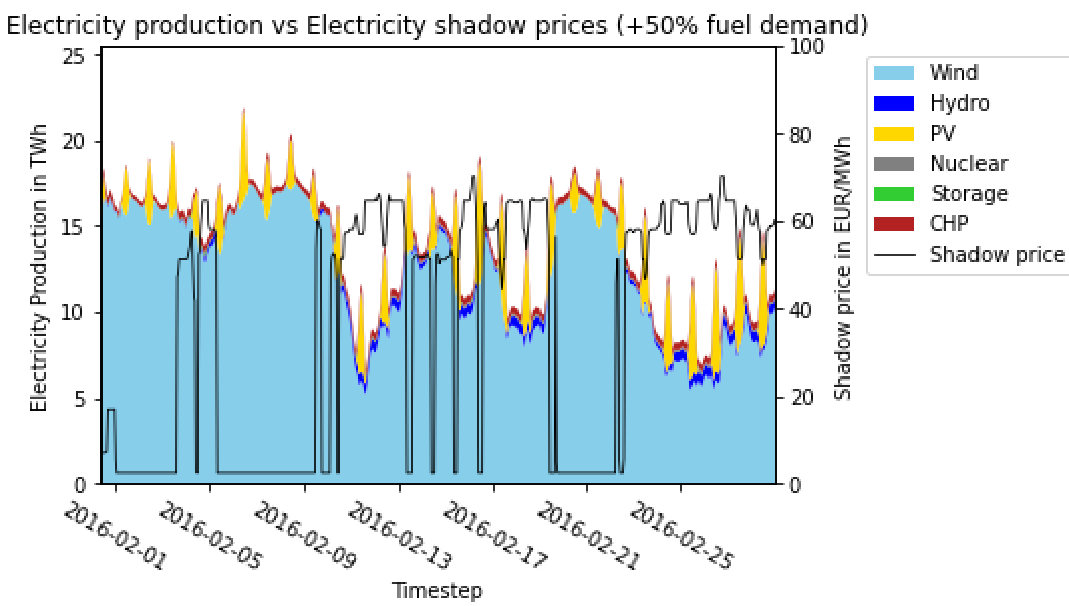


Figure 35: Electricity production by source vs Electricity shadow prices (+50% fuel demand)

The increase in the production of electricity through VRES is a direct consequence of the increase in fuel demand. This is because electricity is used to make hydrogen and hydrogen is used to create synthetic fuels. The significant increase of required VRES need in the form of wind and PV is a consequence of low employment of storage, biofuel, and curtailment. The increase of VRES causes a longer duration of oversupply of electricity. As a result, the price peaks are eliminated and the average price of electricity decreases.

5.2.6 Price and load duration curves

Both the price duration and load duration curves from the Netherlands are analysed to get a better understanding of the electricity market for the different configurations.

Price duration curves

The price duration curve for the optimal hydrogen share configuration across the different weather scenarios is plotted in Figure 36. When looking at Figure 36, one can observe that the electricity shadow price has the longest duration within the 40 EUR/MWh and 60 EUR/MWh range. Moreover, for 70-80% of the time the electricity shadow price is in the range of 20 EUR/MWh and 40 EUR/MWh.

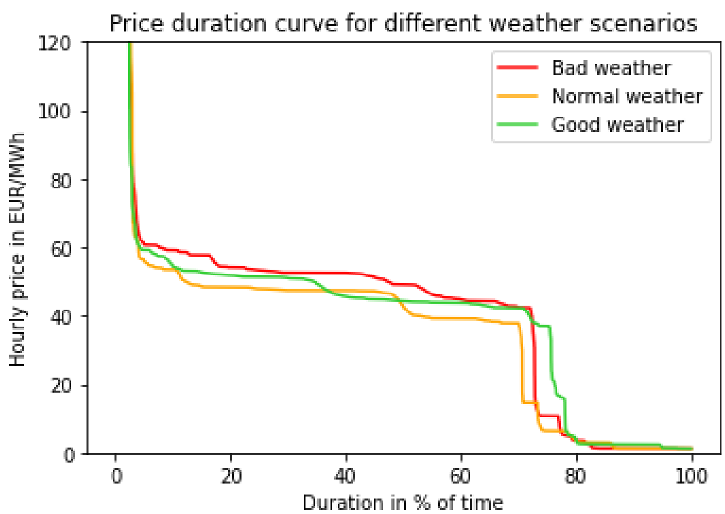


Figure 36: Price duration curve for different weather scenarios

The price duration curves for the increased fuel demands are plotted in Figure 37. It can be observed that with an increase of 30% in the fuel demand, price peaks disappear for the bad and normal weather scenarios. In good weather scenario, there are 4 hours in of price peaks where shadow prices exceed 100 EUR/MWh. Furthermore, it is interesting to note that the price peaks that were in the optimal hydrogen share configurations completely disappears when the fuel demand increases to 50%. The highest shadow price for electricity caps at roughly 80 EUR/MWh for all weather scenarios.

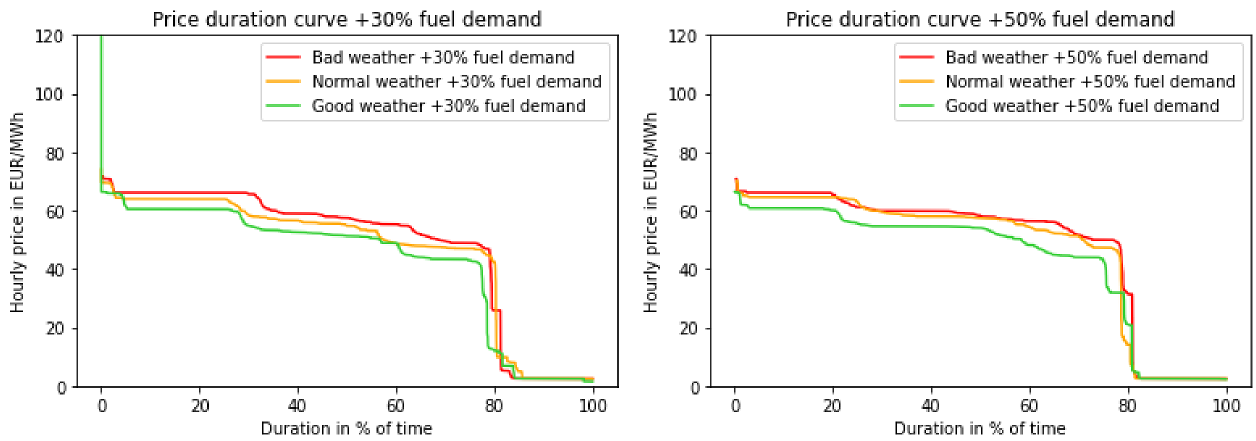


Figure 37: Price duration curve for different weather scenarios with increased fuel demands

Load duration curves

The load duration curve for the optimal share of hydrogen for all the weather scenarios is presented in Figure 38. The time within the duration curves can be divided into three periods. The off-peak hours which represents the 5000 hours per year with the lowest demand. The peak hours which represents the 160 hours per year with the highest demand and the shoulder hours, which represent the remaining 3600 hours in between off-peak and peak demand per year.

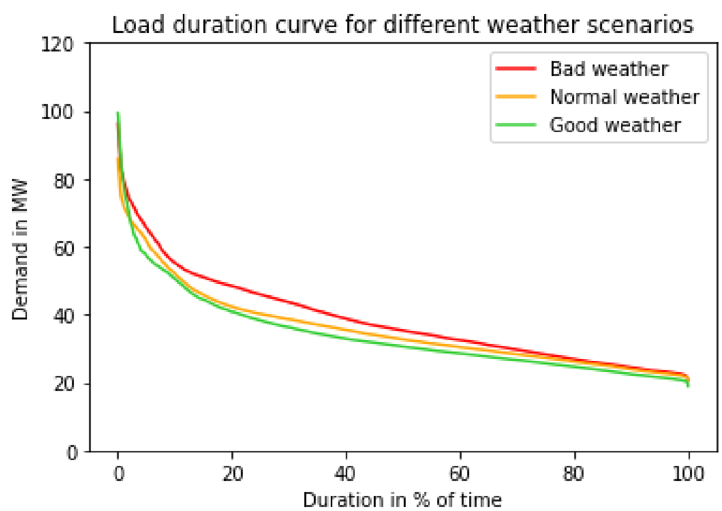


Figure 38: Load duration curve for different weather scenarios (optimal hydrogen share)

The load duration curves for the increased fuel demand are presented in Figure 39. It can be observed that the slope for off-peak hours in the increased fuel demand configurations are a lot steeper compared to the optimal hydrogen share configurations. The demand ranges for the periods can be found in Table 11.

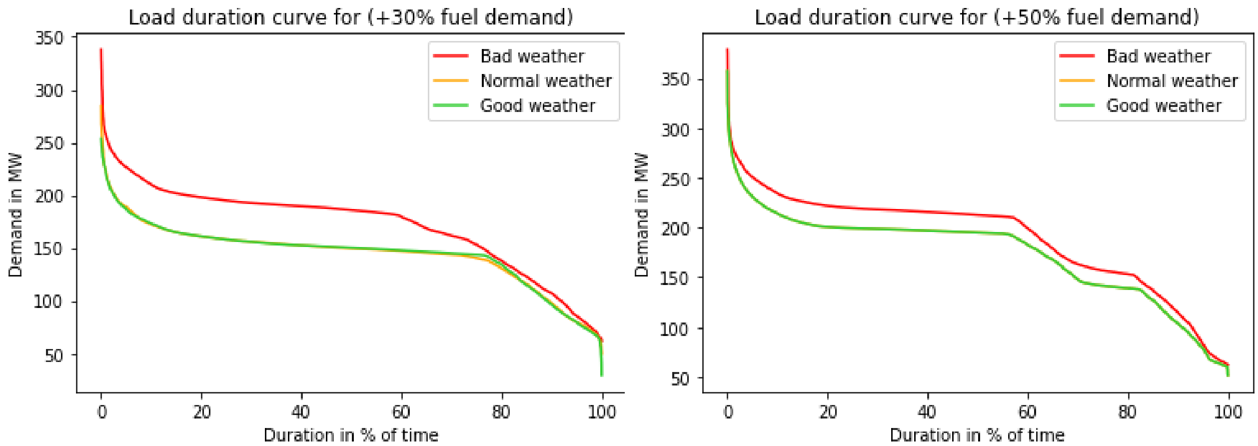


Figure 39: Load duration curves for increased fuel demands

Table 11: Demand range within the load duration curves

	Demand range in MW		
Configuration	Off-peak hours	Shoulder hours	Peak hours
1	20.8 – 37.6	37.6 - 74.2	74.2 – 96.3
2	20.4 – 34.5	34.5 - 69.2	69.2 – 85.9
3	18.9 – 32.1	32.1 – 71.0	71.0 – 99.25
4	61.69 – 188.7	188.7 – 244.5	244.5 – 337.9
5	49.8 – 151.4	151.4 – 205.3	205.3 – 284.4
6	29.4 – 151.6	151.6 – 206.1	206.1 – 253.4
7	61.9 – 214.9	214.9 – 271.6	271.6 – 378.6
8	58.0 – 175.9	175.9 – 232.6	232.6 – 303.8
9	51.6 – 196.2	196.2 – 257.6	257.6 – 357.1

5.2.7 Payback time

The payback time is a metric to measure the length of time required to recover the cost of the initial investment. It can be used to test whether the total system will be profitable or not at the end of its lifetime. Here we analyse the difference in payback time for the different system configurations.

The payback time can be calculated using the following equation:

$$t = \frac{C_{investment}}{R - C_{var}} \tag{10}$$

Where t is the payback time in years, $C_{investment}$ is the investment cost, R the revenue and C_{var} the variable cost in €.

The revenue is calculated by multiplying the average shadow price by the total electricity supplied in the energy system. For simplification purposes, it is assumed that the whole region has a shared electricity price and the shadow price for electricity from the Netherlands is used. In the real-world electricity prices vary by country. An overview of the payback time for the different configurations is presented in Table 12. It can be observed that the payback time does not vary significantly across the different configurations.

Table 12: Overview payback time energy system for the different configurations

Configuration	1	2	3	4	5	6	7	8	9
Payback time (months)	10.1	9.9	9.7	10.0	9.8	9.6	9.9	9.6	9.6

Validation of payback time

Information on the payback time for an entire energy system cannot be found in the literature. However, the payback time for wind turbines is widely studied. Dammeirer et al. (2019) state that the payback time on wind turbines in north-western Europe average at 5.3 months. The model results for the payback time for onshore windfarms is 7.4 months.

Chapter 6 – Discussion

This chapter discusses the outcomes of this master research project and highlights the key limitations of these outcomes.

6.1 Extraction method for shadow prices within the Calliope modelling framework

The first outcome of this study is the development of a method to extract shadow prices from a LP problem within the Calliope framework which has been described in chapter 3. Although the extraction method of dual variables is not unique, the successful implementation of the extraction method within Calliope is novel. This includes the storing of dual variables and the reformatting of data frames to ease data analysis processes. The second outcome is described in chapter 4 and it includes the refinement process of a North Sea Calliope model in which the already established Euro-Calliope model has been scaled down to a prototype North Sea Calliope model to focus on Dutch-related and North Sea related research questions. This includes an extensive debugging process where it is made sure that the model is scaled down properly from the Euro-Calliope model, making sure that the model was feasible and able to run at a 1-hour resolution for a whole year. The North Sea Calliope model is currently fully re-usable and can be uploaded on GitHub as an open-source model online in the academic community.

6.1.1 North Sea Calliope 2020 model limitations

The main limitation for North Sea Calliope model for the year 2020 is that the model is a difficult model to run due to the many constraints imposed on it. After an extensive debugging process, the model is currently debugged enough to produce feasible results in terms giving the right objective values, feasible dual variables values and similar physical total energy supply shares relative to real-world. However, the electricity duals currently seem not representative to the real-world values. Possible causes causing this mismatch in representation could be related to the important fact that the current North Sea Calliope models do not consider ramping costs, shutdown- and start up-costs, and cross region trading. Although the Euro-Calliope model is European-wide high-temporal and high-spatial resolution sector-coupled model, one cannot forget that it is still a simplification of the real-world. It is expected that the issue lies at the 2020 model specially rather than the North Sea Calliope model. More specifically, the issues are only present when the model implements 2020 override files. The first problem occurs when removing sectors from the North Sea Calliope. In essence what happens when for example, the synthetic fuel production for hydrocarbons such as methane, methanol, kerosene and diesel production from biofuel gets removed, the demand for the hydrocarbon fuels have to be met with traditional fossil fuels. Forcing the model to have a higher share of fossil fuels without compensating for the loss of other production options leads to an unsolvable model. Moreover, in the default Calliope model, all fuel share constraints are set to be equality constraints. These stringent constraints have to be relaxed to a inequality constraint in order for the model to be solvable. In essence, the share of fossil fuels now only need to fulfil a minimum demand instead of an exact demand. The second problem is related to the numerical issues where the solver is encountering a numerical trouble due to the high number of parallel constraints. This could be solved running a slower, but more stable Gurobi algorithm.

6.1.2 Best method for the extraction of shadow prices within the Calliope modelling framework

Within this research project, only the Pyomo class within the Python language has been used and tested for the extraction of shadow prices. Other classes used for optimisation modelling such as PuLP and SciPy have not been used nor been tested within the Calliope modelling framework due to project time constraints and misalignment of research scope. It is therefore not possible to determine whether the Pyomo class is the most efficient class for the extraction of shadow prices.

