Developing and Showcasing a Method for Choosing and Comparing Energy Supply System Modelling and Optimisation Software

Vom Promotionsausschuss der
Technischen Universität Hamburg
zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

Von

Mathias Ammon

Aus

Tübingen

2024

1. Gutachter:	Prof. Dr.-Ing. Alfons Kather
2. Gutachter:	Prof. Dr.-Ing. Gerhard Schmitz
Prüfungsvorsitzender:	Prof. Dr.-Ing. Christian Becker

Tag der mündlichen Prüfung: 24.04.2024

Autor: https://orcid.org/0009-0006-8562-1560
DOI: https://doi.org/10.15480/882.9690
Handle: https://hdl.handle.net/11420/47869

Acknowledgement

First and foremost, I would like to extend my sincerest gratitude to Prof. Dr.-Ing. Alfons Kather for the immense level of trust he put in me. The number of liberties granted and possibilities offered is far beyond any PhD student's dream. The past four years were undoubtedly the most successful of my life so far, both professionally and personally. And I will be forever grateful to Mr Kather for offering me this position as a PhD student and guiding and supervising me through this process.

Second of all, I wish to express my utmost thanks to Dr.-Ing. Kristin Abel-Günther for her trust in my capabilities as a researcher and software developer. I am also deeply grateful for her tremendous amount of support during times of financial uncertainty. Furthermore, I would like to thank her, also in the name of all the students, for her insightful comments, suggestions, feedback and support during the time of supervision of the student theses accompanying this work.

I would also like to offer my special thanks to Prof. Dr.-Ing. Gerhard Schmitz for having the will and patience to read and evaluate my thesis and to Prof. Dr.-Ing. Christian Becker for chairing the PhD-defense.

In addition, I would like to express my utmost thanks and respect for MSc. Max Reimer, MSc. Tim Hanke, MSc. Darwin Schnute, BSc. Andreas Jessen, BSc. Cedric Körber, BSc. Arne Wilckens, BSc. Hannes Schoeneich, BSc. Mark Lüdcke, BSc. Timm Sievers and BSc. Leon Zickert for their invaluable contribution to *Tessif* and the whole concept of this thesis.

I also want to thank Dr.-Ing. Tobias Zimmermann and MSc. Arne Buntjer for their highly-valued comments and suggestions during the supervision process of several student theses. On this note, I would also like to thank the two above and the rest of my colleagues at the Institute of Energy Systems, part of the Hamburg University of Technology, for challenging me, supporting me and keeping me on my toes.

Additionally, I would like to thank Dr. EUR. Ing Styllianos Raffailidis for always keeping a keen eye on the formalities and ensuring all necessary steps are undertaken for this thesis to happen.

A big token of appreciation also goes out to Dr.-Ing. Christian Scharfetter and the whole team behind the master program lecture "steam turbines", arguably one of the most practical-oriented, multiperspective courses of the entire program, offering me much-needed distraction and tasks to focus on besides contributing to this work.

Furthermore, a huge "thanks" goes to my wife. I cannot adequately put in words the amount of gratitude I feel towards her for her unwavering support and trust, as well as for her heads up and patience with me. I feel blessed for her having my back all the way.

Last but not least, I would like to express my utmost thanks to my parents, my grandparents and my brother for believing in me, supporting me and being patient with me.

Hamburg, June 2024

Abstract

The *Paris Agreement* under the *United Nations Framework Convention on Climate Change* aims to reduce global greenhouse gas emissions, targeting a maximum 2-degree Celsius temperature increase this century. A key strategy involves increasing renewable energy share and storage capabilities within existing energy supply system infrastructures, necessitating major restructuring processes. Energy supply system modelling and optimisation software can facilitate these changes, which resulted in many commercial and open-source tools with varying backgrounds, objectives, and use cases.

Researchers, engineers, and decision-makers face challenges in selecting the most appropriate software for specific purposes. Although structured comparisons of available options are valuable case and tool-specific nuances often only emerge during result analysis. The current literature lacks a comprehensive result-based approach for choosing energy supply system modelling and optimisation software.

This thesis proposes a result-based Comparative Method for selecting and comparing energy supply system modelling and optimisation software, enabling systematic comparison centred on optimisation results. The Comparative Method encompasses a rigorous preselection phase, a modelling step, and a versatile, application-oriented result-analysis phase. In addition, an outside-of-this-method reusable structured procedure involving a graph-theory-inspired post-processing strategy for energy supply system models is proposed.

To support the scientific community in terms of compiling result-based comparisons, several energy supply system model scenario combinations have been devised. These address modern energy infrastructure challenges and serve as a basis for generating result-based comparisons in subsequent research. In this work, the developed combinations are utilised during an extensive case study to compare four modern, highly capable free and open-source software tools and to illustrate the method's features and applicability.

As part of this research, a free and open-source software framework, called *Tessif*, is developed to streamline the execution of the derived Comparative Method during the case study. Thereby, *Tessif* is designed to also generally assist in modelling energy supply systems and be applicable beyond the scope of this thesis.

In conclusion, this work provides a comprehensive summary of insights on modelling and optimising energy supply systems, along with the associated software. It presents recommendations for prospective software users and outlines a set of core features for the advancement and development of future modelling tools.

Kurzfassung

Das *Pariser Abkommen* im Zuge des *Rahmenübereinkommens der Vereinten Nationen über Klimaänderungen* zielt darauf ab, die weltweiten Treibhausgasemissionen zu reduzieren und den Temperaturanstieg in diesem Jahrhundert auf maximal 2 Grad Celsius zu begrenzen. Eine Schlüsselstrategie spielt dabei die Erhöhung des Anteils erneuerbarer Energien und Speicherkapazitäten innerhalb der existierenden Energieversorgungs-Infrastruktur. Entsprechend umfangreiche Umstrukturierungsprozesse sind zu erwarten. Software zur Modellierung und Optimierung von Energieversorgungssystemen ist in der Lage, diese Umstrukturierungsprozesse sinnvoll zu begleiten. Deshalb existiert inzwischen eine große Anzahl entsprechender kommerzieller und quelloffener Sotware-Werkzeuge mit unterschiedlichen Hintergründen, Zielen und Anwendungsfällen.

Forscher, Ingenieure und Entscheidungsträger stehen vor der Herausforderung, die am besten geeignete Software für bestimmte Zwecke auszuwählen. Obwohl strukturierte Vergleiche der verfügbaren Optionen existieren und wertvoll sind, werden wichtige fall- und werkzeugspezifische Einzelheiten oft erst während der Ergebnisanalyse identifiziert. In der gegenwärtigen Literatur fehlt ein umfassender ergebnisbasierter Ansatz für die Auswahl von Software zur Modellierung und Optimierung von Energieversorgungssystemmodellen.

Diese Arbeit schlägt eine ergebnisbasierte Methode zur Auswahl und zum Vergleich von Software zur Modellierung und Optimierung von Energiesystemen vor, die einen systematischen Vergleich auf Basis der Optimierungsergebnisse durchführt. Die Methode umfasst dabei eine detaillierte Vorauswahlphase, eine Modellierungsphase und eine vielseitige, anwendungsorientierte Ergebnisanalysephase. Darüber hinaus wird ein effizientes, außerhalb dieser Methode, wiederverwendbares Verfahren mit einer von der Graphentheorie inspirierten Post-Processing-Strategie vorgeschlagen.

Um die wissenschaftliche Gemeinschaft in Bezug auf die Zusammenstellung Ergebnis basierender Vergleiche zu unterstützen, werden verschiedene Energiesystem-Modelle und Szenarien entwickelt und empfohlen. Diese orientieren sich dabei an den Fragestellungen und Herausforderungen moderner Energie-Infrastrukturen und zielen darauf ab, eine gemeinsame Grundlage für ergebnisorientierte Vergleiche zu schaffen. Diese Modellszenarien werden im Zuge einer ausführlichen Fallstudie genutzt, um vier moderne, hochleistungsfähige, freie und quelloffene Softwarewerkzeuge miteinander zu vergleichen und die Anwendbarkeit der Methode zu veranschaulichen.

Im Rahmen dieser Arbeit wird zudem ein offenes und quelloffenes Software-Framework namens *Tessif* entwickelt, das die strukturierte und reproduzierbare Ausführung der vergleichenden Methode ermöglicht. *Tessif* ist dabei so konzipiert, dass es nicht nur spezifisch in dieser Thesis, sondern auch allgemein zur Unterstützung bei der Modellierung von Energieversorgungssystemen eingesetzt werden kann.

Abschließend bietet die Arbeit eine umfassende Zusammenfassung von Erkenntnissen über die Modellierung und Optimierung von Energieversorgungssystemen sowie der zugehörigen Software. Es werden Empfehlungen für potenzielle Softwarenutzer vorgestellt und eine Reihe von Schlüsselaspekten für die Neu- und Weiterentwicklung zukünftiger Modellierungswerkzeuge aufgezeigt.

Contents

List of Figures

List of Tables

}

Nomenclature

Abbreviations

AGV	Advanced Graph Visualisation
ASV	Advanced System Visualisation
CAPEX	Capital Expenditure
CHP	Combined Heat and Power
CompC	Component-Commitment
CompCnE	Component-Commitment and Component-Expansion
CompE	Component-Expansion
CRA	Comparative Result Analysis
DIA	Detailed Individual Analysis
ESSMOS	Energy Supply System Modelling and Optimisation Software
EV	Error Value
FAS	Further Analysis Selection
FIAS	Frankfurt Institute of Advanced Studies
FINE	Framework for Integrated Energy Assessment
FOS	Free and Open-Source
FOSS	Free and Open-Source Software
GGV	General Graph Visualisation
GSV	General System Visualisation
HPR	High-Priority Results
IET	Institute of Energy Systems
IGR	Integrated Global Results
IP	Integer Programming
IRC	Identifying Root Causes
ISD	Identifying Significant Differences
KIT	Karlsruhe Institute of Technology
LossLC	Lossless Commitment
LP	Linear Programming

MAE	Mean Absolute Error
MBE	Mean Biased Error
$MILP$	Mixed Integer Linear Programming
$MINLP$	Mixed Integer Non-Linear Programming
MME	Mean Modelling Error
MO	Multiple Output
MSC	Model Scenario Combination
NEV	Normalised Error Value
NLP	Non-Linear Programming
$NMAD$	Normalised Mean Average Deviation
$NMAE$	Normalised Mean Absolute Error
oemof	Open Energy Modelling Framework
openmod	Open Energy Modelling Initiative
$OPEX$	Operational Expenditure
PC	Parameter Check
PCC	Pearson Correlation Coefficient
PLC	Plausibility Check
PRA	Preliminary Result Analysis
$PyPSA$	Python for Power System Analysis
REF	Reference Software Tool Results
RES	None-Reference Software Tool Results
$RMSE$	Root Mean Square Error
SOC	State of Charge
$Tessif$	Transforming Energy Supply System Modelling Framework
TPC	Tabular Parameter Comparison
$TransC$	Transformer-Commitment
$TransCnE$	Transformer-Commitment and Transformer-Expansion
$TransE$	Transformer-Expansion
tsam	time series aggregation module
TUB	Technische Universität Berlin
$TUHH$	Hamburg University of Technology

Greek Symbols

η	Efficiency

σ	Standard Deviation
τ	Modelling Error Specifier

Indices

A, B	Component A/B
Cap	Capacity
Cut	Cutting Plane
$flow$	flow specific
i	Index/Number
in	Inflow related
max	Minimum
min	Maximum
n	Number of Components
obs	Observed
Opt	Optimum
$OrigCap$	Original Capacity
$out, char$	Characteristic Outflow
out	Outflow Related
pre	Predicted
ref	Reference
res	Compared Result
s	Software Tool
t	Time/Timestep
x, y	Variables

Latin Symbols

$\bar{x}$	Arithmetic Mean
A	Matrix Describing Optimisation Constraints in Conjunction With b
b	Vector Describing Optimisation Constraints in Conjunction with A
$c(t)$	Time-Varying Flow Cost Results
c, f	Optimisation Problem Objective Function
C	Installed Capacity
cov	Covariance
CU	Cost Unit
cv	Characteristic Value

$e(t)$	Time-Varying Emission Results
e	Specific Emissions
EMU	Emission Unit
EU	Energy Unit
F	Flow
g	Optimisation Problem Equality Constraints
h	Optimisation Problem Inequality Constraints
K	Error Value Threshold
$mean($	Arithmetic Mean of
n	Amount/Number/Step
P	Power/Load/Energy Flow
P_i	Optimisation Problem
PU	Power Unit
S	Set of Optimisation Problem Solutions
t	Timestep
V	Set of Values Corresponding to Optimisation Problem State Variables
w	Weight Factor
$x \in$	Vector of (Linear) State Variables
x, y	Values
$y \in$	Vector of Integer State Variables

1. Introduction

The research conducted and the corresponding software the author developed in his time at the *Institute of Energy Systems* part of the *Hamburg University of Technology* are presented in this underlying PhD thesis. It represents the boiled-down results, key information, and guidance on using and evaluating the author's findings.

This introductive chapter is the first of 7. It describes the motivation behind the author's intent and the goals formulated. It also provides an initial overview over the thesis structure and the respective chapter contents.

1.1. General Motivation of this Thesis

In Article 2 of the Paris Agreement the United Nations Framework Convention on Climate Change formulated a goal to reduce the anthropogenic climate change impact by reducing global greenhouse gas emissions to limit global warming to well below two degrees Celsius within this century [1]. Subsequently, the Federal Republic of Germany aims to reduce these emissions by 80 to 90% by 2050 compared to the emissions registered in 1990. Along with this goal, national decision-makers attempt to restructure the German energy supply system to reduce its overall emission impact [2]. One option to assist in this restructuring process is to use energy supply system modelling and optimisation software to help answer questions regarding the optimal use of existing infrastructure alongside the virtual exploration of viable transformation pathways [3].

Many possible software options exist to investigate these pathways at the point in time this thesis is written. However, these software tools may differ in their main priorities, strengths and weaknesses, implying the need for a structured comparison of at least some of the options available [4]. An automated comparison with a uniform representation of the modelling results was formulated to be particularly useful [5].

Hence, this work seeks to meet the goals formulated in section 1.2.

1.2. Goals

The primary goal of this thesis is to develop a method for structurally guiding the process of choosing and comparing energy supply system modelling and optimisation software. This developed method is then to be used and showcased by conducting a case study focusing on free and open-source software.

This overarching aim is subdivided into the following list of goals.

1. Propose a sub-method for choosing a particular software based on readily available data without actual usage of the software.
2. Propose a sub-method for a result-based comparison of (pre)selected software tools.
   a) Develop a free open-source software framework for comparing energy supply system modelling software.
   b) Develop energy supply system models in conjunction with relevant and contemporary scenario formulations to investigate the various software tools.
   c) Propose a post-processing strategy to provide a uniform, concise and comprehensive result representation suited for visualisation, inherently unambiguous result data allocation and further numerical analysis.
   d) Propose a strategy to systematically identify and analyse different software tool results obtained by using the same energy supply system model scenario combination.
   e) Propose a strategy to assess the scalability of energy supply system modelling frameworks.
3. Showcase the developed method using modern and capable free and open-source software for modelling and optimising energy supply systems. Make thereby use of the systems developed for goal 2b to focus on the comparison of the selected software-tools.
4. Abstract key lessons learned in the field of modelling and optimising energy supply systems.
5. Provide modelling software-specific comparison results and recommendations based on the conducted case study and the underlying research work.
6. Propose attributes a new modelling software should have, based on the findings of this work.

The above listing is used throughout this thesis as a guideline and reference to map the individual proposals and descriptions to their respective goals. It serves both as a justification of their utility and as an orientation for the user of this method. An assessment of how the above goals are addressed in this thesis can be found at the end of this thesis in Section 6.3.

1.3. Thesis Structure

This thesis is structured into four main chapters, preceded by two conclusive chapters.

Chapter 2 provides this work's theoretical background, including assumptions about energy supply systems and essential term definitions. While these do not directly contribute to achieving the goals outlined in section 1.2, they offer crucial information for relevant sections. The chapter continues with a section on the techniques developed for representing an energy supply system model and the results obtained through investigation, directly addressing Goal 2c. The theory chapter concludes with an introduction to the statistical tools used, which are linked to the proposed strategy for identifying and analysing result differences (Goal 2d).

Chapter 3 introduces the software framework developed to achieve Goal 2a (creating a free and open-source software for comparing energy supply system modelling software), which is closely tied to fulfilling all formulated aims (except for Goal 1). The chapter begins discussing the benefits of such software, as implied by recent literature, followed by an overview of the developed framework's structure, energy supply system components, energy supply system model design, and programmatic design. The chapter also discusses the framework's potential future as free and open-source software and concludes by developing the energy supply system model scenario combinations used in this thesis, directly addressing Goal 2b.

Chapter 4 represents the core development of this thesis, encompassing the research findings and addressing most of the formulated goals. The chapter begins with a visually enhanced conceptual overview, serving as a summarising introduction, a visual guide through the method description, and a reference for future use. The chapter proceeds with a detailed description of the developed method, starting with a preselection method addressing Goal 1, followed by a modelling phase for Goal 2b, and concluding with a structured guideline on a preliminary and comparative result analysis addressing Goals 2c and 2d.

Chapter 5, the last of the four main chapters, demonstrates the developed method's capabilities, uncovers its strengths and weaknesses during practical application, and compares four modern, high-impact free and open-source energy supply system modelling software options. The chapter begins by discussing the similarities of the compared software tools and the rationale for their selection. It then demonstrates the method's application through each step, starting with the preselection, then the modelling, and the preliminary result analysis step, concluding with the comparative result analysis step. Chapter 5, in its entirety, addresses Goal 3.

Following the four main chapters, Chapter 6 presents a summary, conclusion and evaluation, including a critical assessment of this work, explicitly addressing Goals 4 to 6. The thesis ends with an outlook in Chapter 7, recommending further work on the Comparative Method and the developed software framework.

2. Theoretical Background for Developing the Software Framework and Comparative Method

This chapter provides the theoretical background for this thesis. While only Section 2.2 specifically addresses the objectives outlined in Section 1.2, the other sections contribute to the groundwork for related aspects.

Section 2.1 introduces the concept of energy supply system modelling, discussing various steps of the modelling process and aspects of computer-aided result generation. Section 2.2 presents the graph-theory-related tools utilised, developing and discussing essential visualisation techniques combined with comprehensive result data representation. Section 2.3 concludes this chapter by focusing on the statistical tools used, presenting the theoretical background of the developed statistical analysis tools.

2.1. Modelling Energy Supply Systems

Section 2.1 comprises nine subsections that elucidate the general aspects of an energy supply system model, introduce its components, elaborate on the modelling process, focus on various optimisation topics, discuss how modelling and optimisation software tools unify these aspects within a standard interface, and explain the reasoning for choosing a free and open-source software approach.

2.1.1. Defining Energy Supply System Model (ESSM)

In the context of modelling energy supply systems, the ESSM serves as an informative, theoretical representation of the system. The ESSM incorporates only relevant, logically connected information to represent a given energy supply system's status quo and investigate potential alterations for achieving specific goals. Its primary purpose is to enable research on theoretical changes that would be too costly or time-consuming when conducted on the physical system. Modelling the system also allows for isolating aspects that can be reused in subsequent work of different or larger contexts [6].

The ESSM thus refers to the digital twin or virtual replica of an existing energy supply system that could, at least theoretically, exist as a real-world application. These ESSMs serve as the basis for conducting energy supply system simulations or optimisations. They vary in size and complexity from very small and simple to serve as reference or testing environments to large and complex when trying to emulate an existing energy supply system. An ESSM is comprised of its individual components (e.g., power plant, consumer) forming a network.

Establishing an overarching goal is essential when creating an ESSM, as it guides the incorporation of relevant information. Design decisions are held accountable against this goal, including

decisions about ESSM components. In energy supply systems, this typically involves grouping subsets of techno-economic parameters into logical entities, particularly for modelling frameworks like those investigated in Chapter 5. ESSM formulation often covers only parts of an existing supply system due to its scale. It necessitates clear boundaries for precise interactions and reduces system-wide dependencies and parameters to those essential [7].

2.1.2. Components of the ESSM

The components to be modelled are represented as abstract aggregations of behavioural constraints, grouped into several abstract entities that constitute any ESSM throughout this work. As understood within this thesis's scope, a basic and generic ESSM consisting of three supply components, one transformer component, two storage components, two transport components, and two demand components is illustrated in Figure 2.1.

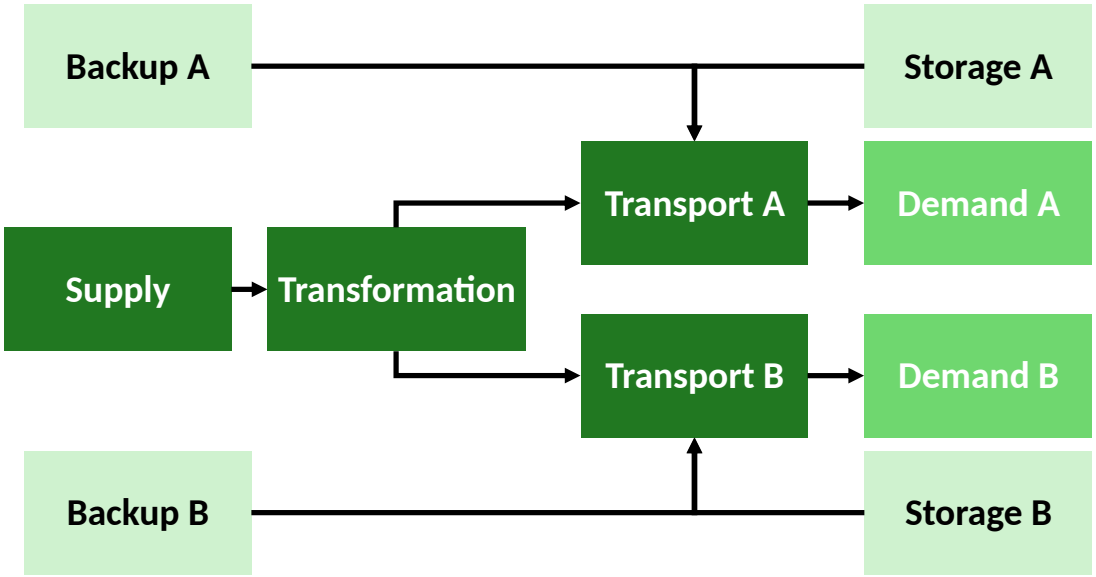

Figure 2.1.: Generic Interconnection Representation of ESSM Components. Inspired by [8].

The depicted ESSM includes one primary energy transformer, which aims to fulfil the demand of components A and B, transferring energy through transport components A and B. The transformer relies on an energy carrier from a supply source. If demands cannot be met, backup supplies A and B connect to the energy transport components. Surpluses or deficits can also be balanced using storage components A and B, provided their respective states of charge permit it.

In the chosen representation, each component is portrayed by a green box, while energy flows are visualised with arrows. The overall assembly of all boxes and arrows constitutes the ESSM.

2.1.3. General Procedure of the Modelling Process

The energy supply system modelling process in this work is regarded as a sequence of steps, illustrated by the abstract flowchart in Figure 2.2.

Input data may include weather data, load profiles, demand profiles, techno-economic component parameters, and secondary global constraints such as land usage, allowed emissions, and available

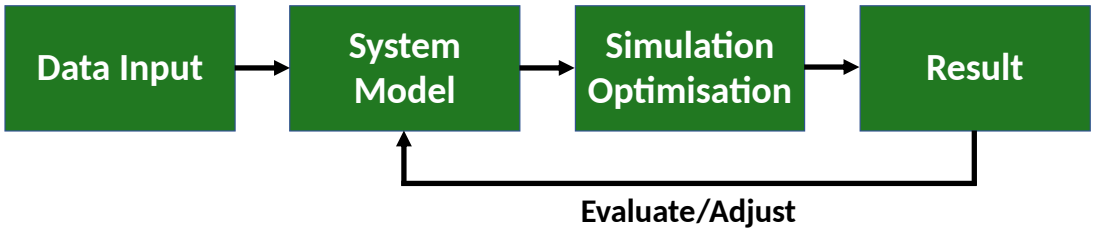

Figure 2.2.: Abstract Flow-Chart Representation of the Modelling Process. Inspired by [8].

resources. The network of interconnected components represents the ESSM containing relevant input data. This ESSM can then be utilised for software-assisted simulation or optimisation, yielding output results that characterise system behaviour subject to the imposed constraints. Results may undergo post-processing and evaluation. If needed, either the input data or the ESSM is adjusted, and the loop repeats. During ESSM creation, these loops are often executed systematically to debug and refine the design, to verify the result plausibility and to estimate required computational resources. If expectations in either domain are unmet, the ESSM is modified to rectify logical inconsistencies or reduce computational effort. This iterative process continues until the desired behaviour is achieved. [9]

2.1.4. Problem Formulation Archetypes of ESSMs

Additional input parameters are required to simulate or optimise an ESSM. These parameters describe a specific investigative problem formulation called a "scenario".

For ESSM **simulations**, these parameters typically include weather or load profile data, as well as a set of techno-economic parameters to be varied between simulations for addressing questions of interest (refer to Subsection 2.1.5 for further background on ESSM simulations). An illustrative example of a **simulation scenario** could be:

How sensitive are carbon dioxide emissions to an increase in coal-fired power plant efficiency?

For **optimisations**, these parameters generally involve a target value to be optimised (usually a cost-minimum) along with secondary system-wide constraints such as total emissions, land usage, or resource utilisation. These goal-defining parameters are grouped into logical entities constituting a specific scenario or problem formulation. An example **optimisation problem formulation** could be:

What amount of energy storage capacity should be installed to achieve a certain degree of autarky least expensively?

Both scenario/problem formulations necessitate integrating a set of parameters into the ESSM by the user. To clarify the distinction between the ESSM and the scenario and the fact that the same scenario can be formulated using multiple ESSMs and vice versa, the term "model scenario combination", abbreviated by "MSC" is used in this work. Thereby, the "energy supply system" part of ESSM is ommitted to facilitate the distinction between MSC and ESSM as well as to shorten the frequently used abbreviation "MSC".

Depending on the modelling process's application field, scenario formulations can be categorised into different archetypes. Hourcade et al. [10] divide the modelling process into three categories, one explicitly addressing the comparison of development trajectories for a given

energy supply system. This category is directly applicable here, as the developed framework (Chapter 3) and compared ESSMOS (Chapter 5) allow for such a focus.

Hourcade et al. also differentiate between ESSM and scenario formulation. Thereby, the ESSM describes the system components and their techno-economic parameters, while the scenario formulates behavioural constraints for achieving specific goals, matching the MSC concept used in this work. The process of comparing development trajectories is termed scenario analysis, typically conducted on the same ESSM using (at least) two scenarios: the **reference** and the **base** scenario.

The **reference** scenario represents the status quo of an ESSM and its future version, assuming no changes occur. It also usually implies that current subsidies are not extended after expiration. The results of this scenario serve as a reference for further comparisons.

The **base** scenario encompasses formulating goals regarding system transformation, such as reducing carbon dioxide emissions or increasing autarky with minimal costs. The scenario results represent the required transformation trajectory to achieve these goals as a basis for comparison to additional scenarios where constraints or endogenous parameters vary.

The two optimisation problem formulation archetypes **commitment problem** and **expansion problem** are defined for this thesis by generalising Hourcade et al.'s concept. These archetypes align with the **reference** and **base** scenarios but do not exclusively focus on future trajectories, thus broadening the application scale to any feasible energy supply system optimisation type. This shift in focus enhances the utility of the model scenario combinations term, particularly within the developed software framework. By combining the defined problem formulation archetype with the underlying MSC, the investigation can be distinctly labelled to accurately and unambiguously describe its scope. For instance, "grid-focused emission-constrained-expansion-problem" denotes an MSC where "grid-focused" characterises the ESSM and "emission-constrained-expansion-problem" describes the specified expansion-problem archetype.

2.1.4.1. Commitment Problem

In energy supply system modelling, the commitment problem archetype represents a typical "constraint optimisation problem" frequently employed in various modelling contexts [11]. A set of variables, defined as $V = \{X_1, X_2, \ldots, X_n\}$ each associated with a non-empty set of possible values, is contextualised using a set of constraints $C = \{C_1, C_2, \ldots, C_N\}$. [12] A solution to this "constraint satisfaction problem" is found if a combination of values satisfies all constraints C. By incorporating a fitness function to compare all possible solutions $S = \{S_1, S_2, \ldots, S_N\}$, a "constraint optimisation problem" is formulated [11]. In terms of energy supply system model optimisation, such a constraint optimisation problem can be formulated as a commitment problem archetype, represented by the following question:

Which available and controllable components should be utilised, when and to what extent, in order to meet a specific energy demand while minimising overall costs, adhering to component and energy transportation constraints, and potentially achieving secondary objectives like an emission target?

Commitment problems are widely employed in energy system modelling, making them a standard requirement for modelling software tools. They often serve as reference scenarios when comparing different model scenario combinations (assuming Hourcade et al.'s [10] terminology is applied). They also assist in daily operations, such as generating maintenance schedules or estimating levelised costs of electricity for day-ahead markets.

2.1.4.2. Expansion Problem

The expansion problem archetype builds upon the commitment problem by introducing additional states of possible values. These typically encompass installed capacities of energy transformation, energy storage, and energy transportation entities. Regarding constraint formulation, the expansion problem closely resembles the commitment problem, as it is also a constraint optimisation problem but with a (significantly) larger set of possible variable states. When applied to the practical use of energy supply system modelling, an expansion problem can be posed as the following question:

Which available or new components need to be expanded or added to achieve specific secondary objectives, such as emission targets, while minimising overall costs, abiding by given component and energy transportation constraints, and still fulfilling all energy demands?

Expansion problem archetypes are often employed to model transformation trajectories that reduce a system's footprint and enhance its resilience or self-sufficiency. They are commonly used in public decision-making for large-scale system transformations and in private supply systems on smaller scales. One current primary application is the emission-constrained expansion problem, which investigates energy supply system transformation trajectories to reduce human impact on global warming.

2.1.5. Differentiating Between Simulation and Optimisation

The terms simulation and optimisation are occasionally used interchangeably. To avoid confusion, they are treated here as distinct processes with separate objectives and, consequently, often markedly different underlying ESSM structures. Worrel et al. [13] distinguish between simulation and optimisation by stating that all endogenous parameters of the underlying ESSM are fixed in a simulation, leaving no remaining degrees of freedom. Thus, the simulation results depend solely on the exogenously defined (input) parameters. To perform a scenario analysis, endogenous parameters are manually varied. Simulations are typically employed to investigate and evaluate various socio-economic or politically induced changes in a given ESSM. Another prevalent application is forecasting future energy demands and associated greenhouse gas emissions [14].

In contrast, optimisation seeks to minimise or maximise specific result values by automatically varying endogenous parameters. The parameters commonly adjusted are component loads in commitment problems and component loads and installed capacities in expansion problems. The ESSM, therefore, possesses degrees of freedom and is represented by a set of equations and inequalities that are solved and evaluated using a fitness function. The solving process is performed by a separate software component, often referred to as "the solver", which utilises tool- and problem-formulation-specific algorithms [15].

2.1.6. Utilising Different Optimisation Time Span Lengths

When performing an ESSM optimisation, the chosen time span length significantly impacts performance and informative value [16]. Varying time span lengths can be grouped into the following three types.

First is the **perfect forecast**, wherein long periods (typically one or several years of hourly resolution) are optimised in a single batch. As the solver pursues a global optimum, the states

of the final time step may impact those of the initial time step. For instance, the wind energy produced on the 31st of December may affect a residual power plant's behaviour on the 1st of January 12 months earlier. Additionally, a long time span, particularly in mixed-integer linear programming (MILP), may result in many variables and constraints, potentially causing infeasible solving times [17].

Second is the **myopic approach**, which limits forecast periods to equidistant time span lengths, usually ranging from several days to months. Each period is optimised independently, considering carry-over constraints such as state of charge, power gradients, or minimum up and downtimes. This approach can significantly reduce computational effort by transforming one large-scale problem into smaller ones, especially with MILP formulations.

Regarding predictability, the myopic approach aligns with weather forecast realities and is deemed more realistic when such data is relevant. However, a longer time span better reflects modern investment realities and is considered advantageous for investment decisions. Optimising solely for consecutive singular time spans may risk short-term optima compromising a long-term optimal solution. This difference can lead to substantial discrepancies between the optimal solutions of the perfect forecast and myopic approaches. [16, 18]

Third is the **rolling horizon** (a specialised myopic approach), which restricts the forecast period or horizon to several equidistant time spans. Each forecast horizon is solved repeatedly and independently. After each iteration, the forecast horizon advances by a brief period, commonly called the decision horizon. The states of all variables moving out of the forecast horizon due to this shift are deemed fixed in subsequent iterations. [17]

The rolling horizon approach can balance the computation time needed for large-scale problems and a be compromise in result quality [19]. However, it necessitates a distinct algorithm to manage data transfer efficiently between horizon shifts. The advantage of this compromise depends on the forecast and decision horizon lengths, which are best chosen based on the total number of time steps and the formulated variables and constraints. Ideally, a parameter study is conducted beforehand to identify (near) optimal horizon lengths. For example, Marquant et al. [19] conducted a *Multiple Urban Energy Hub System* study using a MILP formulation comprising 201,482 variables, with 26,280 integers, across 8,760 hourly time steps. Their parameter study suggests a forecast length between 4 and 14 days and a decision length of 1 or 2 days.

2.1.7. Mathematical Optimisation Programming Types

Modelling an energy supply system to express an investigative scenario as described in Subsections 2.1.4 and 2.1.5 subsequently results in a system of equations and inequations being formulated and passed on to dedicated solver software to find an optimal or near-optimal solution. This set of stated (in)equations can be classified by its complexity which commonly results in three mathematical optimisation programming types: **Linear Programming** (**LP**), **Mixed Integer Linear Programming** (**MILP**) and **Non-Linear Programming** (**NLP**).

As for solver software, currently, two popular free and open-source options exist. The **Coin-or branch-and-cut** (**CBC**) [20] solver and the **GNU Linear Programming Kit** (**GLPK**) [21]. Bauer et al. [22] recommend using the **CBC** solver only for relatively simple ESSMs and the **GLPK** solver for more complex ones. Both solver software tools are implemented and available in the developed software framework.

2.1.7.1. Linear Programming (LP)

Linear programming describes a way to obtain the best outcome in a mathematical model represented by related linear constraints. Reusing Pena et al.'s [12] and Niesse et al.'s [11] expressions of Subsection 2.1.4 in describing a scenario archetype and applying it to Hamacher et al.'s [23] formulation, the following specifications can be made. The described set of variables defined as $V = \{X_1, \ldots, X_n\}$ can be expressed as a vector x of state variables as in $x = (1, \ldots, n)^T$. The mentioned set of constraints $C = \{C_1, \ldots, C_N\}$ can be formulated in the context of LP using the two inequalities $A \cdot x \leq b$ and $x \geq 0$, where A is a given matrix specifying the constraints in conjunction with $b = (1, \ldots, m)^T$. The LP problem can thus be formulated as:

$$\begin{aligned} &\min\ c^T \cdot x \\ &A \cdot x \leq b \\ &x \geq 0. \end{aligned} \tag{2.1}$$

Where $A \in \mathbb{R}^{m,n}$ describes the optimisation of system A via changing variable states x subject to the two constraints $A \cdot x \leq b$ and $x \geq 0$. The optimal solution is represented as the fitness function or objective function minimum $c^T \cdot x$ with $c \in \mathbb{R}^{1,n}$. The state variable parameters x are also called decision variables or activity. The non-negative constraint in equation 2.1 is implied by typical system components not being able to assume negative states as in installed capacities or current states of charge.

The LP problem's inequalities span a bounded convex polytope over which the objective function is to be optimised. The problem can be interpreted as an n-dimensional space in which the allowed states of x are bounded by an up to n-dimensional polygon of up to $n+m$ vertices (with n, m being the dimensions of x, b respectively). The LP tries to find an optimal solution respecting the objective function within this boundary. It can be shown that the optimal solution always is to be found on one of the polytope's vertices.

For an example problem of 2 state variables x_1 and x_2 subject to 5 constraints in total, a possibly resulting polygon of 5 vertices (2-dimensional polytope) in conjunction with an objective function is depicted in Figure 2.3. In this illustration, the dark green line represents the objective function. The perpendicular black arrow depicts the respective optimisation direction, and the bottom right black dot represents the found optimum. The coordinate system axes comprise the two state variables, and the vertices are visualised as light green lines.

Solver tools usually use a **simplex-method** or an **interior-point method** algorithm to solve such problems. The geometric interpretations of both methods are visualised in Figure 2.4.