6.2 Assessing trade-offs within different hydrogen configurations

6.2.1 Weather

The weather scenarios, bad weather, normal weather and good weather within this research project are defined based on the objective value of the optimization model rather than true meteorological data. It is assumed that during bad weather, cheaper sources for energy such as VRES are available to a lesser extent when compared to a good weather. From the results, such as Figure 26 it can be observed that in a good weather there is more onshore wind production compared to normal weather while one would expect that good weather is typically accompanied by less wind relative to normal weather.

6.2.2 Capacity deployment in optimization models

The North Sea Calliope model is a optimization model. It takes technology and demand data to solve a LP problem to find a region-wide optimal value for the objective function given the constraints rather than finding an optimal value for each region and each carrier independently. What this means is that the model always tries to deploy the cheapest technologies first as long as the constraints such as available energy capacity is not exceeded. The result of such optimization model is that technologies such as onshore wind are heavily favoured in comparison to offshore wind leading to a limited deployment of offshore wind. This leads to inaccuracies for not only location-specific capacity deployment such as in the Netherlands, but also region-wide for the whole North Sea region. While the Infrastructure Outlook 2050 from TenneT (2019) reports a higher share of offshore wind compared to onshore wind in the Netherlands, the model gives a higher share of onshore wind due to its lower cost. Moreover, the model overestimates to capacity deployment of onshore wind and underestimate the capacity deployment of offshore wind for the whole North Sea region. Furthermore, the hydrogen capacity for the North Sea region are also higher than the highest estimates of hydrogen use in future 2050 scenarios.

6.2.3 Price stability and electricity shadow prices

The first results from using electricity carrier price information in assessing trade-offs between different configurations have shown that different weather types affect the price stability of electricity. It is not a surprise that electricity prices are affected by different weather conditions, but the use of shadow prices to assess these effects is unique. The results have shown that different shares of hydrogen within an fully sector-coupled energy system could potentially have a stabilising effect on electricity prices. Furthermore, the results have also shown that an increase in hydrogen shares could potentially reduce the average cost of electricity prices despite an increase in the total system costs.

By using energy carrier price information, policymakers can now assess trade-offs in fully sector-coupled energy models in a rather unique way. Where in the past shadow prices are mainly used to assess the hidden cost of non-marketable goods such as GHGs in policy implementations, this research project has shown that shadow prices can be used to analyse trade-offs in different configurations of fully sector-coupled energy models. More specifically, it analyses the stability in price dynamics in different hydrogen configurations in future energy system scenarios. Most energy optimisation models focus on the trade-off between total energy system costs and different energy system configurations, but not the trade-off between price stability and different energy system conditions. Having insights on the price stability of energy systems could potentially favour energy system configuration options where the total costs are higher, but the stability in price is also better. Although the effects on the electricity price dynamics could be analysed using shadow prices in different configurations of fully sector-coupled energy systems. The prices cannot be validated as the scenarios are future-based scenarios. Furthermore, in this study only price information of electricity has been studied extensively. Although the method described in this report could also be used on other energy carriers, this has not been done yet. Moreover, this master thesis project has mainly focussed on power-to-hydrogen and the effects of different hydrogen shares within an energy system. As mentioned in Section 1.2, other forms of sector coupling such as power-to-heat, power-to-gas also have the potential to increase the efficiency and reduce the overall cost of the whole energy system. Within the literature there is a wide consensus that the sector-coupling of the electricity sector and hydrogen sector would contribute to the decarbonisation of energy systems and reduction of electricity prices. Future studies could therefore include the analysis of these other forms of sector coupling to understand the full dynamics within fully sector-coupled energy systems.

Chapter 7 – Conclusions & Recommendations

The goal of this master thesis project is to understand trade-offs between different configurations of a fully sector-coupled energy system model using energy carrier price information. In order to do this, a research proposal was submitted to answer 4 related sub questions and 1 main research question within a total time span of 24 weeks. The main research question stated is: *"How can energy carrier price information be used to understand the trade-offs between different configurations of fully sector-coupled energy system models?"*. The main research question and sub questions are revisited in this chapter. Also recommendations for improvements and future research is given.

7.1 Sub question 1

"What are the methods to extract energy carrier prices from fully sector-coupled energy systems in existing literature"

It turned out that the method found need not to be novel in order to fulfil the main research question. It uses an existing method adapted from the Pyomo optimisation modelling package within the Python programming language. Other potential methods involve the use of the optimisation packages PuLP and SciPy, but these have not been used or tested in this research project since the scope does not include the modelling optimisation. Future research focussed on the optimisation of modelling methods can further analyse of the implementation of these optimisation modelling classes.

7.2 Sub question 2

"How can energy carrier prices in the form of shadow prices be extracted within the Calliope framework in fully sector-coupled energy systems?"

To answer this sub question, first a small-scale Calliope model has been used to test the Python codes for the extraction of energy carrier prices. In the second step, the North Sea Calliope model has been created within the Calliope modelling framework to extract energy carrier prices within a large-scale fully sector-coupled energy system. The North Sea Calliope model is a scaled-down version of the Europe-Calliope model. It consists of 10 nodes, one node for each country, covering all major stakeholders in North Sea projects. The included countries are Belgium, Germany, Denmark, France, Great Britain, Ireland, Luxembourg, the Netherlands, Norway and Sweden. To run 1-hour resolution models, the DelftBlue supercomputer has been used.

Due to the time constraints of this master thesis project, some of the presented methods are not optimised or are developed to perform a single task. Python code currently developed for reformatting the data frame of dual variables is done by looking manually which data and in what column the data should be split into. This is currently only tested on the system balance dual variables. It can therefore not be guaranteed that this method works for the analysis of other dual variables. A more general approach such as a general function in the post-processing step is therefore recommended to reformat the data frame of dual variables. This could also potentially reduce the total running time of a model.

7.3 Sub question 3

"How do the generated price information compare the real-world price information?".

A North Sea Calliope model using technology data from the Danish Energy Agency for the year 2020 has been used to create the North Sea Calliope 2020 model. In combination with weather data from the year 2015, the shadow prices for electricity for the Netherlands are extracted. The extracted data is compared to the ENTSO-E electricity day-ahead prices from the Netherlands for the year 2015, matching the weather year. It turned out that the model data is not an accurate representation of real-world prices. More specifically, the electricity prices in the model data are flatter and are less dynamic compared to real-world data. This could be caused by numerous things, among others, it could be caused by the fact that the model is a simplification of the real-world without ramping costs, shutdown costs and start-up costs. Furthermore, the 2020 North Sea Calliope model is a difficult model to run due to the many constraints what leads to more unrealistic dual variable values due to the manual disabling of locations, technologies and other constraints. The efficiency both numerically and size of the 2020 North Sea Calliope model could therefore be improved by applying a bottom-up approach to build a stand-alone North Sea Calliope model in future research. Also, the inclusion of non-linear constraints such as ramping costs, start-up and shutdown cost is suggested for future research for a more complete model.

7.4 Sub question 4

"What are the price-related trade-offs when varying the share of hydrogen in a fully sector-coupled energy system?"

This master thesis project has tried to assess whether a higher share in hydrogen in the total energy system could lead to more stable price dynamics when looking at the shadow prices of electricity. More specifically, different hydrogen configurations where the fuel demand increases 30% and 50% relative to the optimal hydrogen share configurations has been analysed for bad, good and normal weather scenarios. Improvement in the price stability was observed for all weather scenarios when fuel demand has been increased, meaning the occurrence of price peaks both to the top and bottom are reduced. Increased fuel demand is also coupled with an increase in the electricity supply by VRES resulting in a higher frequency of oversupply of electricity. The increased demand for electricity is caused by the increased demand for electrolysis to create hydrogen. As an effect, electricity shadow prices are more often zero when compared with optimal hydrogen share configurations. Furthermore, it has also been observed that the average electricity prices decreases with an increased fuel demand while the overall LCOE for the technologies and overall payback time for the energy system remain relatively unchanged.

7.5 Main research question

"How can energy carrier price information be used to understand the trade-offs between different configurations of fully sector-coupled energy system models?".

This master thesis project has shown that it is possible to use shadow prices to analyse trade-offs between different configurations of a fully sector-coupled energy system model. This includes price stability, price peaks and price duration analyses of the energy system. Although the developed Calliope model is not yet entirely bug free, it offers a great start and foundation for future research. The Calliope modelling framework offers great flexibility in modelling from which many different parameters can be analysed including investment decision support, operation decision support and scenario analyses. The North Sea Calliope model can be used to research North Sea related topics. Furthermore, in this master thesis project, the energy carrier 'electricity' has been extensively researched using the developed methods. However, the same methods could also be applied to analyse the trade-offs in different configurations of fully sector-coupled energy systems for different energy carriers such as heat or hydrogen.

Bibliography

Althammer, W., & Hille, E. (2016). Measuring climate policy stringency: a shadow price approach. *International Tax and Public Finance*, *23*(4), 607-639.

Aryanpur, V., O'Gallachoir, B., Dai, H., Chen, W., & Glynn, J. (2021). A review of spatial resolution and regionalisation in national-scale energy systems optimisation models. *Energy Strategy Reviews*, *37*, 100702.

Brown, T., Schlachtberger, D., Kies, A., Schramm, S., & Greiner, M. (2018). Synergies of sector coupling and transmission reinforcement in a cost-optimised, highly renewable European energy system. *Energy*, *160*, 720-739.

Blauwe waterstof effent weg voor groene waterstof. (n.d.). TNO. Retrieved July 3, 2022, from https://www.tno.nl/nl/aandachtsgebieden/energietransitie/roadmaps/co2-neutrale-industrie/waterstof-voor-een-duurzame-energievoorziening/blauwe-waterstof-effent-weg-voor-groene-waterstof/#:%7E:text=Groene%20waterstof%20is%20echter%20momenteel,waterstof%20op%20de%20langere%20termijn.

Bloess, A., Schill, W. P., & Zerrahn, A. (2018). Power-to-heat for renewable energy integration: A review of technologies, modeling approaches, and flexibility potentials. *Applied Energy*, *212*, 1611-1626.

Caglayan, D. G., Ryberg, D. S., Heinrichs, H., Linßen, J., Stolten, D., & Robinius, M. (2019). The techno-economic potential of offshore wind energy with optimized future turbine designs in Europe. *Applied energy*, *255*, 113794.

Chapman, A., Itaoka, K., Hirose, K., Davidson, F. T., Nagasawa, K., Lloyd, A. C., ... & Fujii, Y. (2019). A review of four case studies assessing the potential for hydrogen penetration of the future energy system. *International journal of hydrogen energy*, *44*(13), 6371-6382.

Dammeier, L. C., Loriaux, J. M., Steinmann, Z. J., Smits, D. A., Wijnant, I. L., van den Hurk, B., & Huijbregts, M. A. (2019). Space, time, and size dependencies of greenhouse gas payback times of wind turbines in northwestern europe. *Environmental Science & Technology*, *53*(15), 9289-9297.

Data & Statistics. (n.d.). IEA. Retrieved July 3, 2022, from https://www.iea.org/data-and-statistics/data-browser?country=NETHLAND&fuel=Energy%20supply&indicator=TESbySource

Delft High Performance Computing Centre (DHPC), DelftBlue Supercomputer (Phase 1), 2022, https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1

Detz, R. J., Lenzmann, F. O., Sijm, J. P. M., & Weeda, M. (2019). Future Role of Hydrogen in the Netherlands. *A meta-analysis based on a review of recent scenario studies*.

Diewert, W. E. (1974). Applications of duality theory.

Documentation. (2019, January 18). Gurobi. Retrieved July 3, 2022, from https://www.gurobi.com/documentation/9.5/refman/method.html#parameter:Method

Enevoldsen, P., Permien, F. H., Bakhtaoui, I., von Krauland, A. K., Jacobson, M. Z., Xydis, G., ... & Oxley, G. (2019). How much wind power potential does europe have? Examining european wind power potential with an enhanced socio-technical atlas. *Energy Policy*, *132*, 1092-1100.

Foxon, T. J., Hammond, G. P., & Pearson, P. J. (2010). Developing transition pathways for a low carbon electricity system in the UK. *Technological Forecasting and Social Change*, *77*(8), 1203-1213.

GitHub - calliope-project/euro-calliope-2.0: Sector coupled Euro-Calliope (n.d.). GitHub. Retrieved March 29, 2022, from https://github.com/calliope-project/euro-calliope-2.0

Glasgow Climate Pact. (2021, November 13). Glasgow Climate Pact | UNFCC. Retrieved March 30, 2022, from https://unfccc.int/documents/310497

Hörsch, J., & Brown, T. (2017, June). The role of spatial scale in joint optimisations of generation and transmission for European highly renewable scenarios. In *2017 14th international conference on the European Energy Market (EEM)* (pp. 1-7). IEEE.

Infrastructure Outlook 2050. (2019, February 14). TenneT. Retrieved August 5, 2022, from https://www.tennet.eu/fileadmin/user_upload/Company/News/Dutch/2019/Infrastructure_Outlook_2050_appendices_190214.pdf

Laha, P., & Chakraborty, B. (2021). Cost optimal combinations of storage technologies for maximizing renewable integration in Indian power system by 2040: Multi-region approach. *Renewable Energy*, *179*, 233-247.

Lee, M., & Zhang, N. (2012). Technical efficiency, shadow price of carbon dioxide emissions, and substitutability for energy in the Chinese manufacturing industries. *Energy Economics*, *34*(5), 1492-1497.

Lombardi, F., Rocco, M. V., & Colombo, E. (2019). A multi-layer energy modelling methodology to assess the impact of heat-electricity integration strategies: The case of the residential cooking sector in Italy. *Energy*, *170*, 1249-1260.

Lombardi, F., Pickering, B., Colombo, E., & Pfenninger, S. (2020). Policy decision support for renewables deployment through spatially explicit practically optimal alternatives. *Joule*, *4*(10), 2185-2207.

Lopion, P., Markewitz, P., Robinius, M., & Stolten, D. (2018). A review of current challenges and trends in energy systems modeling. *Renewable and sustainable energy reviews*, *96*, 156-166.

Luz, G. P., & e Silva, R. A. (2021). Modeling energy communities with collective photovoltaic self-consumption: Synergies between a small city and a winery in Portugal. *Energies*, *14*(2), 1-26.

Mathematical formulation — Calliope 0.6.8 documentation. (n.d.). Calliope. Retrieved July 6, 2022, from https://calliope.readthedocs.io/en/stable/user/ref_formulation.html

Mangipinto, A. (2020). Development of electric vehicles load profiles for sector coupling in European energy system models.

Martínez-Gordón, R., Morales-España, G., Sijm, J., & Faaij, A. P. C. (2021). A review of the role of spatial resolution in energy systems modelling: Lessons learned and applicability to the North Sea region. *Renewable and Sustainable Energy Reviews*, *141*, 110857.

Maruf, M., & Islam, N. (2019). Sector coupling in the north sea region—a review on the energy system modelling perspective. *Energies*, *12*(22), 4298.

Mazloomi, K., & Gomes, C. (2012). Hydrogen as an energy carrier: Prospects and challenges. *Renewable and Sustainable Energy Reviews*, *16*(5), 3024-3033.