For the **simplex method** ("a)" in Figure 2.4) , the solver traverses the polytope's edges from a starting vertice to its optimal vertice by calculating the vertice-specific objective function values and comparing them to their neighbours. Traversal decisions are subject to the specific algorithm deployed. In practice, a **simplex method** algorithm is often among the fastest option available. [24]

When utilising the **interior-point method** ("b)" in Figure 2.4), the solver traverses the interior of the polytope by finding objective function values inside the feasible region better than the previous one. Again, traversal decisions are subject to the specific algorithm deployed. Although, in practice, often slower than those of the **simplex method**, they are of guaranteed polynomial complexity for LPs, and algorithms exist for **non-linear programming**. **Interior-point method** algorithms, however, are unsuited for solving a series of optimisation problems and are, therefore, unfit for **mixed integer linear programming**. [25]

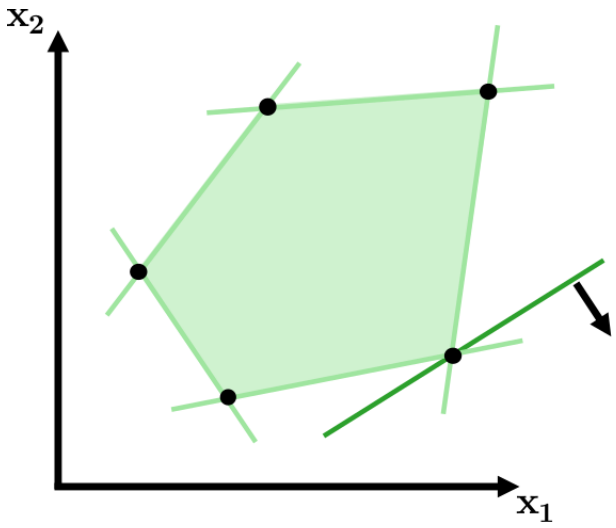

Figure 2.3.: Example LP Problem Visualisation.
Showing 2 State Variables 5 Secondary Constraints.
Objective Function = Dark Green Line; Optimisation Direction = Black Arrow
Found Maximum = Bottom Right Black Dot

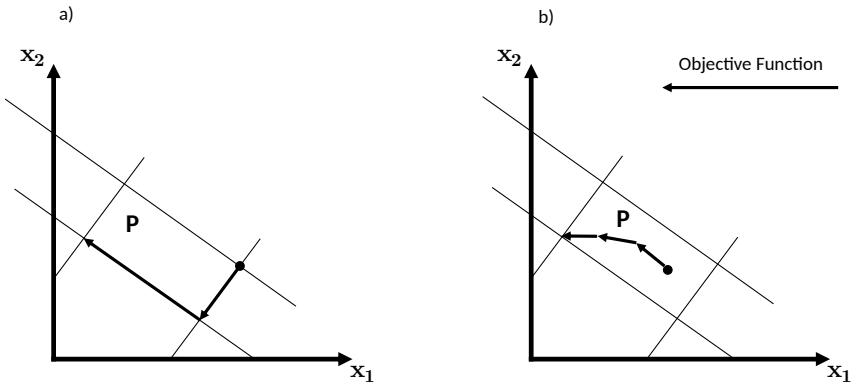

Figure 2.4.: Geometric Interpretation of LP Solving Algorithms. Inspired by [8].
a) **Simplex Method** b) **Interior-Point-Method**

The exact procedure for how these trajectories are mathematically represented and how decisions regarding their directions are made lies not within the author's realm of expert knowledge. It is hence not part of this work. Instead, it is assumed that understanding the basic working principles is of adequate utility and that the deployed solver software tools reached sufficient technical maturity to be used confidently.

2.1.7.2. Mixed Integer Linear Programming (MILP)

Mixed Integer Linear Programming (**MILP**) is a branch of **linear programming** where some of the variables are represented by integers. These variables describe discontinuous behaviour as "active or inactive" or mixed discontinuous and linear behaviour as "active or inactive and if active, then linear". [26]

In ESSMs, typical examples are fossil fuel power plants having a feasible operation range of, e.g., 40 to 100%. Hence, they are described as "either on or off, and if on, then the possible power output lies between 40 and 100%".

The on/off or yes/no decision can mathematically be represented as $y \in \{0,1\}^m$. Expanding an **LP** as described in equation 2.1 as such would then result in a **MILP** formulation that can be formulated as:

$$f(x,y) \to \min \begin{cases} g(x,y) = 0 \\ h(x,y) \leq 0 \end{cases} \text{ for } x \in \mathbb{R}^n,\ y \in \{0,1\}^m. \tag{2.2}$$

Where $f(x,y)$ describes the objective function, g and h the equality and inequality constraints, and x and y the continuous and binary variables. [22]

Finding (near) optimal solutions to mixed integer linear optimisation problems is usually done using the **branch-and-cut** (**BnC**) method, a combination of the integer optimisation problem-solving methods **branch-and-bound** and **cutting plane**. Alternatively, problem-specific heuristics are often developed. These, however, neither provide a standard solving procedure nor offer an estimation of how good a found solution is compared to the yet unknown optimal solution. In practice, **BnC** algorithms are often deployed to only find near-optimal solutions with an acceptable error margin specified beforehand. Since finding a global optimum and proof of its optimality can take a considerable amount of time. [27, 28, 29]

Most modern solver software suits implement one or multiple **branch-and-cut** algorithms due to their generic nature and their capabilities to quickly find close to optimal solutions of well-estimated error margins, serving as a good enough approximation. [20, 21, 30]

Analogous to the geometric LP interpretation, a **MILP** can also be interpreted as an n-dimensional polytope limiting the room of feasible solutions with the additional constraint of some decision variables being able only to take integer values. A two-dimensional example with x_1 being an integer variable and x_2 a continuous variable can be seen in Figure 2.5, along with the objective function (dark green line), the optimisation direction (dashed black line) and the found maximum (big black dot).

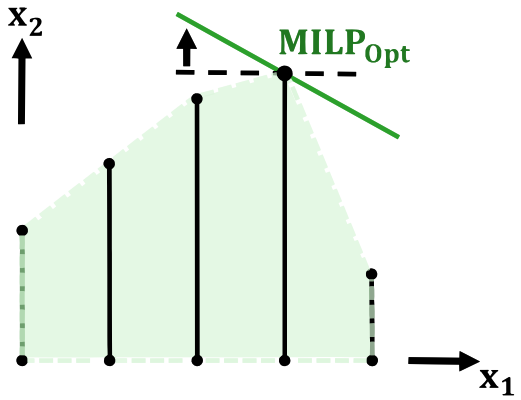

Figure 2.5.: Geometric Interpretation of a **MILP** Solving Algorithm.
x_1 = Integer Variable, x_2 = Continuous Variable
Objective Function = Dark Green Line; Found Maximum = Big Black Dot

The **branch-and-cut** method solves the **MILP** by first relaxing the integer constraints and thus creating a **linear problem**. The found solution satisfies $Ax \leq b$ but usually not $y \in \{0,1\}$. A cutting plane is added as an additional constraint satisfying all **MILP** constraints while excluding the found **LP** solution, as visualised in Figure 2.6 for a 2-dimensional **MILP**.

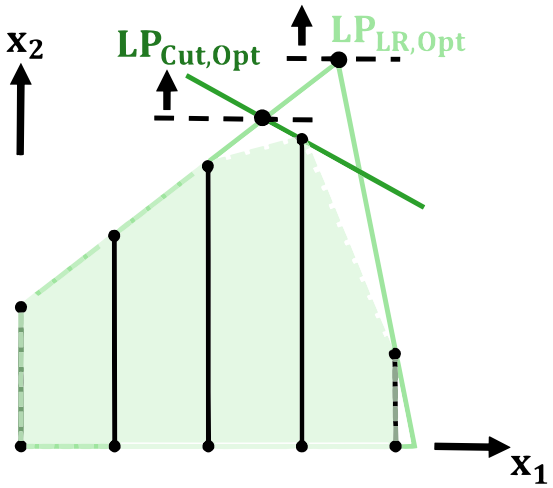

Figure 2.6.: Geometric Interpretation of a **MILP** Cutting Plane Operation.
Relaxation Optimum = $LP_{LR,Opt}$; Cutting Plain Optimum = $LP_{Cut,Opt}$

Thus, a new, smaller **LP** can be formulated, which is then solved again until a viable optimum is found, or no new cutting planes can be added to reduce the LP. In the latter case, the problem is then branched into (at least) two non-overlapping problems. A standard procedure is to branch the problem on one of the integer variables creating two new sub-problems on which the cutting plane and separation process is reiterated until a satisfactory solution is found. A geometric example representation of the described branching is depicted in Figure 2.7.

This reiteration process can be interpreted as creating a decision tree where a root node represents the initial problem. At the same time, the separations can be seen as branches, as depicted in Figure 2.8. Within this visualisation, P_0 represents the initial **LP** relaxed solution where no further cutting planes can be added. Whereas P_1 and P_2 represent the first separated subproblems. The deeper the node inside the tree, the smaller the subproblem.

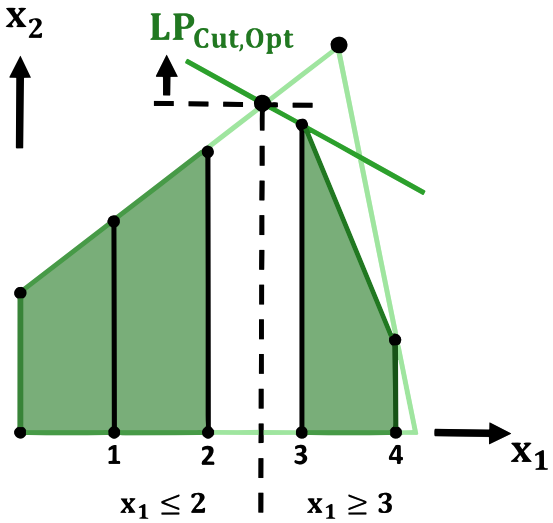

Figure 2.7.: Geometric Interpretation of a **MILP** Branching Operation.
LP Optimum ($x_1 = 2.6$) Branched Into 2 **MILP** Problems of $x_1 \leq 2$ and $x_1 \geq 3$.

The advantage of this interpretation is that the algorithm can now cut off (**bound**) whole parts of the tree where it is clear that no optimal solution is to be found (represented by nodes P_2 and P_{12}). The problem can thus be traversed without enumerating all possible solutions.

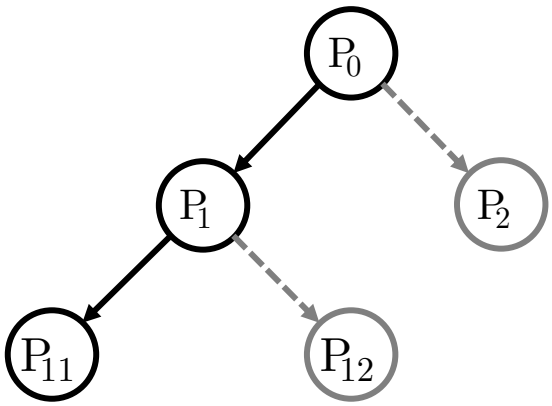

Figure 2.8.: Decision Tree Interpretation of the **Branch-and-Cut** MILP Solving Algorithm.
Greyed-Out Branches Represent Cut-Off Parts.

Due to reiterating similar problems often, **simplex method** algorithms are used in modern **MILP** solver-suits to solve the relaxed **LP** problems. [27, 28, 29]

2.1.7.3. Non-Linear and Mixed Integer Non-Linear Programming (NLP and MINLP)

Energy supply system components are often modelled most accurately using non-linear formulations. A prominent example is the load-dependent efficiency of a fossil fuel power plant. Trying to incorporate non-linear dependencies into ESSMs as described here would thus require the formulation of **non-linear programming** (**NLP**) or even **mixed integer non-linear programming** (**MINLP**) problems. Due to their complexity, however, these are currently not feasible on medium to large-scale ESSMS. Thus no, or at least not many, modern free and open-source energy supply system modelling software tools currently offer these capabilities on a sensible

scale. Some tools, like the compared *FINE* [31] and *oemof* [32] offer linearisation techniques to provide pseudo-non-linear modelling capabilities for specific components or aspects.

2.1.8. Energy Supply System Modelling and Optimisation Software (ESSMOS)

Modern ESSMOS tools typically incorporate all aspects discussed in Section 2.1. They achieve this by offering a set of (usually 4) abstraction layers that guide the process of converting a conceptual ESSM into a machine-readable format for the corresponding mathematical optimisation problem formulation. Figure 2.9 provides a typical design of these layers and the associated translation process.

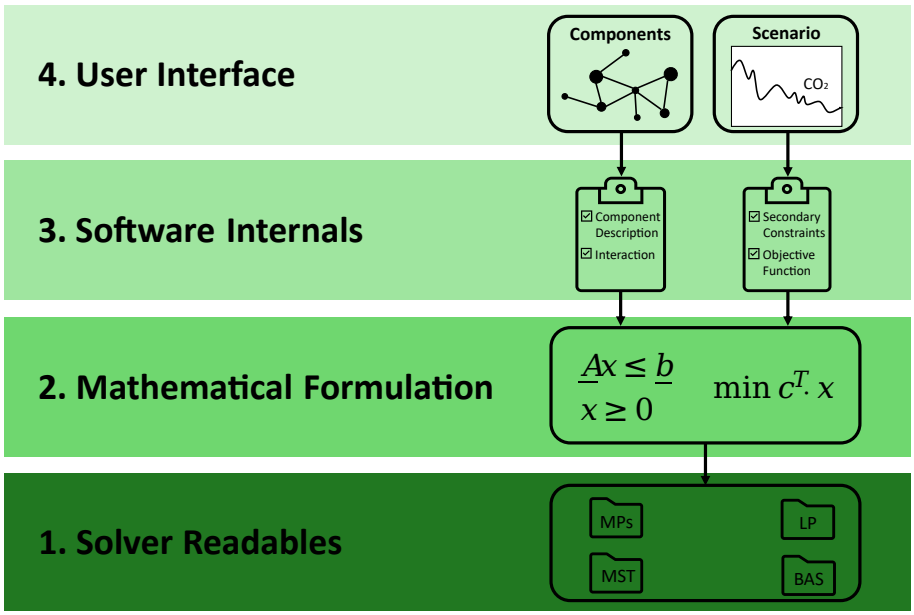

Figure 2.9.: Typical ESSMOS Abstraction Layers.

Users generally interact with the fourth abstraction layer, which is the primary software interface (fourth, light green box in Figure 2.9). This layer facilitates user-guided capabilities for creating an ESSM and one or more corresponding scenarios. The extent and content design may range from graphical user interfaces to common data description files like JSON, XML, or direct programming.

The third abstraction layer in Figure 2.9 typically consists of the software's internal structure, which provides the aforementioned interface through a set of programming codes that describe the ESSM, its components, and the corresponding scenario formulation as an engineering-oriented set of behavioural constraints (e.g., dispatchable power plant load less or equal to 800 MW).

The second abstraction layer in Figure 2.9, represented by an associated set of programming codes, manages the internal translation of the behavioural constraints from the third layer into an actual mathematical optimisation problem formulation, as described in Subsection 2.1.7.

The first abstraction layer in Figure 2.9 transforms the mathematical expressions resulting from the second layer into machine-readable optimisation problem formulations stored in optimisation problem files. These files are then typically passed on to external solver software tools, which attempt to find an optimal solution.

In addition to the general structure of ESSMOS described above, these ESSMOS usually implement several utility or convenience functions. These functions focus on data flow management, providing read and write capabilities and corresponding pre- and post-processing structures. Often, predefined result visualisation code scripts are also supplied.

Although this structural layout is commonly shared, the implementation and modelling details can vary significantly [4, 33]. The first and second abstraction layers are usually quite similar, as solver-readable file formats and mathematical formulations are often independent of the specific optimisation problem application. User interface and internal software representation and translation are what differ between ESSMOS.

2.1.9. Free and Open-Source Software (FOSS)

In accordance with the recommendations of the Open Energy Modelling Initiative (openmod) [34] and in pursuit of goal 3, all software tools employed and developed in this work belong to the FOSS category. The author concurs with the initiative's view that an open software structure will likely enhance transparency and quality while reducing redundancy [34]. Such an increase in transparency may significantly improve public discourse by rendering scientific foundations visible and accessible [35]. However, it is crucial to emphasise that the developed Comparative Method and the software framework created to facilitate the execution of this method are applicable beyond the confines of FOSS.

2.2. Developed Energy Supply System Model (ESSM) and Result Data Representation Techniques

The following section presents an approach to address goal 2c: providing a uniform, concise, and comprehensive result representation. First, Subsection 2.2.1 introduces an approach to combine discrete mathematical graphs and ESSMs. Second, a comprehensive set of visualisation techniques is derived in Subsection 2.2.2 to support the model creation process and result-data processing. Third, a numerical result data representation is developed in Subsection 2.2.3, based on the techniques discussed in Subsections 2.2.1 and 2.2.2.

2.2.1. Interpreting an ESSM as a Discrete Mathematics Graph

Examining the macro-perspective of any given energy supply system reveals two distinct features: the number and type of components present and the interdependencies between them. The primary dependency is the amount of energy flowing from one entity to another at any given time. Modelling and simplifying an energy supply system in this manner facilitates a straightforward transformation of the ESSM into a graph object, as described in discrete mathematics [36]. This interpretation translates the individual entities into graph nodes, and the energy flows into graph vertices or edges.

Abstracting an ESSM into a **discrete mathematics graph** or **graph** in the context of optimisation problems has the following benefits.

1. All optimisation results can be attributed to a node (e.g., installed capacity) or an edge (e.g., resulting net energy flow).

2. The structure or topology of the ESSM and some of its results can be comprehensively visualised using typical graph visualisation procedures, discussed in more detail in Subsection 2.2.2.

3. Although not utilised in this thesis, standard graph analytics can be applied to energy systems models abstracted in this manner. By characterising edges with (specific) flow costs or (specific) flow emissions, for example, shortest path algorithms or well-established clustering techniques can be employed for further analysis. Additionally, alternative optimisation problem formulations based on the graph representation can be deployed. Hörsch et al. [37], for instance, decomposed their graph-modelled energy supply system into a spanning tree and closed cycles to create an alternative problem formulation better suited for their application.

2.2.2. Derived Visualisation Techniques

Visualising complex relationships efficiently and accessibly is crucial in facilitating the communication of these relationships among individuals. Consequently, dedicated sections follow, proposing approaches for visualising the potentially intricate relationships of ESSMs and their components.

Building on the idea of visualising ESSMs as graphs, as described in Subsection 2.2.1, two distinct approaches emerge. The first approach consists of a simple collection of circles or basic geometric shapes connected by uniform arrows, while the second is a multicoloured, visually dense encoded graph representation. These variations are labelled **Generic Graph** or **Generic System Visualisation** (GSV) and **Advanced Graph** or **Advanced System Visualisation** (ASV). Both versions are discussed in the subsequent paragraphs.

2.2.2.1. Simple ESSM Visualisation as Generic Graph

In this thesis, the **generic graph** or **generic system visualisation** (GSV) is the elementary form of displaying an energy supply system modelled as a graph. Each component is represented as a circular-shaped node, while the energy flow connections between the components are depicted as arrows pointing from the originating nodes to the targeted ones. This design corresponds to the mathematical formulation of a directed graph. Several design features have been incorporated into this GSV topology to enhance visual comprehensibility while keeping the information conveyed relatively low for quick human comprehension. Node visualisation features correspond to the aspects listed in Table 2.1, representing the available encoding combinations offered to the user via the respective drawing utility provided by the developed software framework.

In a generic system visualisation, nodes are represented by filled circular shapes (shape row in Table 2.1) and coloured according to one of the clustering options listed in the colour row of Table 2.1. Node size remains consistent for each component (size row in Table 2.1), while node positions are chosen to minimise subjectively perceived chaos (position row in Table 2.1). This approach often results in a hierarchical grid shape for larger energy system models with minimal overlapping edges. The automated hierarchical layout is invoked using the "Dot" option (bottom-right cell in Table 2.1). For non-hierarchical designs, a Fruchterman-Reingold [38, 39] force-directed layout algorithm is used by invoking by the "Neato" option (bottom-right cell in Table 2.1). Alternatively, the user can manually drag and drop node positioning or combine it with the automated positioning schemes.

Table 2.1.: Node Encoded Attributes of the Generic System Visualisation.

Attribute	Description	Options	Examples
Shape	Geometric Form	- Circles	Circle, Concentric Circles
		- Common Geometric Shapes	Rectangle, Octagon, Diamond
Colour	Circular Shape Fill Colour	- Component	Bus, Transformer
		- Energy Carrier	Wind, Electricity
		- Name	PV, Power Plant
		- Node Type	Storage, Fossil
		- Region	Germany, Europe
		- Sector	Power, Heat
Size	Circular Shape Diameter	- Default Size	Fixed to a Predefined Number
Position	Arrangement of Circular Shapes	- Non-/Hierarchical	"Neato", "Dot"
		- Manual Drag and Drop	

While the node design incorporates colour-encoded clustering mechanisms, the GSV edge design is kept simple, only visualising existing connections and their directions. An overview, including corresponding option examples, is provided in Table 2.2. Energy flows are depicted as uniform black-coloured arrows of the same width (colour, line style, and width rows in Table2.2). The length of these arrows does not encode additional information and is subject to node positioning (length row in Table 2.2).

Table 2.2.: Edge Encoded Attributes of the Generic System Visualisation.

Attributes	Description	Options
Colour	Arrow Shaft and Tip	Black
Length	Arrow Shaft Length	Subject to Node Positioning
Line Style	Shaft Line Style	Fixed to a Solid Black Line
Width	Shaft Line Width	Fixed to a Specific Number

Figure 2.10 presents an example ESSM visualisation, utilising the described GSV approach with an automated non-hierarchical node positioning scheme.

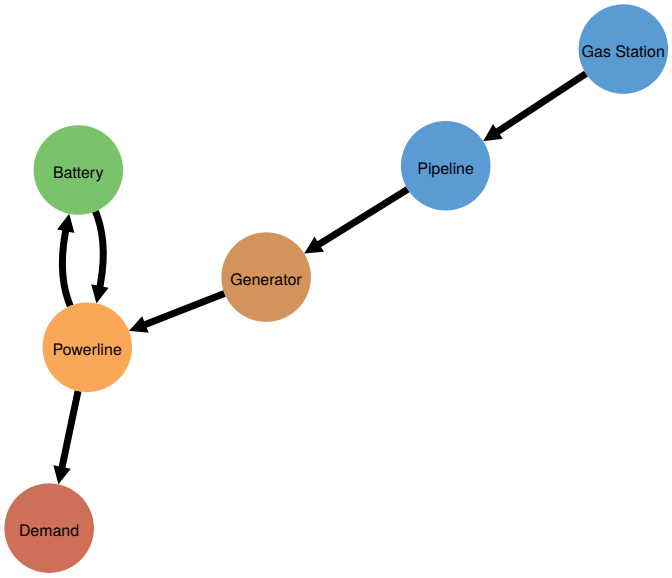

Figure 2.10.: Generic System Visualisation (GSV) Example.
Created Using the `tessif.visualize.dcgrph` Module.

The depicted ESSM is the "minimum working example" taken from *Tessif*'s example library [40]. Components are represented by colour-filled circular shapes of equal size (shape row in Table 2.1), stating the respective component name. The component colours are clustered by name (refer to the options column of the colour row in Table 2.1). In this particular example, the `Pipeline` and `Gas Station` components share a colour to highlight their unconstrained characteristic. Allowed energy flows and their directions are depicted using black-coloured arrows, where edge length, colour, or line style do not encode additional information (refer to Table 2.2).

2.2.2.2. Complex ESSM Visualisation as Advanced Graph

Building upon the GSV approach, the **advanced graph** or **advanced system visualisation** (ASV) technique incorporates post-processed optimisation results to create an informationally dense yet quickly comprehended figure. The ASV serves as the primary entry point for identifying components or clusters warranting further investigation and is designed for use in combination with the comparative result analysis for post-optimisation analysis. In contrast, the GSV is primarily, though not exclusively, intended for model formulation and pre-optimisation analysis.

The ASV utilises the same topology as the GSV, with components displayed as graph nodes and edges representing intercomponent energy flows. Additionally, several optimisation results are incorporated to enhance information density while maintaining a level of conveyed information that allows experienced users to identify and navigate to components of interest quickly.

The results chosen for visual encoding are summarised in Table 2.3. Not all encodings must be used simultaneously and are subject to user choice. The most frequently used attributes throughout this research are marked with an "x" in the "Default" column.

Table 2.3.: Node Encoded Attributes of the Advanced System Visualisation.

Attribute	Description	Options	Default
Shape	Geometric Shape	- circles	x
		- Common Geometric Shapes	
Colour	Geometric Shape Fill Colour	- Component	
		- Energy carrier	
		- Name	x
		- Node type	
		- Region	
		- Sector	
Size	Outer Diameter	- Proportional to Installed Capacity	x
Fill Size	Circular Shape Inner Diameter	- Proportional to Characteristic Value	x
Position	Arrangement of Circular Shapes	- Edge Length Dependent	x
		- Manual Drag and Drop	

In an advanced graph, node shapes can correspond to specific geometric shapes (shape row in Table 2.3) . By default, `Bus` components are represented as three concentric circles with increasing radius, filled with a transparent colour, of radius scaling transparency. This visualisation emphasises the variable capacity of `Bus` components, as their minimum and maximum energy

flow is unconstrained. Other component types are visualised as circular shapes with distinct borders, where border and fill colours are chosen based on one of the clustering choices listed in the options column of the colour row in Table 2.3. Node fill size represents a component's characteristic value[1], typically indicating partially filled nodes (refer to the fill size row in Table 2.3). The outer diameter of the displayed node depends on the component's installed capacity (refer to the size row in Table 2.3), while node positioning uses a Fruchterman-Reingold force-directed layout algorithm, or manual drag and drop to minimise the subjectively perceived chaos (refer to the position row in Table 2.3).

The ASV edge design can incorporate various visual encodings, though these are not mandatory and depend on user preference. The available visual encodings are summarised in Table 2.4.

Table 2.4.: Edge Encoded Attributes of the Advanced System Visualisation.

Attributes	Description	Options	Default
Colour	Arrow Shaft and Tip	- Proportional to Specific or Total Flow Emissions - Hex or RGB Colour Value	x (specific emissions)
Length	Arrow Shaft Length	- Proportional to Specific or Total Flow Cost - Manual Node Drag and Drop	(specific flow costs)
Line Style	Shaft Line Style	- Mapped to Intervals of Specific or Total Flow Cost - Manual Style Setting	x (dotted, dashed, solid)
Width	Shaft Line Width	- Proportional to Summed Up Loads	x

Edge colour, encompassing the arrow shaft and tip, ranges from light grey to black, depending on the specific emissions of the represented energy flow (refer to the colour row in Table 2.4). Higher emissions result in darker arrows. Edge length encodes the specific costs of the energy flow, with longer edges representing higher costs (refer to the length row in Table 2.4). An alternative approach divides the maximum occurring specific flow costs by three, visualising flows from the lowest third as dotted lines, those from the second third as dash-dotted lines, and the remaining flows as solid lines (refer to the line style row in Table 2.4). This distinction emulates line strength scaling with flow costs in the form of edge shape. Lastly, the ASV offers the option to scale edge width with the total summed loads calculated due to optimisation, widening high-energy transport arrow shafts, and narrowing low-energy transport shafts (refer to the width row in Table 2.4).

An example ESSM visualisation, utilising the described ASV pproach can be seen in Figure 2.11. This example again depicts the "minimum working example" alos used for demonstrating the GSV technique.

[1] A component's characteristic value typically corresponds to its capacity factor. More details can be found in Subsection 2.2.3.

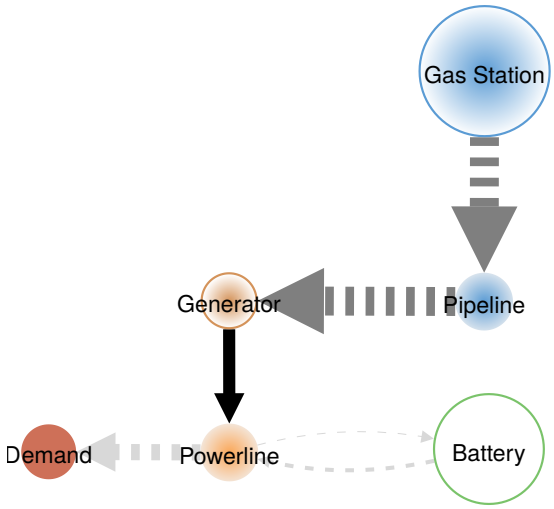

Figure 2.11.: Advanced System Visualisation (ASV) Example.
Created using `tessif.visualise.dcgrph`.

The nodes are illustrated as circular shapes, with colours clustered by name (refer to the options column of the shape and colour row in Table 2.3). The outer circle's diameter is proportional to the installed capacity. In contrast, the inner circle's diameter reflects the component's characteristic value (see the options column of the size and fill size row in Table 2.3). Node positioning was manually set using the drag and drop method (refer to the options column of the position row in Table 2.3).

The edges are visualised as arrows with a shaft and a single arrow tip. All default options in Table 2.4 are employed to encode the post-processed optimisation results visually. The edge colour, represented in greyscale, is proportional to the allocated specific emission value (see the options column of the colour row in Table 2.4). A higher specific emission corresponds to a darker grey shade. In Figure 2.11, the Generator to Powerline edge, thus, exhibits the highest specific emissions among all edges. The line style represents the allocated specific flow costs (refer to the options column of the line style row in Table 2.4). If the line is dotted, specific flow costs fall within one-third of the maximum occurring flow costs (applicable to all edges in this example except the Generator to Powerline edge). A dashed line signifies specific flow costs exceeding one-third but remaining below two-thirds of the maximum specific flow costs (not present in this example). A solid arrow shaft line indicates specific flow costs of at least two-thirds of the maximum occurring flow costs (true for the edge from the Generator to the Powerline component). The depicted shaft line width scales proportionally to the total energy flow between the connected components. A wider arrow shaft represents a larger total energy flow (see also the options column of the width row in Table 2.4). In Figure 2.11 the most energy is, thus, flowing from the Gas Station component to the Pipeline component.

2.2.3. Derived Result Data Representation Technique

Assigning the result data of a particular component to its node representation while assigning its interaction results to the respective edge representation yields a straightforward mapping approach. Table 2.5 depicts the selected node parameter and result mappings.

Component-specific values include the **installed capacity** P_{cap}, the **original capacity** (i.e. before optimisation) $P_{OrigCap}$, the **characteristic value** cv, the **current state of charge** $SOC(t)$ (if applicable), and descriptive specifiers such as **energy carrier**, **energy sector**, **region**, **node type** and

Table 2.5.: Overview of the Proposed Node Data Mapping.

Result Value	Symbol	Example	Type
installed capacity	P_{Cap}	10 MW	Result
original capacity	$P_{OrigCap}$	0 MW	Parameter
characteristic value	cv	0.8	Result
current SOC	$SOC(t)$	1 MWh	Result
energy carrier	-	Electricity	Parameter
energy sector	-	Power	Parameter
region	-	North Germany	Parameter
node type	-	High-Voltage Transmission Line	Parameter
component	-	bus	Parameter
latitude	-	53.46025	Parameter
longitude	-	9.969309	Parameter

component, as well as geospatial information in the form of **latitude** and **longitude** (refer to the result value and symbol columns in Table 2.5).

P_{cap} signifies the **installed capacity** post-optimisation, defined as the maximum gross outflow for a single-output transformer or source, maximum gross outflows for multiple-output transformers (e.g. CHPs), or the maximum state of charge for storage units. These values should be interpreted as results if they are subject to expansion. Otherwise, the **installed capacity** is predetermined and presented as a parameter.

The **characteristic value** cv is defined as $cv = mean(SOC)/P_{cap}$ for storages (with $mean(SOC)$ representing the average state of charge) and $cv = mean\left(F_{char}\right)/P_{cap}$ for all other components. F_{char} describes the component-specific interpretation of a characteristic flow as depicted by equation 2.3. The characteristic flow equals the out/inflow for source/sink components, the outflow for single-output (SO) transformers, and the outflow characterising the component's dispatch behaviour for multiple output (MO) transformers (e.g. heat output for a heat-driven district heating CHP).

$$F_{char} = \begin{cases} P(t)_{out} & \text{for sources and SO transformers} \\ P(t)_{in} & \text{for sinks} \\ P(t)_{out,char} & \text{for MO transformers} \end{cases} \tag{2.3}$$

Among the node specifiers, **region** describes an arbitrarily chosen string so that the user can group a set of components to, e.g. "North Germany". **Node type** allows users to input additional, optional information, such as "back pressure CHP" for a transformer component. The **component** specifier identifies one of *Tessif's* abstract components, as discussed in Section 3.2. This parameter aids post-processing, particularly when utilising supported ESSMOS lacking inherently unambiguous components.

Table 2.6 depicts the selected edge parameter and result mappings. Values mapped to the graph's edges, or component interconnections, encompass the **load** ($P(t)$), the outbound **net energy flow** ($\sum_t P(t)$), and the beforehand needed parameters **specific flow costs** c_{flow} and **specific emissions** (e_{flow}). For convenience and simplified programmatic implementation, **total flow costs** $\sum_t c(t)$ and **total emissions** $\sum_t e(t)$ are also mapped, despite being somewhat redundant (refer to the result value and symbol columns in Table 2.6).

Table 2.6.: Overview of the Proposed Edge Data Mapping.

Result Value	Symbol	Example	Type
load	$P(t)$	1 MW	Result
summed loads	$\sum P(t)$	10 MWh	Result
total flow costs	$\sum c(t)$	100 €	Result
total emissions	$\sum e(t)$	10 $t_{CO2\text{-}eq}$	Result
specific costs	c_{flow}	1 €/MWh	Parameter
specific emissions	e_{flow}	1 $t_{CO2\text{-}eq}$/MWh	Parameter

$P(t)$ represents an interconnection's current energy **load** (e.g. the amount of energy per unit of time). Conventionally, $P(t) > 0$ signifies a flow in the connection's direction, whereas $P(t) < 0$ indicates a flow in the opposite direction. $\sum_t P(t)$ describes an interconnection's **summed loads** across all optimised time steps, calculated for each edge in line with the a priori constrained arrow direction. Each sum is thus interpreted as the net energy outflow.

It is crucial to note that some parameters listed are predetermined and either directly or indirectly subject to user input. The advantage of uniformly representing and assigning results and parameters is the unambiguous and concise mapping of all values of interest through a single interface. Moreover, this representation can be easily generalised to any format capable of representing hierarchical key-value pairings. Consequently, this representation is A) result-generating software-independent and B) post-processing software-independent, rendering it a universally applicable technique.

It is also noteworthy that, despite its apparent simplicity, none of the examined ESSMOS in Chapter 5 offered such an interface. Results had to be inferred or manually pre-processed using often poorly documented functions, or they had to be gathered from different interfaces, sometimes necessitating the post-processing of raw data output. **It is highly recommended to consider the proposed result data representation technique for any newly developed ESSMOS.** It has proven simple, robust, modular, easily extensible, and raw data format-independent.

2.3. Statistical Tools Leveraged for the Identification of Significant Differences

Section 2.3 briefly introduces the statistical tools used for proposing a method to automate or at least support the process of identifying significant differences as described in Subsection 4.5.3. Wilckens [41] selected these tools purposefully and described them in sufficient detail. Hence, only the minimum sensible amount is restated here to give an English layout that provides the complete toolkit to comprehend and utilise the developed method.

2.3.1. Auxiliary Statistical Values Used for the Leveraged Tools

To describe the used Pearson correlation in Subsection 2.3.2, additional statistical values are introduced first. These include arithmetic mean, variance, standard deviation and covariance.

2.3.1.1. Arithmetic Mean

The arithmetic mean $\bar{x}$ can be described using equation 2.4:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{2.4}$$

Where x_i refers to the observed value i and n to the number of values in total [42].

2.3.1.2. Variance and Standard Deviation

The variance describes the mean square deviation of the observed values to the arithmetic mean. It serves as a measurement of dispersion in the sense of how far a set of values is spread around their average. The variance σ^2 can be obtained using equation 2.5:

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \tag{2.5}$$

Where $\bar{x}$ describes the arithmetic mean from above, x_i the i-th observed value and n the total number of values.