Ministerie van Economische Zaken en Klimaat. (2019, June 28). *Klimaatakkoord hoofdstuk Waterstof*. Publicatie | Klimaatakkoord. https://www.klimaatakkoord.nl/themas/waterstof/documenten/publicaties/2019/06/28/klimaatakkoord-hoofdstuk-waterstof

Møller, K. T., Jensen, T. R., Akiba, E., & Li, H. W. (2017). Hydrogen-A sustainable energy carrier. *Progress in Natural Science: Materials International*, *27*(1), 34-40.

Morgenthaler, S., Kuckshinrichs, W., & Witthaut, D. (2020). Optimal system layout and locations for fully renewable high temperature co-electrolysis. *Applied energy*, *260*, 114218.

North Sea Energy. (n.d.). North Sea Energy. Retrieved July 3, 2022, from https://north-sea-energy.eu/en/home/

North Sea Wind Power Hub. (n.d.). NSWPH. Retrieved July 3, 2022, from https://northseawindpowerhub.eu/

Nationaal Waterstof Programma. Retrieved July 3, 2022, from https://nationaalwaterstofprogramma.nl/default.aspx

Our energy, our future. (2020, February 14). WindEurope. Retrieved August 5, 2022, from https://windeurope.org/about-wind/reports/our-energy-our-future/

Pavičević, M., Mangipinto, A., Nijs, W., Lombardi, F., Kavvadias, K., Navarro, J. P. J., ... & Quoilin, S. (2020). The potential of sector coupling in future European energy systems: Soft linking between the Dispa-SET and JRC-EU-TIMES models. *Applied Energy*, *267*, 115100.

Perry, C., & Crellin, K. C. (1982). The precise management meaning of a shadow price. *Interfaces*, *12*(2), 61-63.Díaz, P., Van Vliet, O., & Patt, A. (2017). Do we need gas as a bridging fuel? A case study of the electricity system of Switzerland. *Energies*, *10*(7), 861.

Pfenninger, S., Hawkes, A., & Keirstead, J. (2014). Energy systems modeling for twenty-first century energy challenges. *Renewable and Sustainable Energy Reviews*, *33*, 74-86.

Pfenninger, S., & Keirstead, J. (2015). Renewables, nuclear, or fossil fuels? Scenarios for Great Britain's power system considering costs, emissions and energy security. *Applied Energy*, *152*, 83-93.

Pfenninger, S., & Pickering, B. (2018). Calliope: a multi-scale energy systems modelling framework. *Journal of Open Source Software*, *3*(29), 825.

Pickering, B., Lombardi, F., & Pfenninger, S. (2022). Diversity of options to eliminate fossil fuels and reach carbon neutrality across the entire European energy system. *Joule*.

Power ONshore. (2021, February 26). WindEurope. Retrieved August 5, 2022, from https://windeurope.org/events/poweronshore/#:%7E:text=The%20EU%20will%20need%20750,up%20from%20173%20GW%20today.

Ringkjøb, H. K., Haugan, P. M., & Solbrekke, I. M. (2018). A review of modelling tools for energy and electricity systems with large shares of variable renewables. *Renewable and Sustainable Energy Reviews*, *96*, 440-459.

Rosen, M. A., & Koohi-Fayegh, S. (2016). The prospects for hydrogen as an energy carrier: an overview of hydrogen energy and hydrogen energy systems. *Energy, Ecology and Environment*, *1*(1), 10-29.

Schiebahn, S., Grube, T., Robinius, M., Tietze, V., Kumar, B., & Stolten, D. (2015). Power to gas: Technological overview, systems analysis and economic assessment for a case study in Germany. *International journal of hydrogen energy*, *40*(12), 4285-4294.

Slurm scheduler · Wiki · DHPC / docs · GitLab. (n.d.). GitLab. Retrieved July 3, 2022, from https://gitlab.tudelft.nl/users/sign_in

Statista. (n.d.). *Countries with the highest quality of electricity supply 2019*. Retrieved July 3, 2022, from https://www.statista.com/statistics/268155/ranking-of-the-20-countries-with-the-highest-quality-of-electricity-supply/

Hoogspanningsnet in Gelderland en de Flevopolder bereikt grens voor teruglevering van elektriciteit. (n.d.). TenneT. Retrieved July 3, 2022, from https://www.tennet.eu/nl/tinyurl-storage/nieuws/hoogspanningsnet-in-gelderland-en-de-flevopolder-bereikt-grens-voor-teruglevering-van-elektriciteit/

Technology Data. (2021, November 5). Energistyrelsen. Retrieved August 4, 2022, from https://ens.dk/en/our-services/projections-and-models/technology-data#:%7E:text=The%20Danish%20Energy%20Agency%20and,Energy%20Agency%20for%20energy%20projections.

Tröndle, T. (2020). Supply-side options to reduce land requirements of fully renewable electricity in Europe. *PloS one*, *15*(8), e0236958.

Tröndle, T., Lilliestam, J., Marelli, S., & Pfenninger, S. (2020). Trade-offs between geographic scale, cost, and infrastructure requirements for fully renewable electricity in Europe. *Joule*, *4*(9), 1929-1948.

Tröndle, Tim, Pfenninger, Stefan, & Pickering, Bryn. (2022). Capacity factor time series for solar and wind power on a 50 km^2 grid in Europe [Data set]. Zenodo. https://doi.org/10.5281/zenodo.6559895

Troubleshooting — Calliope 0.6.8 documentation. (n.d.). Calliope. Retrieved July 20, 2022, from https://calliope.readthedocs.io/en/stable/user/troubleshooting.html#

Verkenning aanlanding wind op zee (VAWOZ) - 2031–2040. (n.d.). RVO. Retrieved July 3, 2022, from https://www.rvo.nl/onderwerpen/bureau-energieprojecten/lopende-projecten/noz-2030/vawoz-2031-2040

WindEurope asbl/vzw. (2022, April 2). *Getting fit for 55 and set for 2050*. WindEurope. Retrieved August 5, 2022, from https://windeurope.org/intelligence-platform/product/getting-fit-for-55-and-set-for-2050/

Wei, C., Löschel, A., & Liu, B. (2013). An empirical analysis of the CO2 shadow price in Chinese thermal power enterprises. *Energy Economics*, *40*, 22-31.

Working with Pyomo Models — Pyomo 6.4.1 documentation. (n.d.). Pyomo. Retrieved July 3, 2022, from https://pyomo.readthedocs.io/en/stable/working_models.html

wetten.nl - Regeling - Klimaatwet - BWBR0042394. (2020, 1 January). wetten.nl. Retrieved November 17, 2021, from https://wetten.overheid.nl/BWBR0042394/2020-01-01

Appendices

Appendix A – Laptop specifications

System Manufacturer	HP
System Model	HP ZBook Studio G3
System Type	x64-based PC
Processor	Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz, 2592 Mhz, 4 Core(s), 8 Logical Processor(s)
Installed physical memory	16.0 GB

Appendix B – Calliope models

B1 - Simple model

The files to the simple model can be accessed on:
https://github.com/asow22/simple_model

B2 - North Sea Calliope model

The files to the North Sea Calliope model can be accessed on:
https://surfdrive.surf.nl/files/index.php/s/GOophfS1FWu9dtr

The North Sea 2020 Calliope model which is used in chapter 4 can be found in the 2020 folder. The North Sea 2050 Calliope model which is used in chapter 5 can be found in the 2050 folder.

Appendix C – Building the 2020 North Sea Calliope model

```
import create_input
import run
import calliope
import pandas as pd
from create_override import cap_results_to_override
from utils import process_system_balance_duals

#%%
# CREATING MODEL INPUTS
###

# Select optimisation horizon: 2020, 2030 or 2050
opt_horizon = 2020
path_to_model_yaml = '/home/asow/2022-05_20_lite_north-
sea/{}/model/national/model-2015.yaml'.format(opt_horizon)

# Define override scenarios
scenario_string = {}

scenario_string['2020-1h'] =
"industry_fuel,transport,heat,config_overrides,gas_storage,link_cap_1x,"\
                        "freeze-hydro-
capacities,heat_techs_2020,renewable_techs_2020,"\
                        "transformation_techs_2020,fossil-fuel-
supply,res_1h,"\
                        "add-biofuel,coal_supply,north_sea,"\
                        "kill-fancy-techs,fix-generation-
capacities,demand_share_fuel_current_min"

selected_scenario = '2020-1h'

# Generate and save model inputs
path_to_netcdf_of_model_inputs = '/home/asow/2022-05_20_lite_north-
sea/{}/national/inputs.nc'.format(opt_horizon)
model_input = create_input.build_model(path_to_model_yaml,
scenario_string[selected_scenario], path_to_netcdf_of_model_inputs)
```

```
#%%
# RUNNING THE MODEL & saving results (including duals)
###

path_to_netcdf_of_results = '/home/asow/2022-05_20_lite_north-
sea/results/north-sea_{}.nc'.format(selected_scenario)
model_run, duals = run.run_model(path_to_netcdf_of_model_inputs,
path_to_netcdf_of_results)
system_balance_duals = duals['system_balance_constraint Constraint']
balance_duals = process_system_balance_duals(system_balance_duals)
balance_duals.set_index('timestep',inplace=True)
balance_duals.to_csv('/home/asow/2022-05_20_lite_north-
sea/results/north_sea_balance_duals_{}.csv'.format(selected_scenario))
```

Appendix D – Description of North Sea Calliope 2020 override files

YAML file	Description
Countries	
locations	Defining country coordinates and available technologies per location
north_sea_overrides	Removing all non-North Sea countries from the model
Technologies (location dependent)	
biofuel-supply-2015	Defining biofuel supply
coal_supply	Defining coal supply
directional-rooftop	Defining roof mounted PV supply
fossil-fuel-supply	Defining fossil fuel supply
fuel-distribution	Defining synthetic fuel transmission between countries
fuel-group_constraints_2015	Defining annual fuel demand and industry techs
gas_storage	Defining underground methane storage
heat_group_constraints_2015	Defining maximum heat storage, annual waste supply and grouping of heat technologies
links	Defining the transmission links between countries
vehicle_group_constraints_2015	Defining annual transport distance and transport demand
Technologies	
brown-field-capacities	Allowing technologies to increase in energy capacity
config_overrides	Removing some of the technologies
demand_share	Defining minimum fuel share within the system
demand-techs	Defining demand technologies
heat-techs	Defining heat technologies
legacy-techs	Defining legacy technologies
link-techs	Defining transmission technologies
renewable-techs	Defining renewable technologies
storage-techs	Defining storage technologies
transformation-techs	Defining conversion technologies
transport-techs	Defining transport technologies
2020 overrides	
fix-current-national-capacities	Setting energy capacity technology data to year 2020
fix-demand-share_min	Setting demand share of technologies to 2020 values
heat-techs	Setting heat technology data to year 2020
kill-fancy-techs	Removing technologies that are not widely adopted in 2020
renewable-techs	Setting renewable technology data to year 2020
storage-techs	Setting storage technology data to year 2020
transformation-techs	Setting conversion technology data to year 2020

The number behind the YAML files indicate the weather year for which the technology data is taken.

Appendix E – Job script 2020 model 1h resolution

```
#!/bin/sh

#SBATCH --account=education-tpm-msc
#SBATCH --partition=compute
#SBATCH --time=24:00:00
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=16
#SBATCH --mem-per-cpu=10G
#SBATCH --mail-type=END

srun ../my_national_model_run_2020_1h.py

```

Appendix F – Override files for setting the hydrogen shares for the production of synthetic fuel

```
overrides:
    hydrogen_share_80:
        group_constraints:
            hydrogen_share_diesel:
                techs: [hydrogen_to_liquids]
                carrier_prod_share_min.syn_diesel: 0.8
            hydrogen_share_kerosene:
                techs: [hydrogen_to_liquids]
                carrier_prod_share_min.syn_kerosene: 0.8

            hydrogen_share_methanol:
                techs: [hydrogen_to_methanol]
                carrier_prod_share_min.syn_methanol: 0.8

            hydrogen_share_methane:
                techs: [hydrogen_to_methane]
                carrier_prod_share_min.syn_methane: 0.8
```

Appendix G – Override files for increasing the fuel demand

G.1 Setting fuel demand increase to 30%

```
overrides:
      demand_share_fuel_30:
        group_constraints:
            demand_share_fuel_cooking:
                techs: [gas_hob]
                demand_share_equals.cooking: 0.3
            demand_share_fuel_heat:
                techs: [methane_tech_heat_to_demand,
chp_methane_extraction_tech_heat_to_demand,
chp_methane_back_pressure_simple_tech_heat_to_demand,
chp_methane_back_pressure_combined_tech_heat_to_demand]
                demand_share_equals.heat: 0.3
            demand_share_fuel_transport_light:
                techs: [light_transport_ice]
                demand_share_equals.light_transport: 0.3
            demand_share_fuel_transport_heavy:
                techs: [heavy_transport_ice]
                demand_share_equals.heavy_transport: 0.3
            demand_share_fuel_electricity:
                techs: [ccgt]
                carrier_prod_share_max.electricity: 0.3
```