Due to the squaring in equation 2.5, variance and arithmetic mean have different units associated. Thus, the square root is taken to obtain the standard deviation of the same dimension, as described in equation 2.6. [43]

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{2.6}$$

2.3.1.3. Covariance

The covariance describes the joint variability of two variables. For two variables, x and y, the difference between their respective arithmetic means $\bar{x}$ and $\bar{y}$ is calculated, multiplied for each $i \in [1, n]$ for all of the n values, summed up and then divided by $n-1$ as described in equation 2.7. [44]

$$cov_{x,y} = \frac{1}{n-1} \sum_{i=1}^{n} ((x_i - \bar{x}) \cdot (y_i - \bar{y})) \tag{2.7}$$

The covariance is positive in case greater values of x mainly correspond with greater values of y and vice versa, i.e. in case the variables behave similarly. In the opposite case, however, where lesser values of x mainly correspond with greater values of y and vice versa, covariance results become negative. Hence the covariance shows the tendency of a linear relationship between x and y. It doesn't, however, yield reliable information about the magnitude or degree of the linear relationship. [45]

2.3.2. Measuring Linear Relationship Using the Pearson Correlation Coefficient

The Pearson correlation coefficient (PCC) measures the linear relationship or the linear correlation between two data sets. It can be expressed as the normalised covariance and is calculated as [46]:

$$PCC = \frac{cov_{x,y}}{\sigma_x \cdot \sigma_y} \tag{2.8}$$

$$PCC = \frac{\frac{1}{n-1}\sum_{i=1}^{n}\left((x_i - \bar{x}) \cdot (y_i - \bar{y})\right)}{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.9}$$

Due to the normalisation, the PCC can assume values between $[-1, 1]$. Its sign indicates the orientation or direction of the linear relationship as for the covariance. The PCC's absolute magnitude describes the degree of the linear relationship. [43, 46]

To evaluate the degree of a linear relationship, the information in table 2.7 is used as proposed by Kuckartz et al. [47]:

Table 2.7.: Assumed Degree of Linear Relationship. Evaluated Using the Pearson Correlation Coefficient.

Absolute value of PCC	Degree of Linear Relationship
$\in [0.00, 0.10[$	No
$\in [0.10, 0.30[$	Low
$\in [0.30, 0.50[$	Medium
$\in [0.50, 0.70[$	High
$\in [0.70, 1.00]$	Very High

2.3.3. Modelling Errors Used for the Developed Statistical Identification Technique

The following subsection introduces the basics of the used statistical modelling error analysis. Although only the normalised mean absolute error is used within this thesis, and the case study in particular, all error values seen as potentially fit, are briefly presented in the following. These error values are also implemented and available within the developed software framework. The statistical analysis discussed in Paragraph 4.5.3.1, can thus be readily deployed using each of the below error values.

2.3.3.1. Mean Modelling Error (MME)

The pairwise statistical comparison of a model predicted value $x_{pre,i}$ and an observed or reference value $x_{obs,i}$ for $i = 1, 2, \ldots, n$ is a commonly used method in engineering science [48]. In case the model deviation e_i is defined as $e_i = x_{pre,i} - x_{obs,i}$, then the mean modelling error $\bar{e}_\tau$ (MME) can be generally described using the following equation:

$$\bar{e}_\tau = \sqrt[t]{\sum_{i=1}^{n} w_i |e_i|^\tau \Big/ \sum_{i=1}^{n} w_i} \ . \tag{2.10}$$

where $\tau \geq 1$ describes different kinds or errors and w_i an interval-specific weight factor. This weight factor is, in particular, relevant for non-equidistant intervals. In this work, however, all intervals occurring are of the same length or distance, i.e. $w_i = 1$. For this reason, w_i is neglected in the following. Equation 2.10 thus simplifies to:

$$\bar{e}_\tau = \sqrt[t]{\frac{1}{n} \cdot \sum_{i=1}^{n} |e_i|^\tau} \tag{2.11}$$

The MME itself is not directly used in any error value calculation. It is a basic template from which the following applied error values are developed.

2.3.3.2. Mean Absolute Error (MAE)

For $\tau = 1$, the mean absolute error (MAE) is derived from the mean modelling error (equation 2.11) when using the absolute error value $|e_i|$ as described in equation 2.12 [48]:

$$\mathrm{MAE} = \frac{1}{n} \cdot \sum_{i=1}^{n} |e_i|. \tag{2.12}$$

The MAE describes the absolute deviation of the predicted result values from the observed averaged out over the investigated dataset.

2.3.3.3. Mean Biased Error (MBE)

For $\tau = 1$, the mean biased error (MBE) is derived from the mean modelling error (equation 2.11) when not using the absolute error value as described in equation 2.13 [48]:

$$\mathrm{MBE} = \frac{1}{n} \cdot \sum_{i=1}^{n} e_i. \tag{2.13}$$

The MBE describes the tendency of the predicted result values are overestimate ($\mathrm{MBE} > 0$) or underestimate ($\mathrm{MBE} < 0$) the observed value in average. It is worth noting that an MBE of zero does not imply perfect prediction. Over- and underestimations could potentially just cancel each other out.

2.3.3.4. Root Mean Square Error (RMSE)

For $\tau = 1$, the mean biased error (MBE) is derived from the mean modelling error $\bar{e}_\tau$ as described in equation 2.14 [48]:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} e_i^2}. \tag{2.14}$$

The RMSE is a commonly used metric for examining modelling errors [48, 49, 50]. Large deviations, however, are weighted heavier due to error quadrature. A few large errors may result in a high RMSE, whereas the mean absolute error may be relatively low overall [51].

2.3.3.5. Error Normalisation for Improved Comparability

Several normalisation techniques are proposed below to facilitate the comparison of modelling error results between ESSMOS, as demonstrated throughout this thesis. These techniques apply to all previously mentioned error values (EVs), yielding a subsequent set of normalised error values (NEVs). The NEVs described below are implemented and available in the software framework used for this thesis' statistical analyses. The primary advantage of using normalised EVs is their independence from magnitude, enabling the comparison of time series results of different scales.

Maximum Observed Value Freeman et al. [49] suggest normalising EVs using the maximum observed value $\max(x_{obs})$ as shown in equation 2.15:

$$NEV = \frac{\mathrm{EV}}{\max(x_{obs})}. \tag{2.15}$$

This is especially useful when comparing time series results where the average value and maximum spread differ substantially, for example, when evaluating PV load results.

Maximum Observed Spread When using the maximum observed spread as normalisation, equation 2.15 changes to:

$$NEV = \frac{\mathrm{EV}}{\max(x_{obs}) - \min(x_{obs})}. \tag{2.16}$$

In cases where the maximum observed value $\max(x_{obs})$ is significantly different from the maximum observed spread $\max(x_{obs}) - \min(x_{obs})$ due to large negative or large minimum positive values, it can be sensible to use the maximum spread instead for normalisation.

Observed Arithmetic Mean When using the average observed value $\bar{x}_{obs}$ for normalisation, the NEV is calculated as shown in equation 2.17 below:

$$NEV = \frac{EV}{\bar{x}_{ref}}. \tag{2.17}$$

The main advantage of using the arithmetic mean of the observed value $\bar{x}_{obs}$ over the maximum observed value $\max(x_{obs})$ or the maximum observed spread $\max(x_{obs}) - \min(x_{obs})$ is that of reducing the impact of rare outliers. With the above two normalisation options, one large observed value would lead to a large denominator and, thus, small error values. Using the arithmetic mean, however, this impact is much lower, making it a useful general-purpose normalisation.

2.3.4. Utilising Modelling Errors as a Measurement of Deviation

As described in the literature [52, 53], statistical error analysis is commonly employed in modelling tasks involving observable variables, making it relatively straightforward to determine which values to use as predicted (x_{pre}) and observed (x_{obs}). However, in the context of comparing ESSMOS tools, reliable large-scale validation cases have yet to be generated, against which modelling attempts could be tested. Consequently, the observed or reference value is replaced by the results of one of the ESSMOS tools. This substitution effectively redefines the modelling

error as a deviation of one or multiple results from different ESSMOS concerning a single ESSMOS, leading to the following core statement about the statistical analyses conducted in this thesis:

In this work, modelling errors, in the statistical sense, are utilised to measure result deviation for one or multiple ESSMOS regarding a single ESSMOS.

Thus, the value labelled above as observed value $x_{obs,i}$ is interpreted as the reference result value x_{ref_i} or REF_i, where a specific ESSMOS is selected as the reference. Simultaneously, $x_{obs,i}$, labelled above as the predicted result, is considered the result value $x_{res,i}$ or RES_i to be compared. From the EVs discussed in Subsection 2.3.3, the MAE, normalised by the arithmetic mean, is used as a measure of deviation. This selection is somewhat arbitrary but found to be the most intuitive. Thus, the utilised measure of deviation is calculated as the normalised mean absolute error or deviation (NMAE or NMAD) described in equation 2.18:

$$\mathrm{NMAE} = \mathrm{NMAD} = \frac{1}{n} \cdot \sum_{i=1}^{n} \mathrm{abs}\left(RES_i - REF_i\right) \Big/ \mathrm{mean}(REF) \tag{2.18}$$

Where "abs" denotes the absolute value and "mean" describes the arithmetic mean. The chosen reference ESSMOS results are labelled REF, whereas the remaining ESSMOS results are specified as RES.

3. Developed Software Framework for Transforming Energy Supply System Models

The *Transforming Energy Supply System Modelling Framework* (*Tessif*) is a software framework developed within the context of this thesis. The development directly addresses goal 2a and aims to enhance and advance the field of modelling and optimising energy supply systems.

This chapter is separated into eight sections. First, the motivation for creating such software is explored, drawing upon current literature. Second, Tessif's ESSM component templates are introduced. Third, the design and creation process of an ESSM formulated in *Tessif* is discussed. Fourth, an overview of *Tessif*'s general programmatic and functional design is provided. The chapter proceeds with a fifth section, which elaborates on the conceptual distinction between *Tessif* and other ESSMOS. *Tessif*'s applicability beyond this thesis's scope is examined in the sixth section. The seventh section addresses further development and longevity as free and open-source software. Finally, the eighth section presents the model scenario combinations created through *Tessif*, showcasing the comparative method in Chapter 5.

3.1. Motivation for Developing Another Free and Open Source (FOS) Energy Supply System Modelling and Optimisation Software (ESSMOS)

The following section provides an in-depth discussion about the reasoning behind developing yet another FOS ESSMOS. It is both separate and connected to the general motivation of this thesis. However, it was found beneficial to highlight the logical synergies between the Comparative Method and the developed software Tessif. The motivation discussion is thus following this dual-fold approach and is separated into the general thesis motivation laid out in Section 1.1 and the following.

The comparative analysis of ESSMOS can be split into two aspects. One is the comparison of software-specific characteristics, and two is a result-based comparison. The formulated targets in 1.2 are therefore split into a priori and result-based parts.

Conolly et al. [33] describe the advantages of this kind of software comparison and how it facilitates the beneficial application of these software tools. Their comparison of 68 initial and 37 final tools aims to aid users in their selection process. They compare, among other aspects, the software tools' user count and availability. They also compare criteria such as applicable energy sectors and the modelling approach, differentiating supply-focused and demand-focused. Furthermore, software tools are distinguished between aiming at simulating and optimising a given ESSM.

For similar reasons, Ringkjøb et al. [4] compare 75 software tools in total. While also including programming language, software architecture and available energy supply system com-

ponents. Furthermore, they also compare the mathematical approach/formulation chosen by the individual software.

Ommen et al. [54] focus on the impact of the mathematical formulation approach. They show that accuracy increases from LP to MILP to NLP formulations and that the MILP approach represents the best compromise given the required computational time. This finding is one of the reasons that all but one of the compared ESSMOS in Chapter 5 allow some sort of MILP formulation, with some having the option of transforming NLP to MILP formulations using partwise linearisation.

Although not directly comparable, Wilkerson et al. [55] perform a result-based comparison of three integrated assessment modelling software tools to evaluate carbon policy impacts on energy supply system infrastructure. Showing that results may differ considerably, indicating the same might be expected from energy supply system modelling software.

Priesmann et al. [56] compare 160 power system modelling software results based on varying complexity of the modelled energy system. Testing the hypothesis of a positive correlation between complexity and accuracy, they conclude that a certain degree of complexity yields higher accuracy. However, they also propose carefully balancing problem complexity to obtain acceptable use of computational resources. To conduct their research, they developed their own testing framework to facilitate alternate model formulation, thus indicating the benefits of dedicated software to perform comparisons, resulting in the formulation of Goal 2a to create such software as part of the underlying research.

Since compared software tools often only share a subset of ESSM parameters, Misconel et al. [57] highlight the importance of harmonised data input to make software results comparable. They conclude that harmonising data input is beneficial but time-consuming and potentially challenging. This result leads to their conclusion that comparing less complex ESSMs based on such datasets might be more feasible. Hence the ESSMOS compared in Chapter 5 aim to strike a balance between complexity and accuracy, as recommended by Priesmann et al. [56].

Steinbrink et al. [58] evaluate the benefits of software comparison in the context of modelling smart grids. They conclude that for (relatively complex) ESSMs, automated creation of such models for each of the compared software tools would be advantageous. They highlight the benefits of checking the different software results for plausibility since laboratory-scale validation for systems (and hence their ESSMs) is not feasible on the scale these systems are currently applied. They also conclude that such automated model creation would open the possibility of real-time simulations/optimisations. Using the recommendations from Misconel et al. [57] and Steinbrink et al. [58], Tessif is thus designed to provide harmonised model data input which is then used to create software-specific ESSMs automatically.

Gils et al. [5] follow these recommendations also and compare nine power sector modelling software tools. Their comparison utilises harmonised input data and highly simplified ESSM configurations to isolate and quantify software-specific effects. The respective ESSMs are created using an automated approach. In their conclusion, they encourage software developments utilising a harmonised data input approach in combination with automatic ESSM creation for further similar software tool comparisons.
Furthermore, they perform a result-based comparison, using the total amount of energy transformed as an indicator to identify significant differences, which are then investigated in more detail using the respective load profiles. For that reason, Goals 2c and 2d were formulated, and a guided strategy for comparative post-processing is proposed as part of this thesis.

Summarising the above, it gets clear that an automated comparison of different ESSMOS using

harmonised data input, allowing for uniform, concise and unambiguous result data representation, auxiliary visualisation, and further post-processing coupled with a guided strategy to compare the software based on their results could be of great benefit to researchers, policymakers, and the modelling community. For that reason, the author developed the software *Tessif* at the *Institute of Energy Systems*, part of the *Hamburg University of Technology*, attempting to address all of the above and the auxiliary goals formulated in 1.2.

Currently, many software tools are available to model energy supply systems, with the *Open Energy Modelling Initiative (openmod)* listing 80+ FOSS tools [59] and Ringkjøb et al. comparing 76 software tools [4]. Given this variety, users can benefit from guidance on selecting the appropriate software for specific purposes and, since results may vary, employing multiple tools to obtain a range of potential solutions. Consequently, *Tessif* is developed, inter alia, to facilitate and streamline the comparative process established in this thesis.

3.2. Tessif's Energy Supply System Model (ESSM) Component Templates

Adhering to the fundamental modelling principle of clustering logically connected functionalities into reusable parts [6], *Tessif* organises common energy supply system entities into distinct component templates. These templates are parameterised explicitly during the ESSM creation process to generate the actual components (or component-objects). These modular templates are easily extensible and, subject to practical constraints, interconnectable. *Tessif* is designed to streamline the extension process by enabling the seamless integration of additional component templates, provided they adhere to its specific architectural design scheme.

3.2.1. Tessif's Bus Component Template

Components created via *Tessif*'s `Bus` template are defined as balance objects. They are characterised by their non-interfering behaviour towards their in and outflows. They neither change the total sum of in- and outflows nor the flows' carriers. They may, however, aggregate or split inflows and outflows, provided the total balance is not altered. Examples would be nodes as described in Kirchoff's Law or y-pieces in a pipeline system. It is important to emphasise that `Bus` components do not modify any carrier type. They would thus be suited for connecting (potentially multiple) water sources and sinks, but not, for example, an air hose and a water sink.

3.2.2. Tessif's Source Component Template

Components created via *Tessif*'s `Source` template are defined by having only outflows and no inflows, increasing the overall amount of energy present in the system. The `Source` template provides several interfaces for describing or constraining outflow characteristics. The most common and relevant include:

- Constraining minimum and maximum output at each time step, allowing the solver to determine optimal dispatch

- Restricting minimum and maximum output at each time step to a single value, potentially varying with each time step, enabling the modelling of externally constrained outflows, such as solar panels
- Limiting the total output amount across the entire optimisation time span
- Assigning outflow-specific attributes, such as costs or emissions
- Restricting maximum positive or negative changes in outflow amounts between two time steps
- Permitting solver-determined increases in installed capacity (i.e., maximum outflow) and allocating corresponding capacity-specific expansion costs

Examples encompass renewable energy transformers, such as solar panels or wind turbines, commodities like fossil fuels, and boundary modelling components like power imports or balancing sources.

3.2.3. Tessif's Sink Component Template

Components created via *Tessif*'s `Sink` template are defined by having only inflows and no outflows. They can be comprehended as the opposite of source objects, decreasing the system's overall energy. Examples range from power or heat demands to boundary modelling components, like power exports or excess sinks. *Tessif*'s `Sink` template provides the same interface as the *Tessif* `Source` template, albeit describing the respective inflows.

3.2.4. Tessif's Transformer Component Template

Components created via *Tessif*'s `Transformer` template are characterised by relating input and output flows to each other in a one-directional manner. Possible contextualisation include:

- Aggregating or splitting inflows into potentially multiple outputs
- Multiplying inflows by so-called conversion factors to generate outflows
- Changing flow carrier types

Examples of `Transformer` components range from conventional power plants and heat pumps, over one directional electrical transformers or pressure stations, to lossy mass transfer. The core aspect of `Transformer` components is the `conversion_factors` mapping, which contextualises the component's input and output flows by an efficiency value. This efficiency can either be a singular value (e.g. 1.0) or a time series (e.g. $[0.9, 0.8, 1.0, \ldots)$. However, this mapping has to be stated before any optimisation. Beyond the `conversion_factors` mapping, the `Transformer` input and output flows can be characterised as a combination of sink inflows and source outflows.

3.2.5. Tessif's Storage Component Template

Components created via *Tessif*'s `Storage` template are characterised by relating a single inflow and outflow to an accumulated amount of stored energy. The most relevant and common relations include:

- Multiplying the inflow by an inflow efficiency to determine an increase in the stored energy amount
- Multiplying the outflow by an outflow efficiency to calculate a decrease in the stored energy amount
- Constraining the amount of stored energy between zero and a maximum capacity (which may be infinite)
- Describing idle changes in stored energy between time steps, independent of inflow or outflow
- Allowing solver-driven increases in installed storage capacity with corresponding capacity-specific expansion costs
- Limiting the expansion ratio of installed storage capacity and maximum input and output flow capacity

Examples encompass electrochemical energy storage systems, thermal storage systems, and any other energy storage types where input and output energy carriers are identical.

3.2.6. Tessif's Connector Component Template

Components created via *Tessif*'s `Connector` template are defined by relating two bidirectional flows to each other via an efficiency or loss coefficient. Flow direction is fixed at any particular time step. One flow is therefore characterised as inflow and the other as outflow. This allocation, however, may vary between time steps, effectively creating a bidirectional potential lossy connection between two `Bus` components. It is important to note that the multiplier is non-directional. Thus, for an energy flow P from component B to A, $P_A = \eta \cdot P_B$, whereas $P_B = \eta \cdot P_A$ for a flow from A to B.