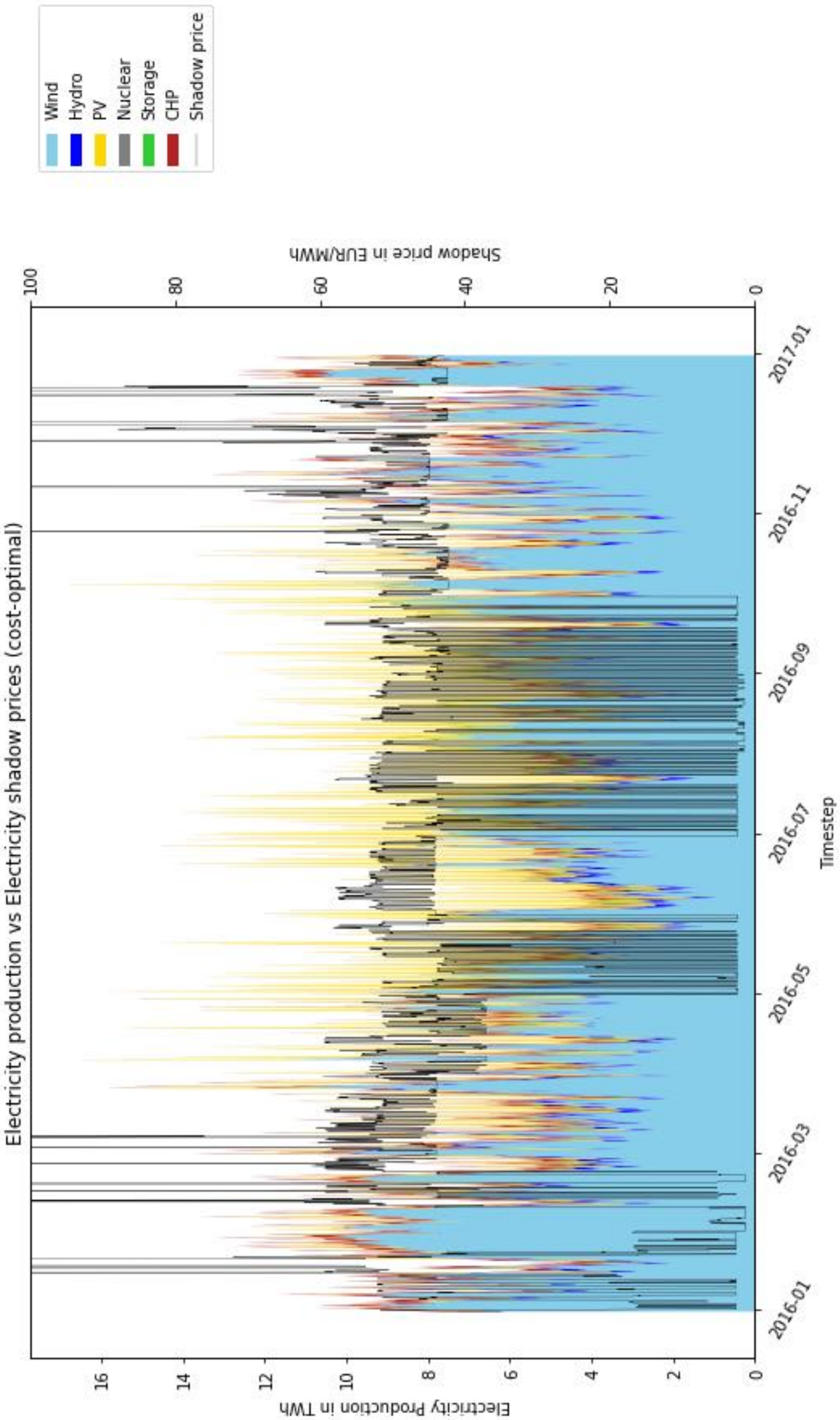
G.2 Setting fuel demand increase to 50%

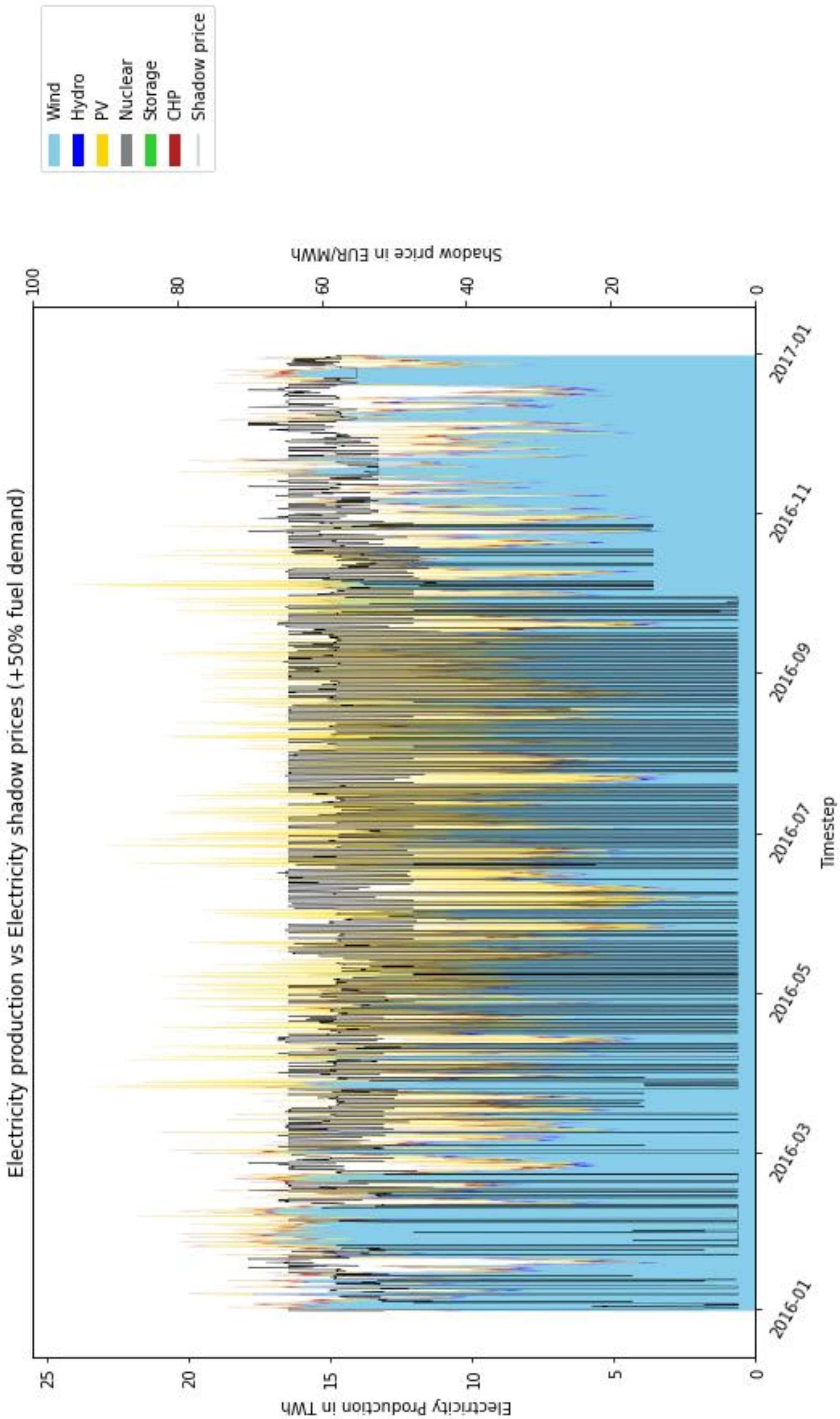
```
overrides:
    demand_share_fuel_50:
        group_constraints:
            demand_share_fuel_cooking:
                techs: [gas_hob]
                demand_share_equals.cooking: 0.5
            demand_share_fuel_heat:
                techs: [methane_tech_heat_to_demand,
chp_methane_extraction_tech_heat_to_demand,
chp_methane_back_pressure_simple_tech_heat_to_demand,
chp_methane_back_pressure_combined_tech_heat_to_demand]
                demand_share_equals.heat: 0.5
            demand_share_fuel_transport_light:
                techs: [light_transport_ice]
                demand_share_equals.light_transport: 0.5
            demand_share_fuel_transport_heavy:
                techs: [heavy_transport_ice]
                demand_share_equals.heavy_transport: 0.5
            demand_share_fuel_electricity:
                techs: [ccgt]
                      carrier_prod_share_max.electricity: 0.5
```

Appendix H – Average capacity factors wind offshore and onshore

Country	Onshore	Offshore
Belgium	0.314	0.445
Germany	0.257	0.499
Denmark	0.442	0.500
France	0.281	0.450
Great Britain	0.437	0.569
Ireland	0.451	0.602
Luxembourg	0.308	0
The Netherlands	0.363	0.528
Norway	0.299	0.512
Sweden	0.302	0.417
Average	0.345	0.452

Appendix I – Electricity production vs electricity shadow price





Electricity production vs Electricity shadow prices (+50% fuel demand)
Electricity Production in TWh
Shadow price in EUR/MWh
Timestep
2016-01
2016-03
2016-05
2016-07
2016-09
2016-11
2017-01
0
5
10
15
20
25
0
20
40
60
80
100
Wind
Hydro
PV
Nuclear
Storage
CHP
Shadow price

Appendix S1 – Python scripts for the analysis of the North Sea Calliope 2020 model

Loading results from the supercluster

```
opt_horizon = 2020
selected_scenario = '2020-1h'
path_to_netcdf_of_results = 'results/supercluster/north-
sea_{}.nc'.format(selected_scenario)
# path_to_duals = 'results\Supercluster archive 27 jun/north-
sea_duals_{}.pickle'.format(selected_scenario)
model = calliope.read_netcdf('results/supercluster/north-
sea_{}.nc'.format(selected_scenario))
# duals = load_duals(path_to_duals)
balance_duals=pd.read_csv('results//supercluster/north_sea_balance_duals_{}
.csv'.format(selected scenario))
```

Analysing TES for the Netherlands

```
#%%Production supply in the Netherlands
supply_techs = list(model_2020-
1h._model_data.techs_supply.to_pandas().index) + list(model_2020-
1h._model_data.techs_supply_plus.to_pandas().index)
NLD_carrier_prod_supply_only = model_2020-
1h.get_formatted_array('carrier_prod').loc[{'techs':supply_techs}].sum(['te
chs','timesteps']).loc[{'locs':'NLD'}].to_pandas()
NLD_carrier_prod_supply_only
=NLD_carrier_prod_supply_only.loc[(NLD_carrier_prod_supply_only != 0)]
#remove rows that contain only 0 values
NLD_carrier_prod_supply_only=(1/10)*NLD_carrier_prod_supply_only #to
convert 100.000MW to TWh, multiply by 0.1

source = ["Coal","Natural gas","Electricity","Biofuels and waste","Oil"]
# total_energy_supply = [172876,1316248,44589,167,93186,210646,1068129] #In
TJ
total_energy_supply = [172876,1316248,137942,210646,1068129] #In TJ
total_energy_supply = np.multiply(total_energy_supply,1/3600) #Convert TJ
to TWh
model_total_energy_supply = [87.84,228.26,35.3076,1.69,309.46]
w=0.4
bar1 = np.arange(len(source))
bar2 = [i+w for i in bar1]

plt.bar(bar1,total_energy_supply,w,label="IEA data")
plt.bar(bar2,model_total_energy_supply,w,label="Model data")
plt.ylabel("Energy supply in TWh")
plt.xlabel("source")
plt.title("TES by source in the Netherlands 2020")
plt.xticks(bar1+w/2,source,rotation=90)
plt.legend()
```

Analyzing electricity duals

```
shadowprice_elec_nld_2020_1h =
balance_duals_2020_1h[(balance_duals_2020_1h.region == 'NLD') &
(balance_duals_2020_1h.carrier == 'electricity')]
shadowprice_elec_nld_2020_1h['timestep'] =
pd.to_datetime(shadowprice_elec_nld_2020_1h['timestep'])

shadowprice_elec_nld_2020_op =
balance_duals_2020_op[(balance_duals_2020_op.region == 'NLD') &
(balance_duals_2020_op.carrier == 'electricity')]
shadowprice_elec_nld_2020_op['timestep'] =
pd.to_datetime(shadowprice_elec_nld_2020_op['timestep'])

plt.figure(figsize=(10,5))
plt.plot(shadowprice_elec_nld_2020_1h['timestep'],
10000*shadowprice_elec_nld_2020_1h['dual-value'], label=('NLD electricity
duals 2020 model'))
plt.plot(shadowprice_elec_nld_2020_1h['timestep'],day_ahead_price_2015['Day
-ahead Price [EUR/MWh]'],label=('NLD ENTSO-E day ahead price 2015'))
plt.xlim(shadowprice_elec_nld_2020_1h['timestep'].iloc[1416],shadowprice_el
ec_nld_2020_1h['timestep'].iloc[2159] ) #filter for the March
plt.title('Electricity duals 2020 North Sea Calliope model')
plt.ylabel('Electricity price in EUR/MWh')
plt.legend()
plt.ylim(ymin=0,ymax=100)


plt.figure(figsize=(10,5))
plt.plot(shadowprice_elec_nld_2020_op['timestep'],
10000*shadowprice_elec_nld_2020_op['dual-value'], label=('NLD electricity
duals 2020 model'))
plt.plot(shadowprice_elec_nld_2020_op['timestep'],day_ahead_price_2015['Day
-ahead Price [EUR/MWh]'],label=('NLD ENTSO-E day ahead price 2015'))
plt.xlim(shadowprice_elec_nld_2020_op['timestep'].iloc[1416],shadowprice_el
ec_nld_2020_op['timestep'].iloc[2159] ) #filter for the March
plt.title('Electricity duals 2020 North Sea Calliope power sector only
model')
plt.ylabel('Electricity price in EUR/MWh')
plt.legend()
plt.ylim(ymin=0,ymax=100)
```

Appendix S2 – Python scripts for the analysis of the North Sea Calliope 2050 model

Loading 2050 models

```
#%% Loading 2050 models
opt_horizon = 2050
selected_scenario = list([2050,'2050-80hydrogen-30fuel','2050-100hydrogen-30fuel','2050-80hydrogen-50fuel'])
weather_yr = [2010,2011,2012,2013,2014,2015,2016,2017,2018]
#optimal scenario
path_to_netcdf_of_results_2010 = 'results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[0],weather_yr[0])
model_2050_2010 = calliope.read_netcdf('results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[0],weather_yr[0]))
balance_duals_2050_2010=pd.read_csv('results//supercluster/2050/north_sea_balance_duals_{}_{}.csv'.format(selected_scenario[0],weather_yr[0]))

path_to_netcdf_of_results_2015 = 'results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[0],weather_yr[5])
model_2050_2015 = calliope.read_netcdf('results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[0],weather_yr[5]))
balance_duals_2050_2015=pd.read_csv('results//supercluster/2050/north_sea_balance_duals_{}_{}.csv'.format(selected_scenario[0],weather_yr[5]))

path_to_netcdf_of_results_2016 = 'results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[0],weather_yr[6])
model_2050_2016 = calliope.read_netcdf('results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[0],weather_yr[6]))
balance_duals_2050_2016=pd.read_csv('results//supercluster/2050/north_sea_balance_duals_{}_{}.csv'.format(selected_scenario[0],weather_yr[6]))

#80% hydrogen share and 30% fuel increase
path_to_netcdf_of_results_2010_8030 = 'results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[1],weather_yr[0])
model_2050_2010_8030 =
calliope.read_netcdf('results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[1],weather_yr[0]))
balance_duals_2050_2010_8030=pd.read_csv('results//supercluster/2050/north_sea_balance_duals_{}_{}.csv'.format(selected_scenario[1],weather_yr[0]))

path_to_netcdf_of_results_2015_8030 = 'results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[1],weather_yr[5])
model_2050_2015_8030 =
calliope.read_netcdf('results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[1],weather_yr[5]))
balance_duals_2050_2015_8030=pd.read_csv('results//supercluster/2050/north_sea_balance_duals_{}_{}.csv'.format(selected_scenario[1],weather_yr[5]))

path_to_netcdf_of_results_2016_8030 = 'results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[1],weather_yr[6])
model_2050_2016_8030 =
calliope.read_netcdf('results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[1],weather_yr[6]))
balance_duals_2050_2016_8030=pd.read_csv('results//supercluster/2050/north_sea_balance_duals_{}_{}.csv'.format(selected_scenario[1],weather_yr[6]))
```

```
#100% hydrogen share and 30% fuel increase
path_to_netcdf_of_results_2010_10030 = 'results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[2],weather_yr[0])
model_2050_2010_10030 =
calliope.read_netcdf('results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[2],weather_yr[0]))
balance_duals_2050_2010_10030=pd.read_csv('results//supercluster/2050/north_sea_balance_duals_{}_{}.csv'.format(selected_scenario[2],weather_yr[0]))

path_to_netcdf_of_results_2015_10030 = 'results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[2],weather_yr[5])
model_2050_2015_10030 =
calliope.read_netcdf('results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[2],weather_yr[5]))
balance_duals_2050_2015_10030=pd.read_csv('results//supercluster/2050/north_sea_balance_duals_{}_{}.csv'.format(selected_scenario[2],weather_yr[5]))

path_to_netcdf_of_results_2016_10030 = 'results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[2],weather_yr[6])
model_2050_2016_10030 =
calliope.read_netcdf('results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[2],weather_yr[6]))
balance_duals_2050_2016_10030=pd.read_csv('results//supercluster/2050/north_sea_balance_duals_{}_{}.csv'.format(selected_scenario[2],weather_yr[6]))

#80% hydrogen share and 50% fuel increase
path_to_netcdf_of_results_2010_8050 = 'results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[3],weather_yr[0])
model_2050_2010_8050 =
calliope.read_netcdf('results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[3],weather_yr[0]))
balance_duals_2050_2010_8050=pd.read_csv('results//supercluster/2050/north_sea_balance_duals_{}_{}.csv'.format(selected_scenario[3],weather_yr[0]))

path_to_netcdf_of_results_2015_8050 = 'results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[3],weather_yr[5])
model_2050_2015_8050 =
calliope.read_netcdf('results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[3],weather_yr[5]))
balance_duals_2050_2015_8050=pd.read_csv('results//supercluster/2050/north_sea_balance_duals_{}_{}.csv'.format(selected_scenario[3],weather_yr[5]))

path_to_netcdf_of_results_2016_8050 = 'results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[3],weather_yr[6])
model_2050_2016_8050 =
calliope.read_netcdf('results/supercluster/2050/north-sea_{}_{}.nc'.format(selected_scenario[3],weather_yr[6]))
balance_duals_2050_2016_8050=pd.read_csv('results//supercluster/2050/north_sea_balance_duals_{}_{}.csv'.format(selected_scenario[3],weather_yr[6]))
```