In addition, connectors are considered active. Therefore, a potential energy flow occurring in a connected node has no direct consequence on the connector's behaviour. Therefore, the solver has to actively use the component and may choose not to. This is where the `Connector` component differs from the `Conveyer` component.

Examples range from power lines over gas pipelines to district heating pipelines. Especially in the case of actively regulated instances of these types. Since connectors are, in most cases, more straightforward to constrain in an optimisation context, they are often utilised over the `Conveyer` component. They, however, do not model lossy flow correctly since the actual efficiency or loss factor is usually dependent on the amount of flow occurring. For the examples above, passive components are, thus, better suited since they allow a flow-related loss modelling approach.

3.2.7. Tessif's Conveyer Component Template

Components created via *Tessif*'s `Conveyer` template were initially designed to model passive energy transport phenomena and to serve as the counterpart to the active `Connector` component. Due to developmental delays, however, it is not incorporated into *Tessif's* initial release. Further information can be found in the Outlook Subsection 7.2.3 and the Appendix Section A.11.

3.3. Tessif's ESSM Design and Creation Process

Tessif's ESSM design is based on Pfenniger et al.'s [60] description and the theoretical foundations discussed in Subsections 2.1.1 and 2.1.2. Consequently, *Tessif* models its ESSM as a collection of energy supply and demand entities, interconnected by transport entities, in combination with energy transformation and energy storage entities. The generic representation of a small example ESSM in Figure 2.1 is reused here and illustrated in Figure 3.1 using *Tessif*'s visualisation capabilities. *Tessif*'s components are depicted as green circles. These components are created by explicitly parameterising *Tessif*'s component templates and correspond to the beforenamed collection of entities. The parameterisation process of the component templates also results in the explicit formulation of possible energy flow connections, visualised as black arrows indicating the flow direction.

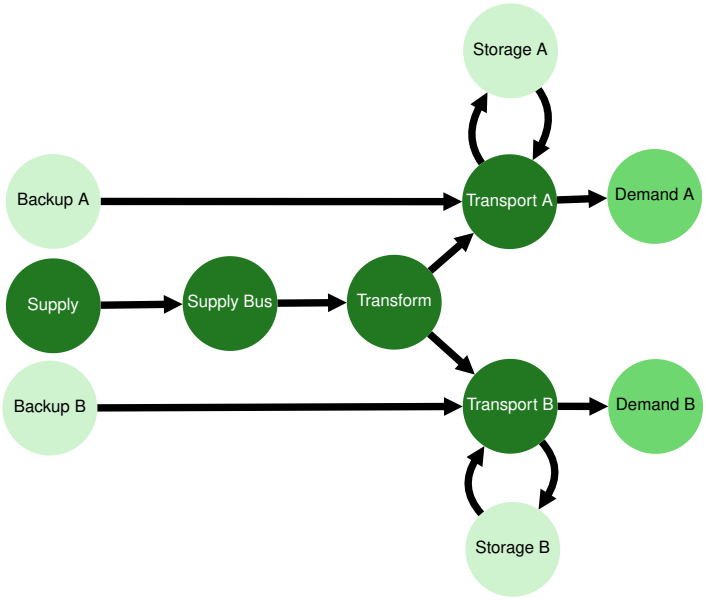

Figure 3.1.: Example Representation of a *Tessif* ESSM Created by Explicitly Parameterising *Tessif*'s Component Templates. The Depicted ESSM is the Same as the Generic Interconnection Representation in Figure 2.1. The Visualisation Was Created Using the `tessif.visualize.dcgrph` Module.

Given this component-focused modular design, *Tessif* thus leans towards a bottom-up model architecture. According to Farzaneh et al., it is, therefore, likely to represent an optimistic approach. However, given the somewhat free interconnectivity in combination with very few required parameters, a top-down modelling approach is also possible.

A *Tessif* ESSM serves as harmonised data input interface. No explicit mathematical constraints are formulated, and no optimisation model is created. These are only constructed when the *Tessif* ESSM is transformed into its supported ESSMOS ESSMs. Hence, the interface characteristics matter most in a *Tessif* ESSM. Consequently, it is tailored towards the use by engineers and how they would parameterise and set up an ESSM. Therefore, it is very liberal concerning the amount of required exogenic constraints while simultaneously offering many optional parameters. This approach provides the primary benefit of being able to cover most, if not all, common ESSM modelling tasks while at the same time being relatively intuitive to use by engineers. It, however, comes with the significant drawback of not being able to check for plausible user-created ESSMs. Thus, sensible parameterisation depends on the specific ESSMOS used and is to be checked

during transformation, increasing the complexity of the already complex task of ESSM data transformation. This design approach requires the user to use visualisation and small extension increments during complex ESSM creation in combination with frequent test runs.

In *Tessif*'s actual code, an ESSM is usually called `energy_system` or `es` and an object of the `AbstractEnergySystem` class residing within the `energy_system` module, part of the `model` subpackage.

3.4. Tessif's General Programmatic and Functional Design

As detailed in Section 3.1, *Tessif* aims to be a utility-based, practical-oriented tool to help compare existing ESSMOS. It is therefore designed to be a framework offering a core set of capabilities and a meticulously created interface for additional software integration. *Tessif* abstracts as much functionality as possible, which makes it independent of the ESSMOS supported. At the same time, *Tessif* synergises well with its integrated ESSMOS by enabling seamless data transformation.

Figure 3.2 displays *Tessif*'s design concept and how the internal data flows connect *Tessif*'s various utilities as three separate layers. The "Design Layer" represents *Tessif*'s most abstract and general user perception. Data input is provided, processed by *Tessif*, and returned to the user as output.

The "Modelling Process Layer" in Figure 3.2 represents the typical software-aided modelling process, applicable to a broad range of software modelling and optimisation tools. *Tessif* preprocesses the provided ESSM data input to execute the modelling and optimisation process using a dedicated third-party tool (depicted in Figure 3.2 as "ESSMOS" trapezoid). The result is post-processed and provided to the user via distinct disk writing and visualisation utilities or as raw ESSMOS output.

The "Data Flow Layer" in Figure 3.2 depicts the most explicit representation of *Tessif*'s design, visualising the actual data flow. The data input can be provided as ESSMOS-specific ESSM data or as *Tessif*-Specific ESSM data (visualised by the two bottom left green parallelograms in Figure 3.2). The prerequisite for these different inputs is that the respective `ESSMOS-name` modules holding the parsing codes are implemented inside the `tessif.transform.mapping2es` and the `tessif.transform.es2es` subsubpackages (depicted as white boxes of the same name in Figure 3.2) where `ESSMOS-name` is to be interpreted as a placeholder for the actual ESSMOS tool's name in lowercase letters, like `fine` or `pypsa`. Currently, only an `oemof` module is implemented for the `tessif.transform.mapping2es` subsubpackage, where as for the `tessif.transform.es2es` subsubpackage individual modules for all ESSMOS compared in Chapter 5 are implemented.
The respective data transformation module output is then passed to the `tessif.optimize` subpackage[1], which invokes the ESSMOS-specific optimisation capabilities (depicted as a white box of the same name in the lower centre of Figure 3.2). The optimisation results are then post-processed by the ESSMOS-specific `tessif.transform.es2mapping` module or immediately returned to the user. If the results are post-processed, they are either directly returned to the user or further processed by the `tessif.identify` or `tessif.visualize` subpackages (depicted as white boxes of the same name in Figure 3.2).

[1] The software framework *Tessif* adheres to North American spelling.

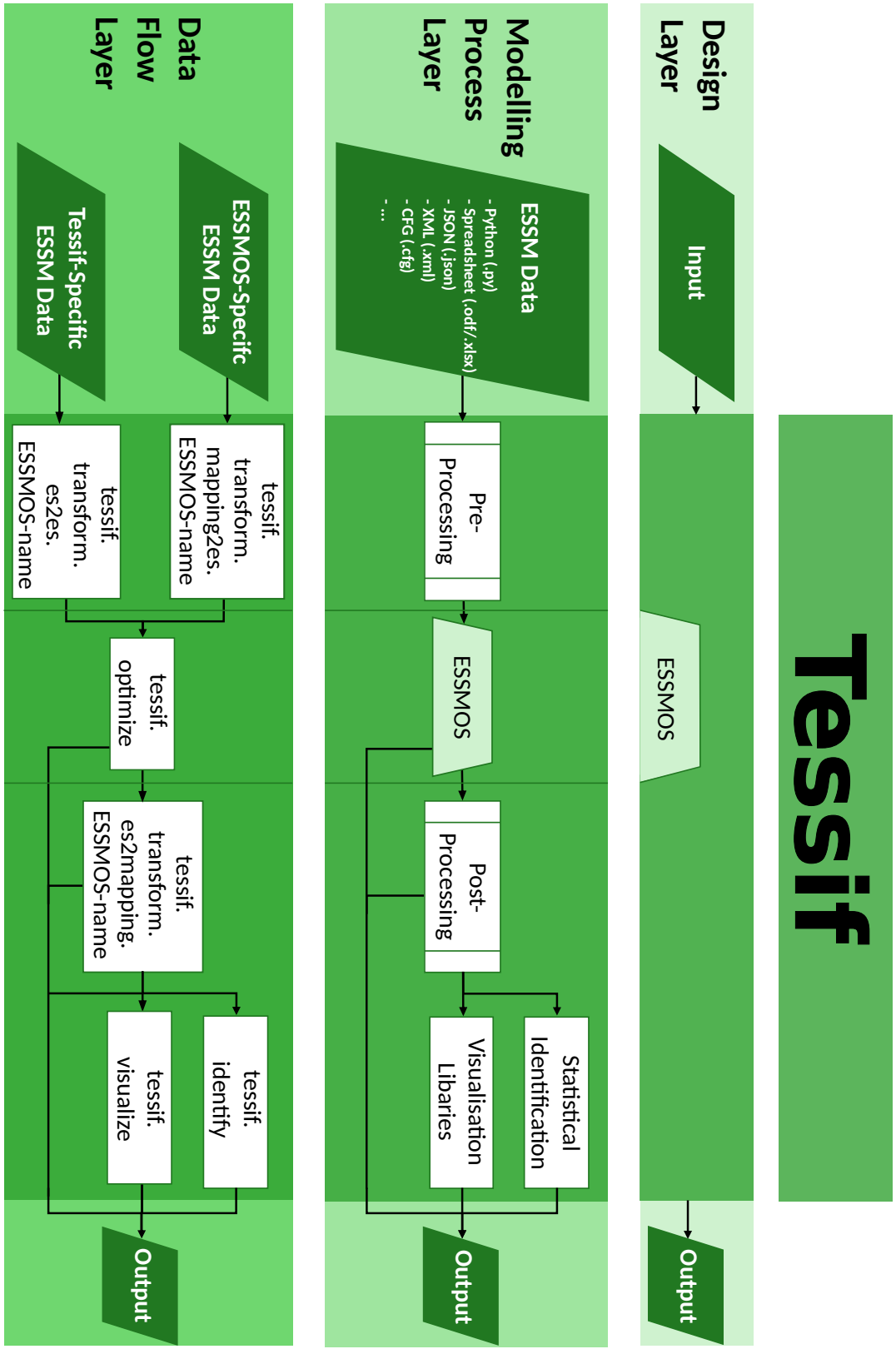

Figure 3.2.: *Tessif*'s Design and Data Flow as an Energy Supply System Modelling and Optimsation Software (ESSMOS).

3.4.1. Relevant Tessif Subpackages Used for Executing the Comparative Method

Tessif's core functionalities are divided into individual subpackages to enhance maintainability and reusability. A detailed elaboration of all *Tessif* subpackages is available in *Tessif*'s online documentation [61]. The relevant subpackages used during the case study in chapter 5 to showcase the Comparative Method are visualised in Figure 3.3 and discussed in the following. Figure 3.3 is thereby a reiteration of the Data Flow Layer depicted in Figure 3.2 with a shifted focus on how the actual ESSM data is processed and transformed into different states using *Tessif*'s individual subpackages.

3.4.1.1. Subpackage Tessif-Transform

The `tessif.transform` subpackage encompasses *Tessif*'s core data handling and abstraction mechanisms, representing the main interface for the integrated ESSMOS. This is depicted both explicitly and implicity in Figure 3.3. the large green rectangles situated on the left and right respectively denote the `tessif.transform` interface, while its individual subpackages, `es2es` and `es2mapping`, prefixed with `tessif.transform`, are presented as correspondingly labelled white boxes.

The green rectangles thereby emphasise the two-fold design of the `tessif.transform` subpackage, which serves as an interface both to transform the harmonized data input using a *Tessif*-specific ESSM to the ESSMOS-specific ESSM data format, and to revert the ESSMOS-specfic ESSM data including the optimisation results back into a harmonised output format. The subpackage, thus, governs both the pre-processing and post-processing operations.

In the pre-processing stage, the `tessif.transform` subpackage processes the *Tessif*-specific ESSM data. This *Tessif*-specific data is transformed to the ESSMOS-specific ESSM-data representation by the individual `ESSMOS-name` modules of the `tessif.transform.es2es` subsubpackage. In Figure 3.3 this ESSMOS-specific ESSM-data representation is visualized by the four green, vertically aligned parallelograms on the left.

The post-processing phase involves the transformation of the ESSMOS-specific ESSM and result data by the dedicated `ESSMOS-name` modules of the `tessif.transform.es2mapping` subsubpackage. The output of this transformation is *Tessif*'s result data representation as developed in Subsection 2.2.3. In Figure 3.3 this result data representation is depicted by the white parallelogram.

3.4.1.2. Subpackage Tessif-Optimize

The subpackage `tessif.optimize` aggregates the utilities to provide so-called "single-push-button" functions to execute ESSMOS-specific optimisation routines. It is depicted as the large light green rectangle in between the `tessif.transform` rectangles in Figure 3.3. Several functions are implemented for each ESSMOS, offering different interfaces to accommodate the most common data input methods. These functions take an ESSMOS-specfic ESSM object as input, solve the underlying optimisation problem using the ESSMOS and return the ESSMOS-specific ESSM object containing the optimisation results. In Figure 3.3 these ESSMOS tools are depicted as four green, vertically aligned rectangles, whereas the ESSM objects containing the optimisation results are visualised by the four green, vertically aligned parallelograms on the right. The design choice to incorporate the result data into the ESSM object necessitates rudimentary

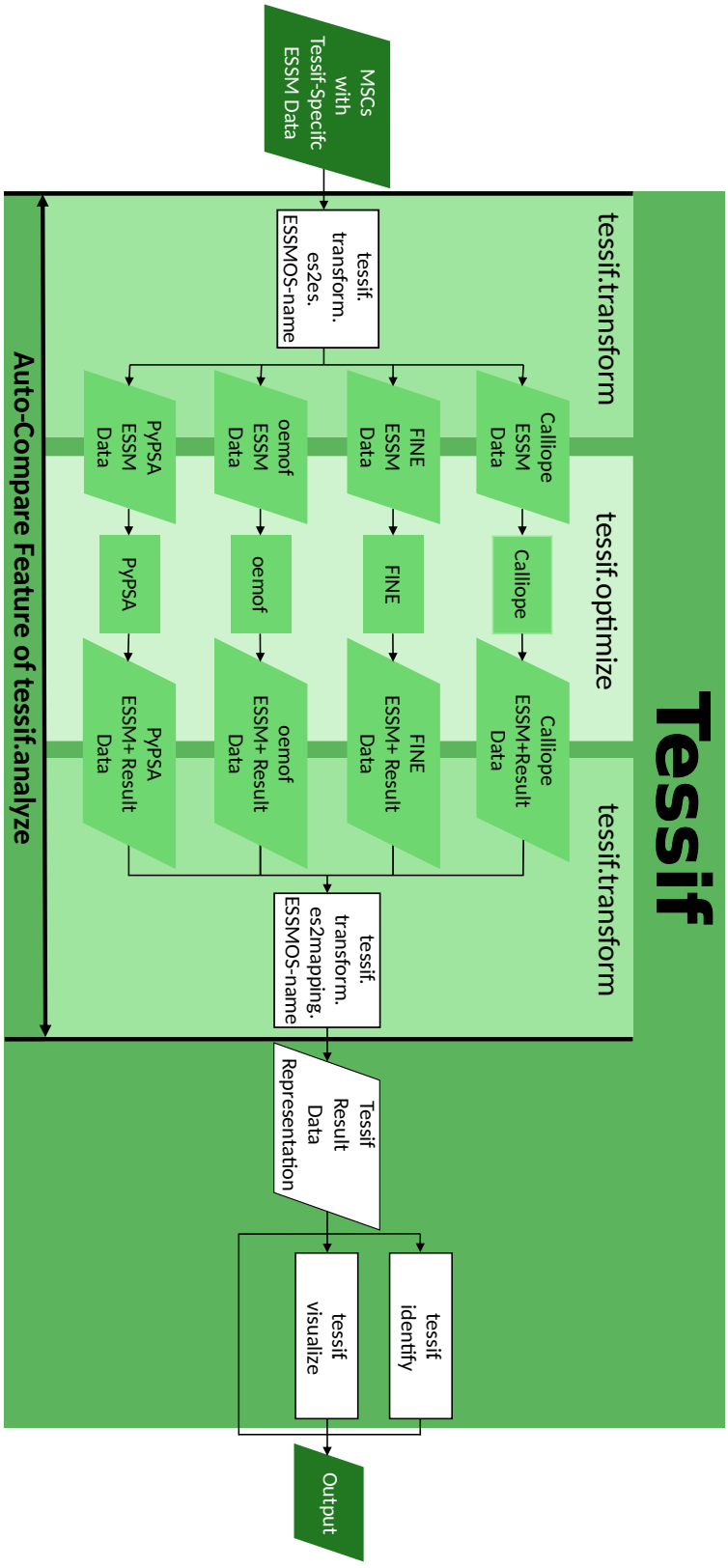

Figure 3.3.: Explicit Data Flow as Occuring Through *Tessif* During the Execution of the Comparativ Method Including all Intermediate Data Format Representations.

post-processing to construct the return object, utilising the respective ESSMOS interface or Python introspection tools.

3.4.1.3. Subpackage Tessif-Analyze

The subpackage `tessif.analyze` facilitates the actual comparison process by providing a simple auto-comparison feature for executing the entire transformation and optimisation chain. The auto-comparison feature is depicted in Figure 3.3 as the two vertical black lines connected by the double-headed black arrow at the bottom of the figure representing the aforementioned chain. The auto-compare feature requires an MSC using the *Tessif*-specifc ESSM data as input in combination with a container of strings representing the different ESSMOS to be compared. This feature is particularly useful for users unfamiliar with *Tessif*, as it automates the whole process of creating the *Tessif* result data representation as indicated in Figure 3.3. All of the initial case study results in chapter 5 were generated using this auto-compare feature of the `tessif.analyze` subpackage.