Setting up electricity duals data

```
#optimal scenario
shadowprice_elec_nld_2010 =
balance_duals_2050_2010[(balance_duals_2050_2010.region == 'NLD') &
(balance_duals_2050_2010.carrier == 'electricity')]
shadowprice_elec_nld_2015 =
balance_duals_2050_2015[(balance_duals_2050_2015.region == 'NLD') &
(balance_duals_2050_2015.carrier == 'electricity')]
shadowprice_elec_nld_2016 =
balance_duals_2050_2016[(balance_duals_2050_2016.region == 'NLD') &
(balance_duals_2050_2016.carrier == 'electricity')]

shadowprice_elec_nld_2010['timestep'] =
pd.to_datetime(shadowprice_elec_nld_2010['timestep'])
shadowprice_elec_nld_2015['timestep'] =
pd.to_datetime(shadowprice_elec_nld_2015['timestep'])
shadowprice_elec_nld_2016['timestep'] =
pd.to_datetime(shadowprice_elec_nld_2016['timestep'])
shadowprice_elec_nld_2016.drop(shadowprice_elec_nld_2016.index[1416:1440],
inplace=True)

#80% hydrogen share and 30% fuel increase
shadowprice_elec_nld_2010_8030 =
balance_duals_2050_2010_8030[(balance_duals_2050_2010_8030.region == 'NLD')
& (balance_duals_2050_2010_8030.carrier == 'electricity')]
shadowprice_elec_nld_2015_8030 =
balance_duals_2050_2015_8030[(balance_duals_2050_2015_8030.region == 'NLD')
& (balance_duals_2050_2015_8030.carrier == 'electricity')]
shadowprice_elec_nld_2016_8030 =
balance_duals_2050_2016_8030[(balance_duals_2050_2016_8030.region == 'NLD')
& (balance_duals_2050_2016_8030.carrier == 'electricity')]

shadowprice_elec_nld_2010_8030['timestep'] =
pd.to_datetime(shadowprice_elec_nld_2010_8030['timestep'])
shadowprice_elec_nld_2015_8030['timestep'] =
pd.to_datetime(shadowprice_elec_nld_2015_8030['timestep'])
shadowprice_elec_nld_2016_8030['timestep'] =
pd.to_datetime(shadowprice_elec_nld_2016_8030['timestep'])
shadowprice_elec_nld_2016_8030.drop(shadowprice_elec_nld_2016_8030.index[14
16:1440], inplace=True)

#100% hydrogen share and 30% fuel increase
shadowprice_elec_nld_2010_10030 =
balance_duals_2050_2010_10030[(balance_duals_2050_2010_10030.region ==
'NLD') & (balance_duals_2050_2010_10030.carrier == 'electricity')]
shadowprice_elec_nld_2015_10030 =
balance_duals_2050_2015_10030[(balance_duals_2050_2015_10030.region ==
'NLD') & (balance_duals_2050_2015_10030.carrier == 'electricity')]
shadowprice_elec_nld_2016_10030 =
balance_duals_2050_2016_10030[(balance_duals_2050_2016_10030.region ==
'NLD') & (balance_duals_2050_2016_10030.carrier == 'electricity')]
```

```
shadowprice_elec_nld_2010_10030['timestep'] =
pd.to_datetime(shadowprice_elec_nld_2010_10030['timestep'])
shadowprice_elec_nld_2015_10030['timestep'] =
pd.to_datetime(shadowprice_elec_nld_2010_10030['timestep'])
shadowprice_elec_nld_2016_10030['timestep'] =
pd.to_datetime(shadowprice_elec_nld_2010_10030['timestep'])
shadowprice_elec_nld_2016_10030.drop(shadowprice_elec_nld_2016_10030.index[
1416:1440], inplace=True)

#80% hydrogen share and 50% fuel increase
shadowprice_elec_nld_2010_8050 =
balance_duals_2050_2010_8050[(balance_duals_2050_2010_8050.region == 'NLD')
& (balance_duals_2050_2010_8050.carrier == 'electricity')]
shadowprice_elec_nld_2015_8050 =
balance_duals_2050_2015_8050[(balance_duals_2050_2015_8050.region == 'NLD')
& (balance_duals_2050_2015_8050.carrier == 'electricity')]
shadowprice_elec_nld_2016_8050 =
balance_duals_2050_2016_8050[(balance_duals_2050_2016_8050.region == 'NLD')
& (balance_duals_2050_2016_8050.carrier == 'electricity')]

shadowprice_elec_nld_2010_8050['timestep'] =
pd.to_datetime(shadowprice_elec_nld_2010_8050['timestep'])
shadowprice_elec_nld_2015_8050['timestep'] =
pd.to_datetime(shadowprice_elec_nld_2015_8050['timestep'])
shadowprice_elec_nld_2016_8050['timestep'] =
pd.to_datetime(shadowprice_elec_nld_2016_8050['timestep'])
shadowprice_elec_nld_2016_8050.drop(shadowprice_elec_nld_2016_8050.index[14
16:1440], inplace=True)
```

TES by source in the Netherlands 2050

```
source = ["Biofuel","Electricity","Synthetic diesel","Synthetic
kerosene","Synthetic methane","Synthetic_methanol","Waste"]
TES_2010 =
[NL_supply_tech_2010['biofuel'],NL_supply_tech_2010['electricity'],NL_suppl
y_tech_2010['syn_diesel'],NL_supply_tech_2010['syn_kerosene'],NL_supply_tec
h_2010['syn_methane'],NL_supply_tech_2010['syn_methanol'],NL_supply_tech_20
10['waste']]
TES_2015 =
[NL_supply_tech_2015['biofuel'],NL_supply_tech_2015['electricity'],NL_suppl
y_tech_2015['syn_diesel'],NL_supply_tech_2015['syn_kerosene'],NL_supply_tec
h_2015['syn_methane'],NL_supply_tech_2015['syn_methanol'],NL_supply_tech_20
15['waste']]
TES_2016 =
[NL_supply_tech_2016['biofuel'],NL_supply_tech_2016['electricity'],NL_suppl
y_tech_2016['syn_diesel'],NL_supply_tech_2016['syn_kerosene'],NL_supply_tec
h_2016['syn_methane'],NL_supply_tech_2016['syn_methanol'],NL_supply_tech_20
16['waste']]
w=0.25
bar1 = np.arange(len(source))
bar2 = [i+w for i in bar1]
bar3 = [i+w for i in bar2]

plt.bar(bar1,TES_2010,w,label="TES in bad weather",color='salmon')
plt.bar(bar2,TES_2016,w,label="TES in normal weather",color='bisque')
plt.bar(bar3,TES_2015,w,label="TES in good weather",color='lightgreen')

plt.ylabel("Energy supply in TWh")
plt.xlabel("source")
plt.title("TES by source in the Netherlands 2050 (optimal hydrogen share)")
plt.xticks(bar1+w,source,rotation=315)
plt.legend(loc='upper center')
# plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left')

#%% comparing model results with infrastructure outlook 2050, tennet
source = ["Electricity","Methane","Hydrogen","Others","Liquid fuels"]
total_energy_supply_2010 =
[NL_supply_tech_2010['electricity']/sum(TES_2010),NL_supply_tech_2010['syn_
methane']/sum(TES_2010),NL_supply_tech_2010['syn_methanol']/sum(TES_2010),(
NL_supply_tech_2010['waste']+NL_supply_tech_2010['biofuel'])/sum(TES_2010),
(NL_supply_tech_2010['syn_diesel']+NL_supply_tech_2010['syn_kerosene'])/sum
(TES_2010)] #In TJ
total_energy_supply_2010 = [x*100 for x in total_energy_supply_2010]
IO_2050_data = [26,23,24,15,12]
w=0.4
bar1 = np.arange(len(source))
bar2 = [i+w for i in bar1]

plt.bar(bar1,IO_2050_data,w,label="Infrastructure Outlook 2050 data")
plt.bar(bar2,total_energy_supply_2010,w,label="Model data")
plt.ylabel("Energy supply in percentage")
plt.xlabel("source")
plt.title("TES by source in the Netherlands 2050 (optimal hydrogen share)")
plt.xticks(bar1+w/2,source,rotation=90)
plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left')
```

```
TES_2010_8030 =
[NL_supply_tech_2010_8030['biofuel'],NL_supply_tech_2010_8030['electricity'
],NL_supply_tech_2010_8030['syn_diesel'],NL_supply_tech_2010_8030['syn_kero
sene'],NL_supply_tech_2010_8030['syn_methane'],NL_supply_tech_2010_8030['sy
n_methanol'],NL_supply_tech_2010_8030['waste']]
TES_2015_8030 =
[NL_supply_tech_2015_8030['biofuel'],NL_supply_tech_2015_8030['electricity'
],NL_supply_tech_2015_8030['syn_diesel'],NL_supply_tech_2015_8030['syn_kero
sene'],NL_supply_tech_2015_8030['syn_methane'],NL_supply_tech_2015_8030['sy
n_methanol'],NL_supply_tech_2015_8030['waste']]
TES_2016_8030 =
[NL_supply_tech_2016_8030['biofuel'],NL_supply_tech_2016_8030['electricity'
],NL_supply_tech_2016_8030['syn_diesel'],NL_supply_tech_2016_8030['syn_kero
sene'],NL_supply_tech_2016_8030['syn_methane'],NL_supply_tech_2016_8030['sy
n_methanol'],NL_supply_tech_2016_8030['waste']]
w=0.25
bar1 = np.arange(len(source))
bar2 = [i+w for i in bar1]
bar3 = [i+w for i in bar2]
plt.bar(bar1,TES_2010_8030,w,label="TES in bad weather",color='salmon')
plt.bar(bar2,TES_2016_8030,w,label="TES in normal weather",color='bisque')
plt.bar(bar3,TES_2015_8030,w,label="TES in good
weather",color='lightgreen')
plt.ylabel("Energy supply in TWh")
plt.xlabel("source")
plt.title("TES by source in the Netherlands 2050 (+30% fuel demand)")
plt.xticks(bar1+w,source,rotation=315)
# plt.legend()

TES_2010_8050 =
[NL_supply_tech_2010_8050['biofuel'],NL_supply_tech_2010_8050['electricity'
],NL_supply_tech_2010_8050['syn_diesel'],NL_supply_tech_2010_8050['syn_kero
sene'],NL_supply_tech_2010_8050['syn_methane'],NL_supply_tech_2010_8050['sy
n_methanol'],NL_supply_tech_2010_8050['waste']]
TES_2015_8050 =
[NL_supply_tech_2015_8050['biofuel'],NL_supply_tech_2015_8050['electricity'
],NL_supply_tech_2015_8050['syn_diesel'],NL_supply_tech_2015_8050['syn_kero
sene'],NL_supply_tech_2015_8050['syn_methane'],NL_supply_tech_2015_8050['sy
n_methanol'],NL_supply_tech_2015_8050['waste']]
TES_2016_8050 =
[NL_supply_tech_2016_8050['biofuel'],NL_supply_tech_2016_8050['electricity'
],NL_supply_tech_2016_8050['syn_diesel'],NL_supply_tech_2016_8050['syn_kero
sene'],NL_supply_tech_2016_8050['syn_methane'],NL_supply_tech_2016_8050['sy
n_methanol'],NL_supply_tech_2016_8050['waste']]
w=0.25
bar1 = np.arange(len(source))
bar2 = [i+w for i in bar1]
bar3 = [i+w for i in bar2]

plt.bar(bar1,TES_2010_8050,w,label="TES in bad weather",color='salmon')
plt.bar(bar2,TES_2016_8050,w,label="TES in normal weather",color='bisque')
plt.bar(bar3,TES_2015_8050,w,label="TES in good
weather",color='lightgreen')

plt.ylabel("Energy supply in TWh")
plt.xlabel("source")
plt.title("TES by source in the Netherlands 2050 (+50% fuel demand)")
plt.xticks(bar1+w,source,rotation=315)
plt.legend()
```

Electricity demand

```
#%% electricity consumption comparison
elec_demand_2016 =
model_2050_2016.get_formatted_array('carrier_con').loc['electricity'].loc['
NLD'].loc[{'techs':demand_techs}].sum(['timesteps']).to_pandas()
elec_demand_2016_8050 =
model_2050_2016_8050.get_formatted_array('carrier_con').loc['electricity'].
loc['NLD'].loc[{'techs':demand_techs}].sum(['timesteps']).to_pandas()
elec_demand_2016 = elec_demand_2016*-0.1
elec_demand_2016_8050 = elec_demand_2016_8050*-0.1

demand_techs =
['battery','demand_elec','electric_heater','electric_hob','electrolysis','h
eavy_transport_ev','hp','hydrogen_to_liquids','light_transport_ev']
data_opt =
[elec_demand_2016[0],elec_demand_2016[1],elec_demand_2016[2],elec_demand_20
16[3],elec_demand_2016[4],elec_demand_2016[5],elec_demand_2016[6],elec_dema
nd_2016[7],elec_demand_2016[8]]
data_8050 =
[elec_demand_2016_8050[0],elec_demand_2016_8050[1],elec_demand_2016_8050[2]
,elec_demand_2016_8050[3],elec_demand_2016_8050[4],elec_demand_2016_8050[5]
,elec_demand_2016_8050[6],elec_demand_2016_8050[7],elec_demand_2016_8050[8]
]
w=0.25
bar1 = np.arange(len(demand_techs))
bar2 = [i+w for i in bar1]

plt.bar(bar1,data_opt,w,label="electricity demand optimal hydrogen
share",color='lightgray')
plt.bar(bar2,data_8050,w,label="electricity demand +50% fuel
demand",color='blue')

plt.ylabel("Electricity demand in TWh")
plt.xlabel("source")
plt.title("Electricity demand by source in the Netherlands (normal
weather)")
plt.xticks(bar1+w,demand_techs,rotation=90)
plt.yscale("symlog")
plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left')
```

Total electricity production by source

```
electricity_techs =
['wind_offshore','wind_onshore_monopoly','wind_onshore_competing','hydro_ru
n_of_river','hydro_reservoir','open_field_pv','roof_mounted_pv','nuclear','
battery','pumped_hydro','chp_biofuel_extraction','chp_wte_back_pressure']


elec_production_source_2010 =
model_2050_2010.get_formatted_array('carrier_prod').loc[{'techs':electricit
y_techs}].loc[{'carriers':'electricity'}].sum(['timesteps','locs']).to_pand
as()
elec_production_source_2015 =
model_2050_2015.get_formatted_array('carrier_prod').loc[{'techs':electricit
y_techs}].loc[{'carriers':'electricity'}].sum(['timesteps','locs']).to_pand
as()
elec_production_source_2016 =
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':electricit
y_techs}].loc[{'carriers':'electricity'}].sum(['timesteps','locs']).to_pand
as()

elec_production_source_2010 = elec_production_source_2010*0.1
elec_production_source_2015 = elec_production_source_2015*0.1
elec_production_source_2016 = elec_production_source_2016*0.1

sources = elec_production_source_2010.index
w=0.25
bar1 = np.arange(len(sources))
bar2 = [i+w for i in bar1]
bar3 = [i+w for i in bar2]


plt.bar(bar1,elec_production_source_2010,w,label="bad
weather",color='salmon')
plt.bar(bar2,elec_production_source_2016,w,label="normal
weather",color='bisque')
plt.bar(bar3,elec_production_source_2015,w,label="good
weather",color='lightgreen')
plt.legend()

plt.xticks(bar1+w,sources,rotation=90)
plt.title("Total electricity production by source North Sea region")
plt.ylabel('Electricity production in TWh')
plt.xlabel('source')
```

```
#%% electricity production by source for different hydrogen configurations
elec_production_source_2010_8030 =
model_2050_2010_8030.get_formatted_array('carrier_prod').loc[{'techs':elect
ricity_techs}].loc[{'carriers':'electricity'}].sum(['timesteps','locs']).to
_pandas()
elec_production_source_2015_8030 =
model_2050_2015_8030.get_formatted_array('carrier_prod').loc[{'techs':elect
ricity_techs}].loc[{'carriers':'electricity'}].sum(['timesteps','locs']).to
_pandas()
elec_production_source_2016_8030 =
model_2050_2016_8030.get_formatted_array('carrier_prod').loc[{'techs':elect
ricity_techs}].loc[{'carriers':'electricity'}].sum(['timesteps','locs']).to
_pandas()

elec_production_source_2010_8030 = elec_production_source_2010_8030*0.1
elec_production_source_2015_8030 = elec_production_source_2015_8030*0.1
elec_production_source_2016_8030 = elec_production_source_2016_8030*0.1

elec_production_source_2010_8050 =
model_2050_2010_8050.get_formatted_array('carrier_prod').loc[{'techs':elect
ricity_techs}].loc[{'carriers':'electricity'}].sum(['timesteps','locs']).to
_pandas()
elec_production_source_2015_8050 =
model_2050_2015_8050.get_formatted_array('carrier_prod').loc[{'techs':elect
ricity_techs}].loc[{'carriers':'electricity'}].sum(['timesteps','locs']).to
_pandas()
elec_production_source_2016_8050 =
model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':elect
ricity_techs}].loc[{'carriers':'electricity'}].sum(['timesteps','locs']).to
_pandas()

elec_production_source_2010_8050 = elec_production_source_2010_8050*0.1
elec_production_source_2015_8050 = elec_production_source_2015_8050*0.1
elec_production_source_2016_8050 = elec_production_source_2016_8050*0.1

sources = elec_production_source_2010.index
w=0.25
bar1 = np.arange(len(sources))
bar2 = [i+w for i in bar1]
bar3 = [i+w for i in bar2]


plt.bar(bar1,elec_production_source_2010_8030,w,label="bad
weather",color='salmon')
plt.bar(bar2,elec_production_source_2016_8030,w,label="normal
weather",color='bisque')
plt.bar(bar3,elec_production_source_2015_8030,w,label="good
weather",color='lightgreen')
plt.legend()

plt.xticks(bar1+w,sources,rotation=90)
plt.title("Total electricity production by source North Sea region (+30%
fuel demand)")
plt.ylabel('Electricity production in TWh')
plt.xlabel('source')
```

```
#%% electricity production by source NLD
elec_production_source_2010_NLD =
model_2050_2010.get_formatted_array('carrier_prod').loc[{'techs':electricit
y_techs}].loc[{'carriers':'electricity'}].loc[{'locs':'NLD'}].sum(['timeste
ps']).to_pandas()
elec_production_source_2015_NLD =
model_2050_2015.get_formatted_array('carrier_prod').loc[{'techs':electricit
y_techs}].loc[{'carriers':'electricity'}].loc[{'locs':'NLD'}].sum(['timeste
ps']).to_pandas()
elec_production_source_2016_NLD =
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':electricit
y_techs}].loc[{'carriers':'electricity'}].loc[{'locs':'NLD'}].sum(['timeste
ps']).to_pandas()

elec_production_source_2010_NLD = elec_production_source_2010*0.1
elec_production_source_2015_NLD = elec_production_source_2015*0.1
elec_production_source_2016_NLD = elec_production_source_2016*0.1

sources = elec_production_source_2010.index
w=0.25
bar1 = np.arange(len(sources))
bar2 = [i+w for i in bar1]
bar3 = [i+w for i in bar2]


plt.bar(bar1,elec_production_source_2010_NLD,w,label="bad
weather",color='salmon')
plt.bar(bar2,elec_production_source_2016_NLD,w,label="normal
weather",color='bisque')
plt.bar(bar3,elec_production_source_2015_NLD,w,label="good
weather",color='lightgreen')
plt.legend()

plt.xticks(bar1+w,sources,rotation=90)
plt.title("Total electricity production by source in the Netherlands")
plt.ylabel('Electricity production)
plt.xlabel('source')
```

Total energy system cost

```
weather_type = ["Bad weather","Normal weather","Good weather"]
obj_opt = [4.79675751e+02,4.35732732e+02,4.18938085e+02]
obj_8030 = [6.42844135e+02,5.88498512e+02,5.54155026e+02]
obj_8050 = [7.41755933e+02,6.94771637e+02,6.49037273e+02]
w=0.25
bar1 = np.arange(len(weather_type))
bar2 = [i+w for i in bar1]
bar3 = [i+w for i in bar2]

plt.bar(bar1,obj_opt,w,label="Optimal hydrogen scenario",color='palegreen')
plt.bar(bar2,obj_8030,w,label="+30% fuel demand",color='lightgreen')
plt.bar(bar3,obj_8050,w,label="+50% fuel demand",color='limegreen')

plt.ylabel("x billion euros")
plt.xlabel("Weather type")
plt.title("Total energy system cost for different hydrogen configurations")
plt.xticks(bar1+w,weather_type)
plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left')
```

Levelized cost of energy

```
lcoe_2010 =
1e4*model_2050_2010.results.systemwide_levelised_cost.loc[{'carriers':
'electricity', 'costs':'monetary'}].to_pandas()
lcoe_2015 =
1e4*model_2050_2015.results.systemwide_levelised_cost.loc[{'carriers':
'electricity', 'costs':'monetary'}].to_pandas()
lcoe_2016 =
1e4*model_2050_2016.results.systemwide_levelised_cost.loc[{'carriers':
'electricity', 'costs':'monetary'}].to_pandas()

lcoe_2010.replace(np.inf, np.nan, inplace=True)
lcoe_2015.replace(np.inf, np.nan, inplace=True)
lcoe_2016.replace(np.inf, np.nan, inplace=True)
lcoe_2010 = lcoe_2010.dropna()
lcoe_2015 = lcoe_2015.dropna()
lcoe_2016 = lcoe_2016.dropna()

techs = lcoe_2010.index
w=0.25
bar1 = np.arange(len(techs))
bar2 = [i+w for i in bar1]
bar3 = [i+w for i in bar2]


plt.bar(bar1,lcoe_2010,w,label="bad weather",color='salmon')
plt.bar(bar2,lcoe_2015,w,label="normal weather",color='bisque')
plt.bar(bar3,lcoe_2016,w,label="good weather",color='lightgreen')
plt.legend()

plt.xticks(bar1+w,techs,rotation=90)
plt.title("Systemwide LCOE of technologies")
plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left')
```

```
#%%ANALYZING LEVELIZED COST FOR DIFFERENT HYDROGEN SHARES
electricity_techs =
['wind_offshore','wind_onshore_monopoly','wind_onshore_competing','hydro_ru
n_of_river','hydro_reservoir','open_field_pv','roof_mounted_pv','nuclear','
battery','pumped_hydro','chp_biofuel_extraction','chp_wte_back_pressure']

lcoe_2016 =
1e4*model_2050_2016.results.systemwide_levelised_cost.loc[{'carriers':
'electricity', 'costs':'monetary','techs':electricity_techs}].to_pandas()
lcoe_2016_8030 =
1e4*model_2050_2016_8030.results.systemwide_levelised_cost.loc[{'carriers':
'electricity', 'costs':'monetary','techs':electricity_techs}].to_pandas()
lcoe_2016_8050 =
1e4*model_2050_2016_8050.results.systemwide_levelised_cost.loc[{'carriers':
'electricity', 'costs':'monetary','techs':electricity_techs}].to_pandas()

lcoe_2016.replace(np.inf, np.nan, inplace=True)
lcoe_2016_8030.replace(np.inf, np.nan, inplace=True)
lcoe_2016_8050.replace(np.inf, np.nan, inplace=True)
lcoe_2016 = lcoe_2016.dropna()
lcoe_2016_8030 = lcoe_2016_8030.dropna()
lcoe_2016_8050 = lcoe_2016_8050.dropna()

techs = lcoe_2016.index

w=0.25
bar1 = np.arange(len(techs))
bar2 = [i+w for i in bar1]
bar3 = [i+w for i in bar2]

plt.bar(bar1,lcoe_2016,w,label="Optimal scenario",color='palegreen')
plt.bar(bar2,lcoe_2016_8030,w,label="+30% fuel demand",color='lightgreen')
plt.bar(bar3,lcoe_2016_8050,w,label="+50% fuel demand",color='limegreen')

plt.ylabel("LCOE in €/MWh")
plt.xlabel("Technology")
plt.title("Systemwide LCOE of technologies for different hydrogen shares
(normal weather)")
plt.xticks(bar1+w,techs,rotation=90)
plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left')
plt.yscale("log")
```

```
lcoe_2010 =
1e4*model_2050_2010.results.systemwide_levelised_cost.loc[{'carriers':
'electricity', 'costs':'monetary'}].to_pandas()
lcoe_2010_8030 =
1e4*model_2050_2010_8030.results.systemwide_levelised_cost.loc[{'carriers':
'electricity', 'costs':'monetary'}].to_pandas()
lcoe_2010_8050 =
1e4*model_2050_2010_8050.results.systemwide_levelised_cost.loc[{'carriers':
'electricity', 'costs':'monetary'}].to_pandas()

lcoe_2010.replace(np.inf, np.nan, inplace=True)
lcoe_2010_8030.replace(np.inf, np.nan, inplace=True)
lcoe_2010_8050.replace(np.inf, np.nan, inplace=True)
lcoe_2010 = lcoe_2010.dropna()
lcoe_2010_8030 = lcoe_2010_8030.dropna()
lcoe_2010_8050 = lcoe_2010_8050.dropna()

techs = lcoe_2010.index

w=0.25
bar1 = np.arange(len(techs))
bar2 = [i+w for i in bar1]
bar3 = [i+w for i in bar2]

plt.bar(bar1,lcoe_2010,w,label="Cost-optimal scenario",color='palegreen')
plt.bar(bar2,lcoe_2010_8030,w,label="+30% fuel demand",color='lightgreen')
plt.bar(bar3,lcoe_2010_8050,w,label="+50% fuel demand",color='limegreen')


plt.ylabel("LCOE in €/MWh")
plt.xlabel("Weather type")
plt.title("Systemwide LCOE of technologies for different hydrogen shares
(bad weather)")
plt.xticks(bar1+w,techs,rotation=90)
plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left')
```

```
lcoe_2015.replace(np.inf, np.nan, inplace=True)
lcoe_2015_8030.replace(np.inf, np.nan, inplace=True)
lcoe_2015_8050.replace(np.inf, np.nan, inplace=True)
lcoe_2015 = lcoe_2015.dropna()
lcoe_2015_8030 = lcoe_2015_8030.dropna()
lcoe_2015_8050 = lcoe_2015_8050.dropna()

techs = lcoe_2015.index

w=0.25
bar1 = np.arange(len(techs))
bar2 = [i+w for i in bar1]
bar3 = [i+w for i in bar2]

plt.bar(bar1,lcoe_2015,w,label="Cost-optimal scenario",color='palegreen')
plt.bar(bar2,lcoe_2015_8030,w,label="+30% fuel demand",color='lightgreen')
plt.bar(bar3,lcoe_2015_8050,w,label="+50% fuel demand",color='limegreen')


plt.ylabel("LCOE in €/MWh")
plt.xlabel("Weather type")
plt.title("Systemwide LCOE of technologies for different hydrogen shares
(good weather)")
plt.xticks(bar1+w,techs,rotation=90)
plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left')
```

```
#%% lcoe electrolysers
lcoe_electrolyser_2010 =
1e4*model_2050_2010.results.systemwide_levelised_cost.loc[{'carriers':
'hydrogen','techs':'electrolysis','costs':'monetary'}].to_pandas()
lcoe_electrolyser_2010_8030 =
1e4*model_2050_2010_8030.results.systemwide_levelised_cost.loc[{'carriers':
'hydrogen','techs':'electrolysis','costs':'monetary'}].to_pandas()
lcoe_electrolyser_2010_8050 =
1e4*model_2050_2010_8050.results.systemwide_levelised_cost.loc[{'carriers':
'hydrogen','techs':'electrolysis','costs':'monetary'}].to_pandas()

lcoe_electrolyser_2016 =
1e4*model_2050_2016.results.systemwide_levelised_cost.loc[{'carriers':
'hydrogen','techs':'electrolysis','costs':'monetary'}].to_pandas()
lcoe_electrolyser_2016_8030 =
1e4*model_2050_2016_8030.results.systemwide_levelised_cost.loc[{'carriers':
'hydrogen','techs':'electrolysis','costs':'monetary'}].to_pandas()
lcoe_electrolyser_2016_8050 =
1e4*model_2050_2016_8050.results.systemwide_levelised_cost.loc[{'carriers':
'hydrogen','techs':'electrolysis','costs':'monetary'}].to_pandas()

lcoe_electrolyser_2015 =
1e4*model_2050_2015.results.systemwide_levelised_cost.loc[{'carriers':
'hydrogen','techs':'electrolysis','costs':'monetary'}].to_pandas()
lcoe_electrolyser_2015_8030 =
1e4*model_2050_2015_8030.results.systemwide_levelised_cost.loc[{'carriers':
'hydrogen','techs':'electrolysis','costs':'monetary'}].to_pandas()
lcoe_electrolyser_2015_8050 =
1e4*model_2050_2015_8050.results.systemwide_levelised_cost.loc[{'carriers':
'hydrogen','techs':'electrolysis','costs':'monetary'}].to_pandas()

weather_type = ["Bad weather","Normal weather","Good weather"]
lcoe_opt =
[lcoe_electrolyser_2010,lcoe_electrolyser_2016,lcoe_electrolyser_2015]
lcoe_8030 =
[lcoe_electrolyser_2010_8030,lcoe_electrolyser_2016_8030,lcoe_electrolyser_
2015_8030]
lcoe_8050 =
[lcoe_electrolyser_2010_8050,lcoe_electrolyser_2016_8050,lcoe_electrolyser_
2015_8050]
w=0.25
bar1 = np.arange(len(weather_type))
bar2 = [i+w for i in bar1]
bar3 = [i+w for i in bar2]

plt.bar(bar1,lcoe_opt,w,label="Optimal hydrogen
scenario",color='palegreen')
plt.bar(bar2,lcoe_8030,w,label="+30% fuel demand",color='lightgreen')
plt.bar(bar3,lcoe_8050,w,label="+50% fuel demand",color='limegreen')

plt.ylabel("LCOE in EUR/MWh")
plt.xlabel("Weather type")
plt.title("LCOE for electrolysis for different hydrogen configurations")
plt.xticks(bar1+w,weather_type)
plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left')
```