3.4.1.4. Subpackage Tessif-Identify

The `tessif.identify` subpackage for statistical identification provides one of the core features of the comparative result analysis by implementing the techniques for identifying significant differences, as proposed in Subsection 4.5.3. It is depicted as white box in Figure 3.3 and specifically consists of statistical detection tools for automatically identifying relevant differences in time-varying and static results. Its applicability is shown in the case study, in Subsection 5.5.4, building on the theoretical foundations discussed in Section 2.3. Beyond the exact parameterisation shown in this thesis, it offers the ability to use any arbitrary threshold, error value or correlation technique. It, therefore, provides excellent usability even beyond the scope of this work.

3.4.1.5. Subpackage Tessif-Visualize

The `tessif.visualize` subpackage depicted as white box in Figure 3.3 provides the necessary tools to create the most commonly used visualisations when working with *Tessif*. All of the general and advanced system visualisations, as well as most bar charts drawn throughout this thesis were created using the `tessif.visualize` subpackage.

3.5. Differences Between Tessif and Other ESSMOS Tools

Tessif is very distinct from most modern ESSMOS in that it does not include a separate optimisation problem solver interface. Thus, a *Tessif* ESSM cannot be optimised without prior transformation into one of its supported ESSMOS. This is not a design prerequisite. However, it distinguishes *Tessif* and reveals its primary purpose: The possibility of providing harmonised data input and data output processing tools. These capabilities may be used in combination with the presented Comparative Method, but their usefulness has proven to reach far beyond. So much so that prior *Tessif* users often return using *Tessif* just to visualise or analyse a particular ESSM and use *Tessif* alongside other modelling projects.

In essence, *Tessif* is to be mainly interpreted as a framework capable of transforming energy supply system modelling data.

It is imperative to bear in mind that the *Tessif* ESSM representation, alongside the ESSM representation of all other ESSMOS, are created using the building blocks provided by the respective software. The abstract idea of the user's ESSM is, therefore, created by a set of features provided by each software. A change in a given ESSM is, thus, typically achieved by changing the composition of these features and does not require any core software code modifications. Therefore, the respective ESSM representation is to be seen as an associated but separate entity regarding the ESSMOS it was created with. Applied to the context of this thesis, this also implies that the abstract idea or concept of the developed Comparative Method is an associated but separate entity from both *Tessif* and the ESSMs created with *Tessif*. And indeed, the described method can be executed entirely independent of *Tessif*. However, *Tessif* significantly streamlines and facilitates this process. At the same time, *Tessif* can be used as any other ESSMOS in terms of modelling specfic ESSMs with the added benefit of simultaneously generating a set of results through different ESSMOS highlighting unexpected problems and increasing the generated results' plausibility.

3.6. Tessif's Applicability Beyond the Scope of this Thesis

Apart from the goals formulated in 1.2, oriented more at the scientific aspect of energy supply system modelling software comparison, *Tessif* also assists in dealing with the following more practical problems:

1. Given an expansion or commitment problem, what is the expected range of possible solutions using different ESSMOS designed for the same purpose? Assuming that different origin leads to different presuppositions and focus of such software, implying potential differences in what is modelled and, thus, possible result variance on equal input data.

   Tessif helps answer that by automatically transforming its ESSM (representing the harmonised data input) into that of its supported ESSMOS. An optimisation attempt can then be initialised on these ESSMOS-specific ESSMs. In case of successful optimisation, *Tessif* then collects and aggregates the results to facilitate comparison and thus highlight the (potential) range or deviation of generated results.

2. Given the number of possible software candidates, what would be the best software to perform a specific in-depth analysis when modelling an energy supply system?

   Tessif helps answer that question by providing three significant sources of information:

   - A user guide [61] on how *Tessif* handles model transformation. In its essence, an overview of what solver constraints are formulated by each software's system-model components and how they relate to *Tessif's* approach.
   - A set of application examples [40] for common model scenario combinations, highlighting similarities and differences in results and modelling approaches. They cover the entire spectrum from minimum working examples over comparison-focused examples (including those in Subsection 4.3.1) to full-fletched case studies of specific research endeavours.
   - A hierarchical, abstract and object-oriented-code-base [62] providing a *Tessif*-to-supported-ESSMOS transformation framework. Being, despite its complexity, as sim-

ple and extendable as possible. Allowing complex ESSMOS-specific transformation and general applicability to other component-based ESSMOS.

3.7. Further Development and Longevity as Free and Open-Source Software

Tessif is designed to be used, developed, improved and expanded by the (scientific) community. However, since the current lead developer and author of this thesis will no longer be able to support *Tessif* full-time, a change in lead developer is desired. Otherwise, *Tessif* will likely become impractical and borderline useless in a few years since future updates of the supported ESSMOS will not be reflected. The existing ESSMOS version support will not be affected and will thus always be functional.

Tessif uses public interfaces to communicate with its supported software tools. Since they are expected to be kept backwards compatible, no major problems are anticipated in the case of supported software tool updates. Backwards incompatible changes will, however, require updated transformation utilities.

Tessif's core transformation algorithms inside the `tessif.transform` subpackage are designed to be abstract and modular. Concisely defining the additional inputs required to expand the collection of supported tools. This facilitates the process of expanding *Tessif*. Master course students with no prior programming experience were able to incorporate an additional software tool within the timeframe of completing their master's thesis. An experienced Python developer without prior knowledge of *Tessif* is expected to need about two months of full-time development to implement an additional tool.

Tessif provides a comprehensive example and test library, simplifying the process of altering or expanding the code base. A continuous integration environment is set up streamlining the quality management process and ongoing design. Thus, a solid foundation exists for *Tessif* becoming a community driven project in the future.

3.8. Model Scenario Combinations (MSCs) Created Through Tessif for Showcasing the Comparative Method

To reach Goal 2b, a set of MSCs is created using *Tessif*'s creation template provided by the `tessif.model.energy_system` module.

The term "MSC" is discussed in detail in Subsection 2.1.4. It is, in essence, a combination of the words "ESSM" and "scenario". ESSM thereby refers to the assembly of components and their interconnections. In constrast, a scenario is defined as an entity that sets the ESSM in context with an investigated time frame, secondary objectives like emission targets, and a general question or goal motivating the investigation. The union of an ESSM and a scenario is called model scenario combination (MSC).

3.8.1. Component-Focused MSCs

Two component-focused MSCs built upon a singular ESSM are used in this thesis. Component-focused in this context refers to several implications and predispositions listed below:

1. The energy system is modelled using the graph theory approach, discussed in Section 2.2.
2. Each component follows a certain set of energy-transforming constraints representative of its underlying real-world application, as described in Subsection 3.2.
3. A singular node or component may represent an agglomeration of real-world applications. For example, several coal-fired power plants may be aggregated into a single transformer component.
4. The supply system considers power and heat demands. The mobility sector is neglected.
5. Sector coupling between heat and power is modelled using CHP plants and power-to-heat components.
6. Temporal decoupling of energy production and demand is realised by modelling heat and power storage systems.
7. Demands, renewable energy production, installed capacities, efficiencies and costs are estimated using values found in the literature published until December 2021.
8. The components and their interconnections are chosen such that the resulting ESSM represents a fictional isolated energy supply system loosely related to the macro perspective of a system like that of Germany.
9. There are no connection losses between any two components.
10. There is no disambiguation between possibly occurring differences in potential levels of a given energy carrier (like pressure or voltage).

Points 1 to 3 represent the modelling approach of the software developed within the scope of the research done for this thesis (Chapter 3.). Points 4 to 8 are practical constraints concerning the ESSM generation process. Aspects 9 and 10 are the most important for differentiating the component-focused ESSM from the grid-focused.

3.8.1.1. Modelling Goals

In addition to the points stated above, the developed component-focused MSCs aim to model the following five behaviours:

1. Integrate volatile renewable energy sources into a system with existing conventional power plants.
2. Incorporate energy storage technologies into a system with existing conventional power plants and volatile renewable energy sources.
3. Cost-optimally dispatch the controllable components based on exogenously defined costs and emissions parameters, while adhering to an overall emission limit.
4. Cost-optimally expand and dispatch available components based on exogenously defined costs and emissions parameters, while maintaining an overall emission limit.

5. Model year-round, hourly-resolved energy demands based on ambient climatic conditions.

3.8.1.2. Created Component-Focused MSCs for Use in the Case Study

Two component-focused MSCs were created for this thesis using the above predispositions.

1. The first MSC is a pure commitment problem without expandable components or emission constraints. This model scenario combination is labelled Component-Commitment or CompC in the following.
2. The second MSC is an expansion problem with an emission limit of $0.25 Mt_{CO2,eq}$. This emission constraint allows small amounts of fossil fuel generation while promoting renewable energy and storage utilisation. This combination is labelled Component-Expansion (CompE).

An in-depth discussion about predispositions and a detailed analysis of the design decisions behind the component-focused MSCs can be found in the work of Reimer [63]. In his work, Reimer also conducted a very comprehensive result investigation, analysing and contextualising a large number of different results. Reimer's work, thus, provides a valuable extension to the comparative-method-focused results of this thesis.

The underlying parameter assumptions of the component-focused MSCs are reiterated in Table A.1 of Appendix Section A.1. A generic system visualisation (GSV) of the developed component-based ESSM can be seen in Figure 3.4. Each component is represented as a coloured circle of uniform size, labelled according to the component's name. The interconnections between these components are represented by black arrows, which indicate the permitted directions of energy flow. The two bus components that represent the `Heat` and `Powerline` are drawn with a larger radius than the rest. This design choice facilitates easier identification of the different energy carriers, as well as their respective sinks and sources.

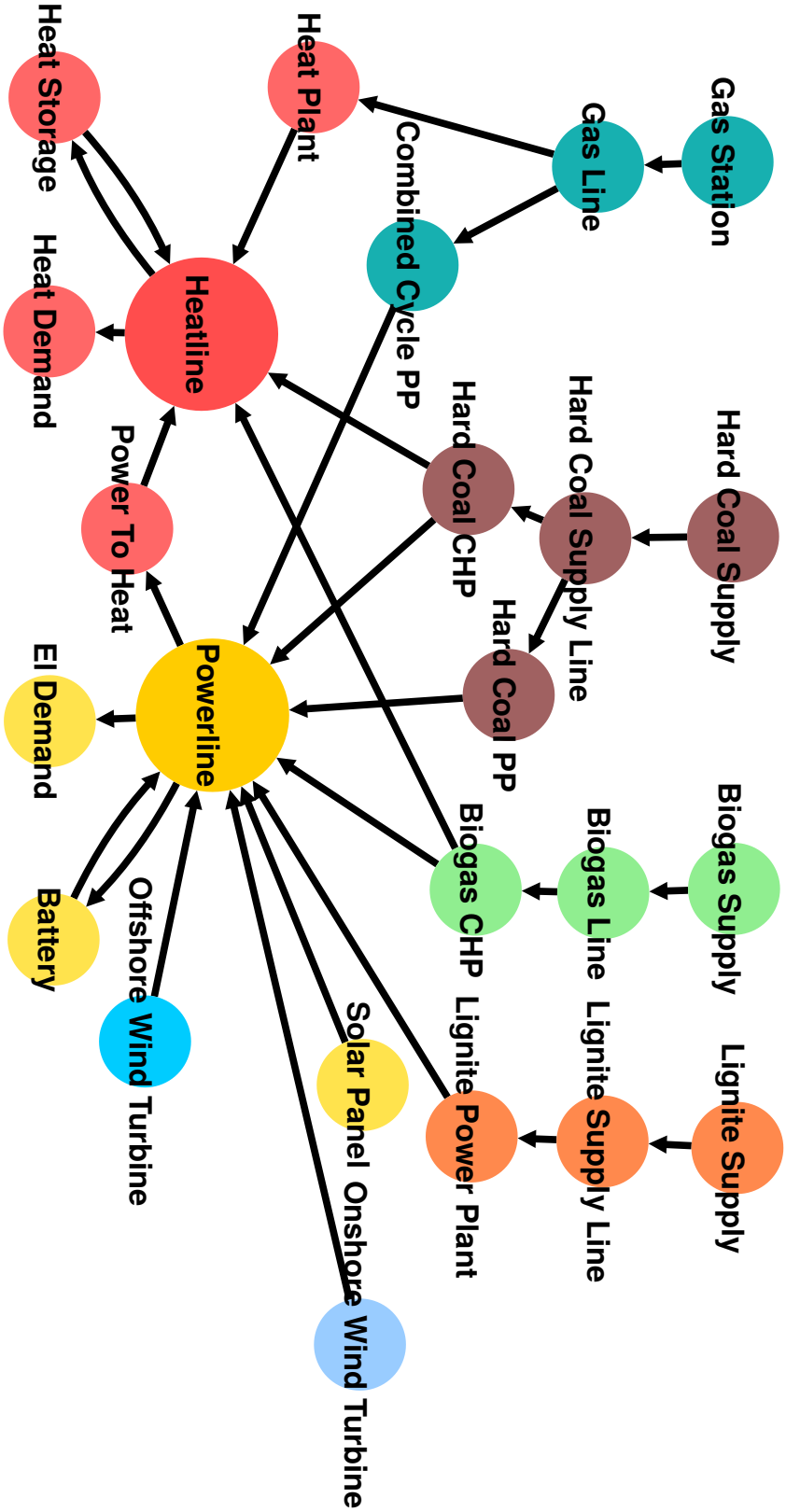

Figure 3.4.: Generic System Visualisation (GSV) of the Component-Focused ESSM. Figure Created Using the `tessif.visualize.dcgrph` Module. ESSM Specifically Designed by Reimer [63] for Comparing ESSMOS. ESSM Not Designed for Investigating Actual Energy Systems in the Context of Applied Research.

3.8.2. Grid-Focused MSCs

Two grid-focused ESSM, with two distinct scenarios, are used in the case study for the ESSMOS evaluation and comparison, resulting in four grid-focused MSCs. The term "grid-focused" encompasses various implications and predispositions. While the first eight predispositions are identical to those for component-focused MSCs, the following two distinguish between component and grid-focused MSCs:

1. Energy transportation losses between components are either modelled actively (using variable flow rates and a loss factor) or passively (exclusively for electricity transportation, employing voltage angles and impedance values). Given that large-scale electrical energy transportation data is readily available, the grid-focused ESSM primarily focuses on the transportation of electrical energy. In theory, transportation of other carriers through different grids, such as chemical energy via gas grids or thermal energy via district heating grids, could be modelled similarly. Active component modelling would be identical, as they do not differentiate between transportation media. However, for passive modelling, additional algorithms and components are required.
2. Potential levels are considered only for modelling electricity transport losses using passive components. No disambiguation is performed for other components. However, since *Tessif*'s passive transportation component is not yet fully implemented, this implication is relevant only for future developed grid-focused MSCs.

3.8.2.1. Modelling Goals

The developed grid-focused MSCs address the following modelling goals:

1. Model transportation losses.
2. Model grid congestion issues and the subsequent need for redispatch.
3. Model transportation capacity expansions to avoid grid congestion and redispatch.

3.8.2.2. Created Grid-Focused MSCs for Use in the Case Study

Four grid-focused MSCs are created for this thesis using the above predispositions:

1. The first Model Scenario Combination (MSC) is a lossless ESSM combined with a pure commitment problem without expandable components or emission constraints. This combination is labelled Lossless-Commitment (LossLC) and serves as a basic foundation for comparing the remaining combinations. A corresponding generic system visualisation can be found in Figure 3.5. The system is conceptually divided into four component clusters: `Low Voltage Grid`, `Medium Voltage Grid`, `High Voltage Grid`, and `District Heating Grid`. These subgrids are connected using `Connector` components (see Subsection 3.2.6), which do not constrain transport efficiency or the amount of transportation.
2. The second MSC emulates grid losses in a pure commitment problem. This combination is labelled No-Congestion Transformer-Commitment (No-Congestion TransC). It models grid behaviour by replacing each subgrid-connecting bus of the LossLC MSC with two one-directional transformer components to constrain transport efficiency and the maximum transfer amount. However, within this No-Congestion TransC, the grid capacity is chosen to be high enough to transmit all occurring transfer loads, hence the "no-congestion" prefix.

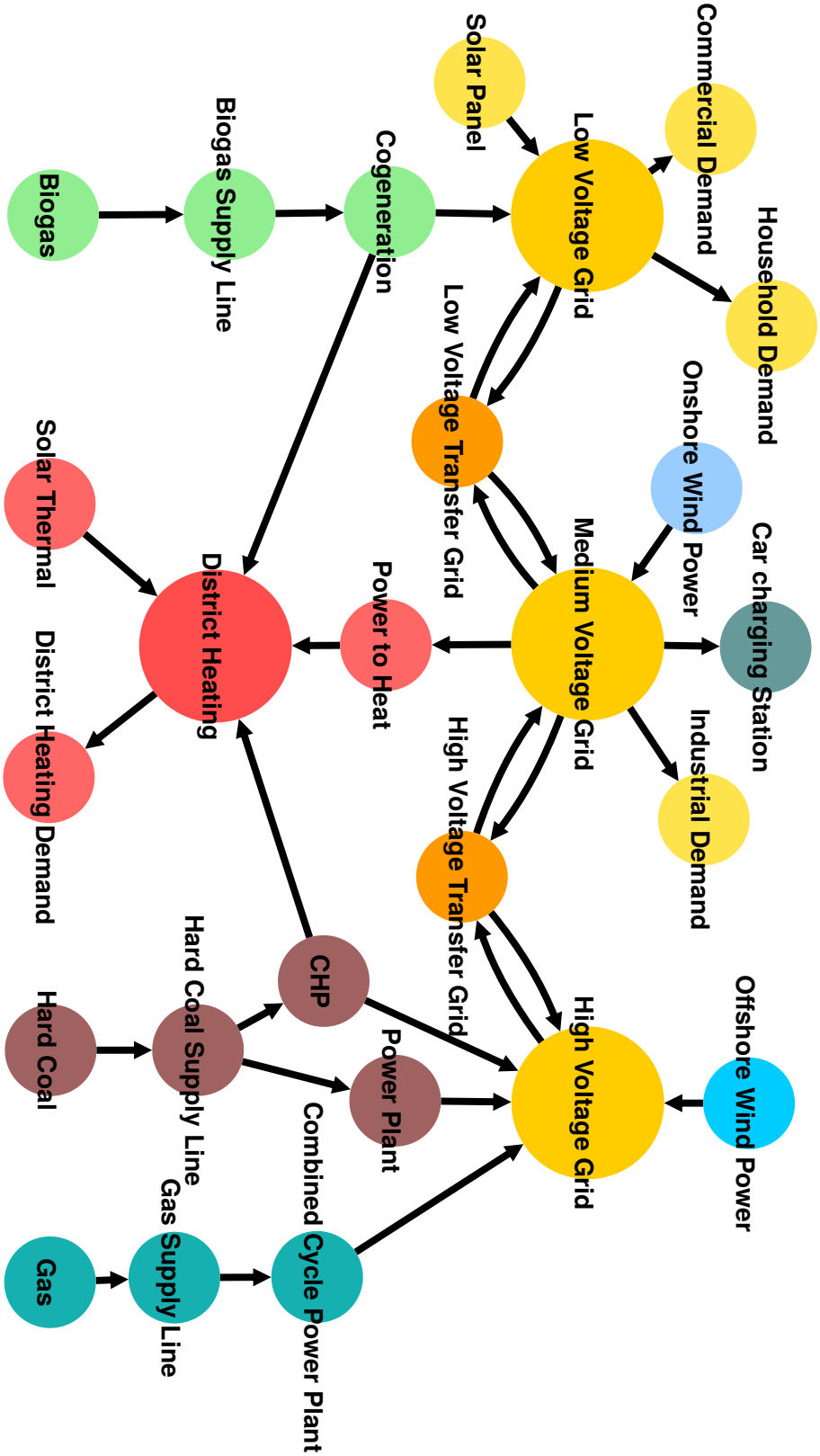

Figure 3.5.: Generic System Visualisation (GSV) of the Grid-Focused LossLC ESSM. Figure Created Using the `tessif.visualize.dcgrph` Module. ESSM Specifically Designed by Hanke [64] for Comparing ESSMOS Tools. ESSM Not Designed for Investigating Actual Energy Systems in the Context of Applied Research.