Price stability boxplots

```
bad_weather_data = pd.DataFrame({"Optimal scenario":
10000*shadowprice_elec_nld_2010['dual-value'],
                             "+30% fuel":
10000*shadowprice_elec_nld_2010_8030['dual-value'],
                             "+50% fuel":
10000*shadowprice_elec_nld_2010_8050['dual-value']
                             })


good_weather_data = pd.DataFrame({"Optimal scenario":
10000*shadowprice_elec_nld_2015['dual-value'],
                             "+30% fuel":
10000*shadowprice_elec_nld_2015_8030['dual-value'],
                             "+50% fuel":
10000*shadowprice_elec_nld_2015_8050['dual-value']
                             })


normal_weather_data = pd.DataFrame({"Optimal scenario":
10000*shadowprice_elec_nld_2016['dual-value'],
                             "+30% fuel":
10000*shadowprice_elec_nld_2016_8030['dual-value'],
                             "+50% fuel":
10000*shadowprice_elec_nld_2016_8050['dual-value'],
                             })

ax1 = bad_weather_data[['Optimal scenario','+30% fuel','+50%
fuel']].plot(kind='box', ylabel='Shadow price in
EUR/MWh',title='Electricity duals NLD bad weather ',showfliers=False,
showmeans=True, figsize=(3,6),widths=0.5)
plt.ylim(ymin=0,ymax=120)
plt.xticks(rotation=330)
ax2 = good_weather_data[['Optimal scenario','+30% fuel','+50%
fuel']].plot(kind='box', title='Electricity duals NLD good weather
',showfliers=False, showmeans=True, figsize=(3,6),widths=0.5)
plt.ylim(ymin=0,ymax=120)
plt.xticks(rotation=315)
ax3 = normal_weather_data[['Optimal scenario','+30% fuel','+50%
fuel']].plot(kind='box', title='Electricity duals NLD normal weather
',showfliers=False, showmeans=True, figsize=(3,6),widths=0.5)
plt.ylim(ymin=0,ymax=120)
plt.xticks(rotation=315)
```

Time series electricity duals vs shadow price

```
#Normal weather optimal
wind =
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':'wind_offs
hore'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas() +
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':'wind_onsh
ore_monopoly'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas() +
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':'wind_onsh
ore_competing'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()
hydro =
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':'hydro_run
_of_river'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas() +
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':'hydro_res
ervoir'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()
pv =
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':'open_fiel
d_pv'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas() +
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':'roof_moun
ted_pv'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()
nuclear =
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':'nuclear'}
].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()
# biofuel =
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':'biofuel_s
upply'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()
# waste =
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':'waste_sup
ply'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()
storage =
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':'battery'}
].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas() +
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':'pumped_hy
dro'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()
chp =
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':'chp_biofu
el_extraction'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()
+
model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':'chp_wte_b
ack_pressure'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()

tot_elec_production = wind+hydro+pv+nuclear+storage+chp
```

```
pal = ["skyblue", "blue", "gold", "grey","limegreen","firebrick"]
fig,ax = plt.subplots()
plt.title('Electricity production vs Electricity shadow prices (optimal
hydrogen share)')
data2 = plt.stackplot(tot_elec_production.index,
wind,hydro,pv,nuclear,storage,chp,
labels=['Wind','Hydro','PV','Nuclear','Storage','CHP'],colors=pal)
ax.set_xlabel("Timestep")
ax.set_ylabel("Electricity Production in TWh")

ax2=ax.twinx()
data3 =
plt.plot(shadowprice_elec_nld_2016['timestep'],10000*shadowprice_elec_nld_2
016['dual-value'],label=('Shadow price'),color='black',linewidth='0.8')
ax2.set_ylabel("Shadow price in EUR/MWh")
ax2.set_ylim(ymin=0,ymax=100)

data = data2+data3
labs = [l.get_label() for l in data]
ax.legend(data, labs, bbox_to_anchor=(1.12, 1.0), loc='upper left')
ax.tick_params(axis='x', rotation=330)
plt.show()
```

```
#%%Normal weather +50% fuel demand
wind_8050 =
model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':'wind
_offshore'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas() +
model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':'wind
_onshore_monopoly'}].sum(['locs']).loc[{'carriers':'electricity'}].to_panda
s() +
model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':'wind
_onshore_competing'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pand
as()
hydro_8050 =
model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':'hydr
o_run_of_river'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()
+
model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':'hydr
o_reservoir'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()
pv_8050 =
model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':'open
_field_pv'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas() +
model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':'roof
_mounted_pv'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()
nuclear_8050 =
model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':'nucl
ear'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()
# biofuel =
model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':'biof
uel_supply'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()
# waste =
model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':'wast
e_supply'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()
storage_8050 =
model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':'batt
ery'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas() +
model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':'pump
ed_hydro'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pandas()
chp_8050 =
model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':'chp_
biofuel_extraction'}].sum(['locs']).loc[{'carriers':'electricity'}].to_pand
as() +
model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':'chp_
wte_back_pressure'}].sum(['locs']).loc[{'carriers':'electricity'}].to_panda
s()

tot_elec_production_8050 =
wind_8050+hydro_8050+pv_8050+nuclear_8050+storage_8050+chp_8050
```

```
pal = ["skyblue", "blue", "gold", "grey","limegreen","firebrick"]
fig,ax = plt.subplots()
plt.title('Electricity production vs Electricity shadow prices (+50% fuel
demand)')
# data1 = plt.plot(tot_elec_production.index, -
1*0.1*tot_elec_production,label='Electricity production',linestyle='--')
data2 =
plt.stackplot(tot_elec_production_8050.index,wind_8050,hydro_8050,pv_8050,n
uclear_8050,storage_8050,chp_8050,
labels=['Wind','Hydro','PV','Nuclear','Storage','CHP'],colors=pal)
ax.set_xlabel("Timestep")
ax.set_ylabel("Electricity Production in TWh")
# line = plt.axhline(y=5, color='red',linewidth=0.5)
# ax.set_ylim(ymin=-1.6,ymax=-0.3)

ax2=ax.twinx()
data3 =
plt.plot(shadowprice_elec_nld_2016_8050['timestep'],10000*shadowprice_elec_
nld_2016_8050['dual-value'],label=('Shadow
price'),color='black',linewidth='0.8')
ax2.set_ylabel("Shadow price in EUR/MWh")
ax2.set_ylim(ymin=0,ymax=100)

data = data2+data3
labs = [l.get_label() for l in data]
ax.legend(data, labs, bbox_to_anchor=(1.12, 1.0), loc='upper left')
ax.tick_params(axis='x', rotation=330)
# plt.legend()
plt.show()
```

Duration curves

```
#%%DURATION CURVE FUNCTION
def deriveDurationVals(vals, valBinResol):
    samplVals = []
    percExceeded = []
    vals = pd.Series(vals)
    numVals = len(vals)
    min_value = vals.min()
    max_value = vals.max()

    for val in np.arange(min_value, max_value, valBinResol):
        samplVals.append(val)
        binExceededPerc = len(vals[vals > val])*100/numVals
        percExceeded.append(binExceededPerc)

    return {'sampl_vals': samplVals, 'perc_exceeded': percExceeded}
```

```
#%% PRICE DURATION DATA
# data samples bad weather
optimal_scenario_2010 = 10000*shadowprice_elec_nld_2010['dual-value']
fuel_demand_30_2010 = 10000*shadowprice_elec_nld_2010_8030['dual-value']
fuel_demand_50_2010 = 10000*shadowprice_elec_nld_2010_8050['dual-value']

# data samples all weather optimal scenario
optimal_scenario_2010 = 10000*shadowprice_elec_nld_2010['dual-value']
optimal_scenario_2016 = 10000*shadowprice_elec_nld_2016['dual-value']
optimal_scenario_2015 = 10000*shadowprice_elec_nld_2015['dual-value']

# data samples all +30% fuel demand scenarios
fuel_demand_2010_30 = 10000*shadowprice_elec_nld_2010_8030['dual-value']
fuel_demand_2016_30 = 10000*shadowprice_elec_nld_2016_8030['dual-value']
fuel_demand_2015_30 = 10000*shadowprice_elec_nld_2015_8030['dual-value']

# data samples all +50% fuel demand scenarios
fuel_demand_2010_50 = 10000*shadowprice_elec_nld_2010_8050['dual-value']
fuel_demand_2016_50 = 10000*shadowprice_elec_nld_2016_8050['dual-value']
fuel_demand_2015_50 = 10000*shadowprice_elec_nld_2015_8050['dual-value']

# derive duration plot values using the function
pc_opt_scenario_2010_opt = deriveDurationVals(optimal_scenario_2010, 0.01)
pc_opt_scenario_2010_8030 = deriveDurationVals(fuel_demand_30_2010, 0.01)
pc_opt_scenario_2010_8050 = deriveDurationVals(fuel_demand_50_2010, 0.01)

# derive duration plot values using the function +50% fuel demand
pc_scenario_2010_8030 = deriveDurationVals(fuel_demand_2010_30, 0.01)
pc_scenario_2016_8030 = deriveDurationVals(fuel_demand_2016_30, 0.01)
pc_scenario_2015_8030 = deriveDurationVals(fuel_demand_2015_30, 0.01)

# derive duration plot values using the function +50% fuel demand
pc_scenario_2010_8050 = deriveDurationVals(fuel_demand_2010_50, 0.01)
pc_scenario_2016_8050 = deriveDurationVals(fuel_demand_2016_50, 0.01)
pc_scenario_2015_8050 = deriveDurationVals(fuel_demand_2015_50, 0.01)

# derive duration plot values using the function all weather optimal
scenario
pc_opt_scenario_2010 = deriveDurationVals(optimal_scenario_2010, 0.01)
pc_opt_scenario_2015 = deriveDurationVals(optimal_scenario_2016, 0.01)
pc_opt_scenario_2016 = deriveDurationVals(optimal_scenario_2015, 0.01)
```

```
#%% plot the duration curve all weather
fig, ax = plt.subplots()
ax.plot(pc_opt_scenario_2010["perc_exceeded"],
pc_opt_scenario_2010["sampl_vals"],label='Bad weather',color='red')
ax.plot(pc_opt_scenario_2016["perc_exceeded"],
pc_opt_scenario_2016["sampl_vals"],label='Normal weather',color='orange')
ax.plot(pc_opt_scenario_2015["perc_exceeded"],
pc_opt_scenario_2015["sampl_vals"],label='Good weather',color='limegreen')
plt.ylim(ymin=0,ymax=120)
plt.title("Price duration curve for different weather scenarios")
plt.ylabel("Hourly price in EUR/MWh")
plt.xlabel("Duration in % of time")
plt.legend()
plt.show()

#%% plot the duration curve for 30% fuel demand
fig, ax = plt.subplots()
ax.plot(pc_scenario_2010_8030["perc_exceeded"],
pc_scenario_2010_8030["sampl_vals"],label='Bad weather +30% fuel
demand',color='red')
ax.plot(pc_scenario_2016_8030["perc_exceeded"],
pc_scenario_2016_8030["sampl_vals"],label='Normal weather +30% fuel
demand',color='orange')
ax.plot(pc_scenario_2015_8030["perc_exceeded"],
pc_scenario_2015_8030["sampl_vals"],label='Good weather +30% fuel
demand',color='limegreen')
plt.ylim(ymin=0,ymax=120)
plt.title("Price duration curve +30% fuel demand")
plt.ylabel("Hourly price in EUR/MWh")
plt.xlabel("Duration in % of time")
plt.legend()
plt.show()

#%% plot the duration curve for 50% fuel demand
fig, ax = plt.subplots()
ax.plot(pc_scenario_2010_8050["perc_exceeded"],
pc_scenario_2010_8050["sampl_vals"],label='Bad weather +50% fuel
demand',color='red')
ax.plot(pc_scenario_2016_8050["perc_exceeded"],
pc_scenario_2016_8050["sampl_vals"],label='Normal weather +50% fuel
demand',color='orange')
ax.plot(pc_scenario_2015_8050["perc_exceeded"],
pc_scenario_2015_8050["sampl_vals"],label='Good weather +50% fuel
demand',color='limegreen')
plt.ylim(ymin=0,ymax=120)
plt.title("Price duration curve +50% fuel demand")
plt.ylabel("Hourly price in EUR/MWh")
plt.xlabel("Duration in % of time")
plt.legend()
plt.show()
```

```
#%% LOAD DURATION CURVE DATA
demand_techs =
['battery','demand_elec','electric_heater','electric_hob','electrolysis','h
eavy_transport_ev','hp','hydrogen_to_liquids','light_transport_ev']
load_duration_2010 =
model_2050_2010.get_formatted_array('carrier_con').loc['electricity'].loc['
NLD'].loc[{'techs':demand_techs}].sum(['techs']).to_pandas()
load_duration_2015 =
model_2050_2015.get_formatted_array('carrier_con').loc['electricity'].loc['
NLD'].loc[{'techs':demand_techs}].sum(['techs']).to_pandas()
load_duration_2016 =
model_2050_2016.get_formatted_array('carrier_con').loc['electricity'].loc['
NLD'].loc[{'techs':demand_techs}].sum(['techs']).to_pandas()
#load duration curve data +30% fuel demand
load_duration_2010_8030 =
model_2050_2010_8030.get_formatted_array('carrier_con').loc['electricity'].
loc['NLD'].loc[{'techs':demand_techs}].sum(['techs']).to_pandas()
load_duration_2015_8030 =
model_2050_2015_8030.get_formatted_array('carrier_con').loc['electricity'].
loc['NLD'].loc[{'techs':demand_techs}].sum(['techs']).to_pandas()
load_duration_2016_8030 =
model_2050_2016_8030.get_formatted_array('carrier_con').loc['electricity'].
loc['NLD'].loc[{'techs':demand_techs}].sum(['techs']).to_pandas()
#load duration curve data +50% fuel demand
load_duration_2010_8050 =
model_2050_2010_8050.get_formatted_array('carrier_con').loc['electricity'].
loc['NLD'].loc[{'techs':demand_techs}].sum(['techs']).to_pandas()
load_duration_2015_8050 =
model_2050_2015_8050.get_formatted_array('carrier_con').loc['electricity'].
loc['NLD'].loc[{'techs':demand_techs}].sum(['techs']).to_pandas()
load_duration_2016_8050 =
model_2050_2016_8050.get_formatted_array('carrier_con').loc['electricity'].
loc['NLD'].loc[{'techs':demand_techs}].sum(['techs']).to_pandas()
#convert units to MW
load_duration_2010 = load_duration_2010*-100
load_duration_2015 = load_duration_2015*-100
load_duration_2016 = load_duration_2016*-100
load_duration_2010_8030 = load_duration_2010_8030*-100
load_duration_2015_8030 = load_duration_2015_8030*-100
load_duration_2016_8030 = load_duration_2016_8030*-100
load_duration_2010_8050 = load_duration_2010_8050*-100
load_duration_2015_8050 = load_duration_2015_8050*-100
load_duration_2016_8050 = load_duration_2016_8050*-100
# derive duration plot values using the function all weather optimal
scenario
pc_ld_2010 = deriveDurationVals(load_duration_2010, 0.01)
pc_ld_2016 = deriveDurationVals(load_duration_2016, 0.01)
pc_ld_2015 = deriveDurationVals(load_duration_2015, 0.01)
# derive duration plot values using the function +30% fuel demand
pc_ld_2010_8030 = deriveDurationVals(load_duration_2010_8030, 0.01)
pc_ld_2016_8030 = deriveDurationVals(load_duration_2016_8030, 0.01)
pc_ld_2015_8030 = deriveDurationVals(load_duration_2015_8030, 0.01)
# derive duration plot values using the function +50% fuel demand
pc_ld_2010_8050 = deriveDurationVals(load_duration_2010_8050, 0.01)
pc_ld_2016_8050 = deriveDurationVals(load_duration_2015_8050, 0.01)
pc_ld_2015_8050 = deriveDurationVals(load_duration_2015_8050, 0.01)
```

```
#%% plot load duration curves optimal scenario
fig, ax = plt.subplots()
ax.plot(pc_ld_2010["perc_exceeded"], pc_ld_2010["sampl_vals"],label='Bad
weather',color='red')
ax.plot(pc_ld_2016["perc_exceeded"], pc_ld_2016["sampl_vals"],label='Normal
weather',color='orange')
ax.plot(pc_ld_2015["perc_exceeded"], pc_ld_2015["sampl_vals"],label='Good
weather',color='limegreen')
plt.ylim(ymin=0,ymax=120)
plt.title("Load duration curve for different weather scenarios")
plt.ylabel("Demand in MW")
plt.xlabel("Duration in % of time")
plt.legend()
plt.show()

#%% plot load duration curves +30% fuel demand
fig, ax = plt.subplots()
ax.plot(pc_ld_2010_8030["perc_exceeded"],
pc_ld_2010_8030["sampl_vals"],label='Bad weather',color='red')
ax.plot(pc_ld_2016_8030["perc_exceeded"],
pc_ld_2016_8030["sampl_vals"],label='Normal weather',color='orange')
ax.plot(pc_ld_2015_8030["perc_exceeded"],
pc_ld_2015_8030["sampl_vals"],label='Good weather',color='limegreen')
# plt.ylim(ymin=0,ymax=120)
plt.title("Load duration curve for (+30% fuel demand)")
plt.ylabel("Demand in MW")
plt.xlabel("Duration in % of time")
plt.legend()
plt.show()

#%% plot load duration curves +50% fuel demand
fig, ax = plt.subplots()
ax.plot(pc_ld_2010_8050["perc_exceeded"],
pc_ld_2010_8050["sampl_vals"],label='Bad weather',color='red')
ax.plot(pc_ld_2016_8050["perc_exceeded"],
pc_ld_2016_8050["sampl_vals"],label='Normal weather',color='orange')
ax.plot(pc_ld_2015_8050["perc_exceeded"],
pc_ld_2015_8050["sampl_vals"],label='Good weather',color='limegreen')
# plt.ylim(ymin=0,ymax=120)
plt.title("Load duration curve for (+50% fuel demand)")
plt.ylabel("Demand in MW")
plt.xlabel("Duration in % of time")
plt.legend()
plt.show()
```

Payback time

```
#TOTAL ELECTRICITY SYSTEM COST IN BILLION €
cost_2010 =
model_2050_2010.get_formatted_array('cost_investment').loc['monetary'].loc[
{'techs':electricity_techs}].sum(['locs']).to_pandas().sum()
cost_2010_8030 =
model_2050_2010_8030.get_formatted_array('cost_investment').loc['monetary']
.loc[{'techs':electricity_techs}].sum(['locs']).to_pandas().sum()
cost_2010_8050 =
model_2050_2010_8050.get_formatted_array('cost_investment').loc['monetary']
.loc[{'techs':electricity_techs}].sum(['locs']).to_pandas().sum()

cost_2015 =
model_2050_2015.get_formatted_array('cost_investment').loc['monetary'].loc[
{'techs':electricity_techs}].sum(['locs']).to_pandas().sum()
cost_2015_8030 =
model_2050_2015_8030.get_formatted_array('cost_investment').loc['monetary']
.loc[{'techs':electricity_techs}].sum(['locs']).to_pandas().sum()
cost_2015_8050 =
model_2050_2015_8050.get_formatted_array('cost_investment').loc['monetary']
.loc[{'techs':electricity_techs}].sum(['locs']).to_pandas().sum()

cost_2016 =
model_2050_2016.get_formatted_array('cost_investment').loc['monetary'].loc[
{'techs':electricity_techs}].sum(['locs']).to_pandas().sum()
cost_2016_8030 =
model_2050_2016_8030.get_formatted_array('cost_investment').loc['monetary']
.loc[{'techs':electricity_techs}].sum(['locs']).to_pandas().sum()
cost_2016_8050 =
model_2050_2016_8050.get_formatted_array('cost_investment').loc['monetary']
.loc[{'techs':electricity_techs}].sum(['locs']).to_pandas().sum()
```

```
#VARIABLE COST OF SYSTEM
var_cost_2010 =
model_2050_2010.get_formatted_array('cost_var').loc['monetary'].loc[{'techs
':electricity_techs}].sum(['locs','timesteps']).to_pandas().sum()
var_cost_2010_8030 =
model_2050_2010_8030.get_formatted_array('cost_var').loc['monetary'].loc[{'
techs':electricity_techs}].sum(['locs','timesteps']).to_pandas().sum()
var_cost_2010_8050 =
model_2050_2010_8050.get_formatted_array('cost_var').loc['monetary'].loc[{'
techs':electricity_techs}].sum(['locs','timesteps']).to_pandas().sum()

var_cost_2015 =
model_2050_2015.get_formatted_array('cost_var').loc['monetary'].loc[{'techs
':electricity_techs}].sum(['locs','timesteps']).to_pandas().sum()
var_cost_2015_8030 =
model_2050_2015_8030.get_formatted_array('cost_var').loc['monetary'].loc[{'
techs':electricity_techs}].sum(['locs','timesteps']).to_pandas().sum()
var_cost_2015_8050 =
model_2050_2015_8050.get_formatted_array('cost_var').loc['monetary'].loc[{'
techs':electricity_techs}].sum(['locs','timesteps']).to_pandas().sum()

var_cost_2016 =
model_2050_2016.get_formatted_array('cost_var').loc['monetary'].loc[{'techs
':electricity_techs}].sum(['locs','timesteps']).to_pandas().sum()
var_cost_2016_8030 =
model_2050_2016_8030.get_formatted_array('cost_var').loc['monetary'].loc[{'
techs':electricity_techs}].sum(['locs','timesteps']).to_pandas().sum()
var_cost_2016_8050 =
model_2050_2016_8050.get_formatted_array('cost_var').loc['monetary'].loc[{'
techs':electricity_techs}].sum(['locs','timesteps']).to_pandas().sum()
```

```
#CONVERT BILLION EUROS TO EUROS
cost_2010 = cost_2010*1e9
cost_2010_8030 = cost_2010_8030*1e9
cost_2010_8050 = cost_2010_8050*1e9
cost_2015 = cost_2015*1e9
cost_2015_8030 = cost_2015_8030*1e9
cost_2015_8050 = cost_2015_8050*1e9
cost_2016 = cost_2016*1e9
cost_2016_8030 = cost_2016_8030*1e9
cost_2016_8050 = cost_2016_8050*1e9

var_cost_2010 = var_cost_2010*1e9
var_cost_2010_8030 = var_cost_2010_8030*1e9
var_cost_2010_8050 = var_cost_2010_8050*1e9
var_cost_2015 = var_cost_2015*1e9
var_cost_2015_8030 = var_cost_2015_8030*1e9
var_cost_2015_8050 = var_cost_2015_8050*1e9
var_cost_2016 = var_cost_2016*1e9
var_cost_2016_8030 = var_cost_2016_8030*1e9
var_cost_2016_8050 = var_cost_2016_8050*1e9
```

```
#REVENUE FROM ELECTRICITY
#total electricity production in MWh
total_elec_production_2010 =
1e5*model_2050_2010.get_formatted_array('carrier_prod').loc[{'techs':electr
icity_techs}].loc[{'carriers':'electricity'}].sum(['techs','timesteps','loc
s']).to_pandas()
total_elec_production_2010_8030 =
1e5*model_2050_2010_8030.get_formatted_array('carrier_prod').loc[{'techs':e
lectricity_techs}].loc[{'carriers':'electricity'}].sum(['techs','timesteps'
,'locs']).to_pandas()
total_elec_production_2010_8050 =
1e5*model_2050_2010_8050.get_formatted_array('carrier_prod').loc[{'techs':e
lectricity_techs}].loc[{'carriers':'electricity'}].sum(['techs','timesteps'
,'locs']).to_pandas()

total_elec_production_2015 =
1e5*model_2050_2015.get_formatted_array('carrier_prod').loc[{'techs':electr
icity_techs}].loc[{'carriers':'electricity'}].sum(['techs','timesteps','loc
s']).to_pandas()
total_elec_production_2015_8030 =
1e5*model_2050_2015_8030.get_formatted_array('carrier_prod').loc[{'techs':e
lectricity_techs}].loc[{'carriers':'electricity'}].sum(['techs','timesteps'
,'locs']).to_pandas()
total_elec_production_2015_8050 =
1e5*model_2050_2015_8050.get_formatted_array('carrier_prod').loc[{'techs':e
lectricity_techs}].loc[{'carriers':'electricity'}].sum(['techs','timesteps'
,'locs']).to_pandas()

total_elec_production_2016 =
1e5*model_2050_2016.get_formatted_array('carrier_prod').loc[{'techs':electr
icity_techs}].loc[{'carriers':'electricity'}].sum(['techs','timesteps','loc
s']).to_pandas()
total_elec_production_2016_8030 =
1e5*model_2050_2016_8030.get_formatted_array('carrier_prod').loc[{'techs':e
lectricity_techs}].loc[{'carriers':'electricity'}].sum(['techs','timesteps'
,'locs']).to_pandas()
total_elec_production_2016_8050 =
1e5*model_2050_2016_8050.get_formatted_array('carrier_prod').loc[{'techs':e
lectricity_techs}].loc[{'carriers':'electricity'}].sum(['techs','timesteps'
,'locs']).to_pandas()
```

```
#average electricity prices in €/MWh
avg_shadow_price_2010 = statistics.mean(shadowprice_elec_nld_2010['dual-
value'])*10000
avg_shadow_price_2010_8030 =
statistics.mean(shadowprice_elec_nld_2010_8030['dual-value'])*10000
avg_shadow_price_2010_8050 =
statistics.mean(shadowprice_elec_nld_2010_8050['dual-value'])*10000

avg_shadow_price_2015 = statistics.mean(shadowprice_elec_nld_2015['dual-
value'])*10000
avg_shadow_price_2015_8030 =
statistics.mean(shadowprice_elec_nld_2015_8030['dual-value'])*10000
avg_shadow_price_2015_8050 =
statistics.mean(shadowprice_elec_nld_2015_8050['dual-value'])*10000

avg_shadow_price_2016 = statistics.mean(shadowprice_elec_nld_2016['dual-
value'])*10000
avg_shadow_price_2016_8030 =
statistics.mean(shadowprice_elec_nld_2016_8030['dual-value'])*10000
avg_shadow_price_2016_8050 =
statistics.mean(shadowprice_elec_nld_2016_8050['dual-value'])*10000
```

```
#revenue €
revenue_2010 = total_elec_production_2010*avg_shadow_price_2010
revenue_2010_8030 =
total_elec_production_2010_8030*avg_shadow_price_2010_8030
revenue_2010_8050 =
total_elec_production_2010_8050*avg_shadow_price_2010_8050

revenue_2015 = total_elec_production_2015*avg_shadow_price_2015
revenue_2015_8030 =
total_elec_production_2015_8030*avg_shadow_price_2015_8030
revenue_2015_8050 =
total_elec_production_2015_8050*avg_shadow_price_2015_8050

revenue_2016 = total_elec_production_2016*avg_shadow_price_2016
revenue_2016_8030 =
total_elec_production_2016_8030*avg_shadow_price_2016_8030
revenue_2016_8050 =
total_elec_production_2016_8050*avg_shadow_price_2016_8050
```

```
#net profit in €
net_profit_2010 = revenue_2010 - var_cost_2010
net_profit_2010_8030 = revenue_2010_8030 - var_cost_2010_8030
net_profit_2010_8050 = revenue_2010_8050 - var_cost_2010_8050

net_profit_2015 = revenue_2015 - var_cost_2015
net_profit_2015_8030 = revenue_2015_8030 - var_cost_2015_8030
net_profit_2015_8050 = revenue_2015_8050 - var_cost_2015_8050

net_profit_2016 = revenue_2016 - var_cost_2016
net_profit_2016_8030 = revenue_2016_8030 - var_cost_2016_8030
net_profit_2016_8050 = revenue_2016_8050 - var_cost_2016_8050

#payback time in years
payback_time_2010 = cost_2010/net_profit_2010
payback_time_2010_8030 = cost_2010_8030/net_profit_2010_8030
payback_time_2010_8050 = cost_2010_8050/net_profit_2010_8050

payback_time_2015 = cost_2015/net_profit_2015
payback_time_2015_8030 = cost_2015_8030/net_profit_2015_8030
payback_time_2015_8050 = cost_2015_8050/net_profit_2015_8050

payback_time_2016 = cost_2016/net_profit_2016
payback_time_2016_8030 = cost_2016_8030/net_profit_2016_8030
payback_time_2016_8050 = cost_2016_8050/net_profit_2016_8050
```

```
#%% COST RECOVERY ANALYSIS ONSHORE WIND
onshore_wind = ['wind_onshore_monopoly','wind_onshore_competeing']

cost_onshore_wind_2016 =
model_2050_2016.get_formatted_array('cost_investment').loc['monetary'].loc[
{'techs':onshore_wind}].sum(['locs']).to_pandas().sum()
cost_onshore_wind_2016 = 1e9*cost_onshore_wind_2016 #convert billion euros
to euros

var_cost_onshore_wind_2016 =
model_2050_2016.get_formatted_array('cost_var').loc['monetary'].loc[{'techs
':onshore_wind}].sum(['locs','timesteps']).to_pandas().sum()
var_cost_onshore_wind_2016 = 1e9*var_cost_onshore_wind_2016

elec_supplied =
model_2050_2016.get_formatted_array('carrier_prod').loc[{'carriers':'electr
icity'}].loc[{'techs':onshore_wind}].sum(['techs','timesteps','locs']).to_p
andas()
elec_supplied = 1e5*elec_supplied #convert to MWh

avg_shadow_price_2016 = statistics.mean(shadowprice_elec_nld_2016['dual-
value'])*10000

revenue = avg_shadow_price_2016*elec_supplied
net_profit = revenue - var_cost_onshore_wind_2016

payback_time =
(cost_onshore_wind_2016+var_cost_onshore_wind_2016)/net_profit
```