This MSC operates like the LossLC, with the added transport losses between subgrids. A generic graph representation of the ESSM can be found in Figure 3.6.

3. The third MSC uses the same ESSM as the second MSC but with a significantly lower transfer grid capacity. Since no expansion is allowed, grid congestion occurs, and the subsequent need for redispatch is modelled by forcing the solver to use the expensive balancing pair of sink and source at each subgrid level. This MSC is named Congestion Transformer-Commitment (Congestion TransC).

4. The fourth MSC uses the same ESSM as the second and third MSCs but with expandable subgrid interconnections. This MSC is labelled Transformer-Expansion (TransE). It primarily focuses on grid expansion by allowing the solver to increase transportation capacities if necessary. The balancing pairs of sink and source are parameterised so that some grid expansion is less expensive than none. Simultaneously, cost parameters are chosen such that a compromise between balancing redispatch and grid expansion is the least expensive. This design choice emulates typical grid expansion investigations.

An in-depth discussion about predispositions and a detailed analysis of the design decisions behind the four grid-focused MSCs can be found in the work of Hanke [64]. In his work, Hanke also conducted a detailed result investigation, analysing and contextualising the numerical results. Hanke's work, thus, provides a valuable extension to the comparative-method-focused results of this thesis.

The underlying parameter assumptions can be found in Appendix Section A.2, Table A.2. Additional parameters introduced in this thesis for the TransC and TransE MSCs are listed in Table 3.1. They were added to further provoke grid congestion and subsequent redispatch phenomena, and to facilitate their detection. The relatively low OPEX and CAPEX values of 10€/MWh and 10€/MW seen in the OPEX and CAPEX columns of Table 3.1 were chosen to incentivise energy transport over localised generation. The relatively high OPEX values of 300€/MWh (refer to the OPEX column in Table 3.1) for the balancing source and sink component pairs were chosen so that these components would only be used if no other solution is available. Hence, they are only utilised in cases of grid congestion. For the LossLC MSC no additional parameter assumptions are introduced.

Table 3.1.: Additional Parameter Assumptions for the Transformer Commitment and Transformer Expansion (TransCnE) MSCs.

MSC	Component	OPEX €/MWh	CAPEX €/MW	Capacity MW	Efficiency
No-Congestion	Transfer Components	10	-	60000	0.93
Congestion	Transfer Components	10	-	20000	0.93
	Balancing Source/Sink Pair	300	-	infinite	-
TransE	Transfer Components	10	10	1	0.93
	Balancing Source/Sink Pair	300	-	infinite	-

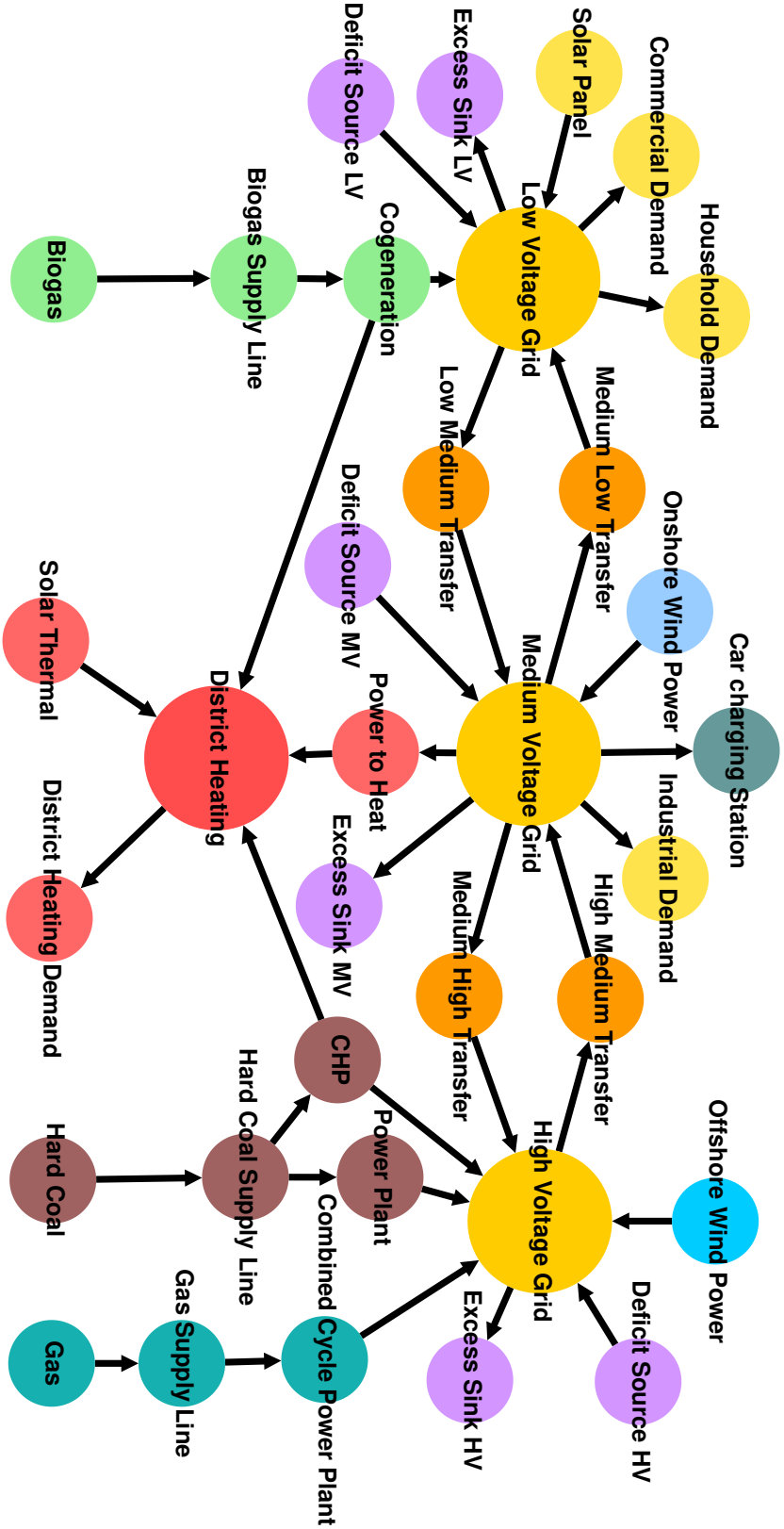

Figure 3.6.: Generic System Visualisation (GSV) of the Grid-Focused TransChnE ESSM. Figure Created Using the `tessif.visualize.dcgrph` Module. ESSM Specifically Designed by Hanke [64] for Comparing ESSMOS Tools. ESSM Not Designed for Investigating Actual Energy Systems in the Context of Applied Research.

4. Developed Method for Choosing and Comparing Energy Supply System Modelling and Optmisation Software

Chapter 4 constitutes the second of the two core contributions of this thesis. It lays out the proposed strategy to choose a particular ESSMOS tool out of a potentially large pool of candidates based on a sophisticated preselection and a detailed result-based comparison. The chapter formulates a step-by-step procedure and discusses the respective advantages and disadvantages while highlighting optional steps, offering alternative approaches and stating additional application contexts.

4.1. Conceptual Overview of the Developed Comparative Method

The developed Comparative Method (CM) consists of five steps labelled comparative-method-steps or CM-steps for short. These CM-steps are divided into subsets of actions and are generally designed to be executed in the order presented with optional actions labelled as such. The five comparative-method-steps of this systematic procedure are as follows:

CM-Step 1: Preselection (Section 4.2)

CM-Step 2: Modelling and Optimising (Section 4.3)

CM-Step 3: Preliminary Result Analysis (Section 4.4)

CM-Step 4: Comparative Result Analysis (Section 4.5)

CM-Step 5: Scalability Assessment (Outlook chapter; Section 7.1)

Figure 4.1 provides a visual overview of the five aforementioned comparative-method-steps, including the respective actions to perform. Each of these accordingly labelled sub-steps represents a set of operations explained in detail throughout this chapter, highlighted by the corresponding keywords of the section headings in bold letters.

Given this work's emphasis on result-based comparisons, comparative-method-steps one and two can be considered somewhat loosely associated or auxiliary. Theoretically, those could be replaced with arbitrary output data from different software tools. These CM-steps, however, do tightly link to the formulated goals of this thesis (see Section 1.2). Modelling practice has shown that utilising CM-step one and two can significantly increase the informative value of the comparison. They were specifically designed to aid energy supply system modelling and ESSMOS selection.

Within the scope of this thesis, comparative-method-steps one to four are considered as a template for the Comparative Method and are executed during the case study including all sub-steps.

Beyond this work's application, CM-steps three and four can also be utilised outside the Comparative Method as a guided postprocessing strategy. CM-step five is somewhat of an additional investigation technique and can, in cases scalability is the primary concern, be the only relevant step of the proposed Comparative Method.

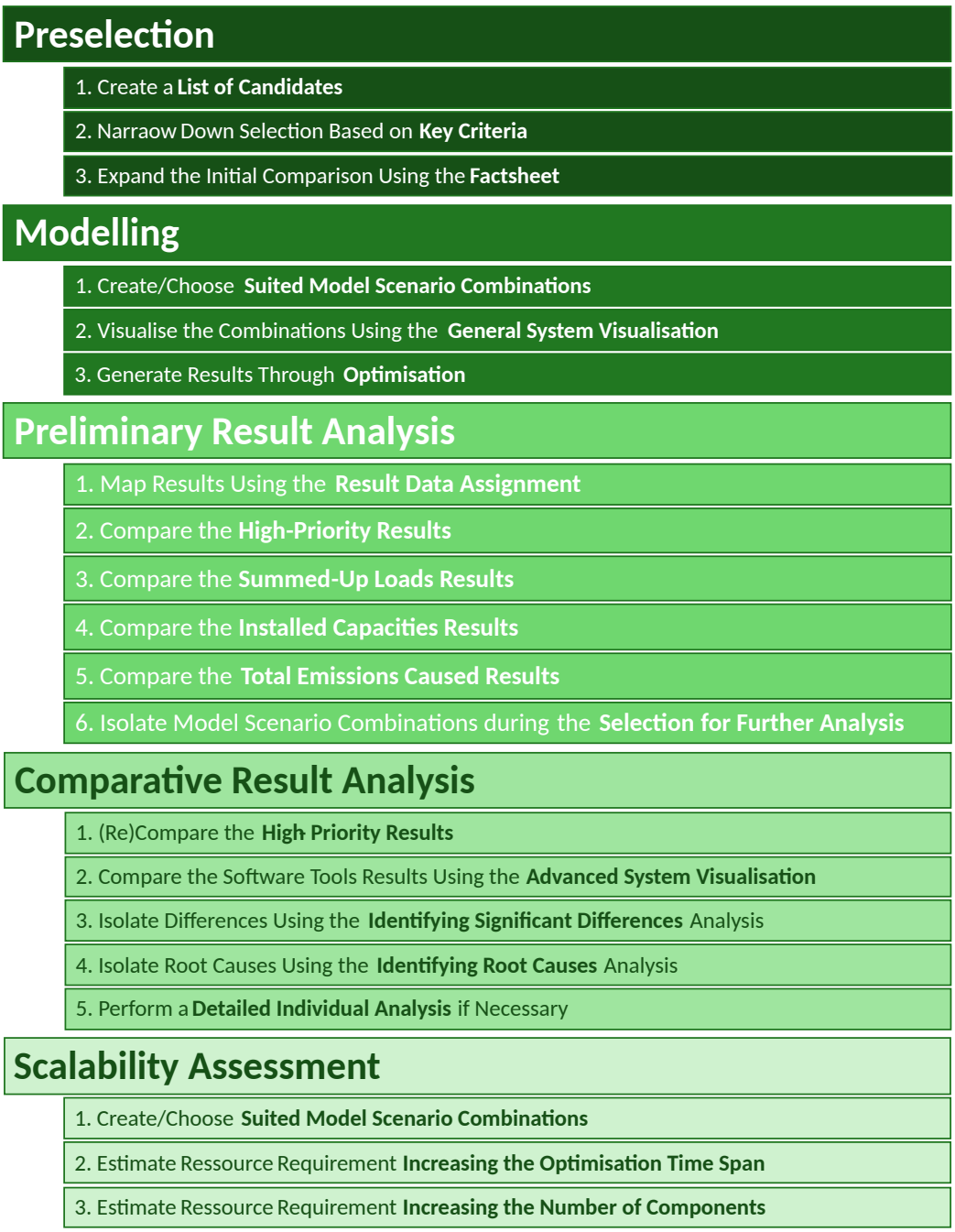

Figure 4.1.: Overview of the Developed Comparative Method.

4.2. CM-Step 1: Preselection of Suited Free and Open-Source (FOS) Energy Supply System Modelling and Optimisation Software (ESSMOS)

Goal 3, formulated in Section 1.2, requests using modern FOS ESSMOS to test and demonstrate the developed Comparative Method. The presented preselection procedure, therefore, concentrates on isolating an appropriate set of these tools. Since Goal 3 further specifies these tools to be modern and highly capable, the focus within this thesis is thus directed to FOS ESSMOS that are (like *Tessif*) energy supply system modelling frameworks since they are considered the most advanced and universally applicable [65].

However, it is essential to note that the general procedure of the entire method and the proposed preselection are not limited to only FOSS tools.

4.2.1. Preselection-Step 1: Creating a List of Potential Candidates

The primary goal of the first preselection-step is to acquire a pool of possible candidates to be used during the comparison. It follows the concept of quantity over quality and is designed to be the first selective endeavour, where only the obviously unfit software tools are sorted out. Recommended places to gather information to create a list of possible candidates are the openmod initiative [59] and the comparisons of Ringkjøb et al. [4] and Conolly et al. [33].

4.2.2. Preselection-Step 2: Key-Criteria-Based Reduction in the Number of Candidates

The second preselection-step aims at drastically reducing the number of potentially investigated software tools. This reduction is achieved by identifying modelling demands of primary importance, which are thus indispensable to the task and subsequently disqualifying software tools not meeting these particular demands.

The listed key-criteria emerged throughout the research on this topic in cooperation with the works of Körber [66]. They are considered somewhat arbitrary and very subjective. Nevertheless, many well-received recommendations were given by the author based on this key-criteria list. **The key-criteria presented are thus seen as one significant result of this thesis.** They are, therefore, ordered in a tabular template, depicted in Table 4.1, for future reference and reuse. Each key-criterion (see the left column in Table 4.1) potentially encompasses several key-criterion-categories (see the right column in Table 4.1).

4.2.2.1. Key-Criterion 1: The Software Tool Author's Intent

Understanding the software author's intent was found to be the most straightforward way to rule a software tool in or out. Any verbal or written intent is strongly suggested to be taken into heavy account if accessible. Information about the author's intent often encompasses the core tasks the tool was designed for, its strengths and weaknesses, and its disambiguation from other similar software (see also the author's intent criterion in Table 4.1). If available, using key-criterion one is, thus, recommended to be the first selective criterion to be applied to the list of candidates.

Table 4.1.: Key Criteria to Reduce the Number of Candidates During Preselection-Step 2.

#	Key-Criterion	Categories
1	The Authors Intent	-
2	Resolution	Reported Suitability Temporal Spatial
3	Scalability	Reported Suitability Subjective Impression
4	Reduction of Computational Effort	Temporal Aggregation Spatial Aggregation Intermittency Rolling Horizon
5	CHP Components	Pre-built-in CHP Component Template Inflow Constraint Only Pre-Built-In Field of Operation
6	Energy Storage Components	Initial SOC Type of Emission Allocation Mandatory Cyclic SOC Individual Cyclic SOC
7	Energy Transport	Active Passive

4.2.2.2. Key-Criterion 2: Available Resolution

Often, temporal and spatial resolutions are the main constraining factors regarding the modelling task. Hence, one of the most important questions to be answered when modelling an energy supply system is that of resolution. It differs immensely between tasks and is often a good indicator to rule out software tools unable to deal with the required resolution. However, in most energy supply system modelling use cases, an hourly temporal resolution represents the quasi-standard. It allows for a semi-dynamic analysis, where constraints like weather changes or power plants' cold starts can be modelled. At the same time, highly dynamic phenomena like control energy can be neglected.

The software tool's documentation typically includes information about the applicable resolution and is seen as a useful first indicator (see also the reported suitability category of the resolution criterion in Table 4.1).

Spatial resolution, on the other hand, is usually formulated quite vague since it often does not affect system modelling directly but rather weather data input and external constraints like land requirements (see also the spatial category of the resolution criterion in Table 4.1).

Key-criterion two, especially its temporal resolution aspect, should be used as the second selective criterion to reduce the number of possible candidates (see also the temporal category of the resolution criterion in Table 4.1).

4.2.2.3. Key-Criterion 3: Techniques for Improving Scalability

The question of scalability could be formulated as follows: How well does the optimisation model scale in terms of the number of components and the investigated time frame?

The question ties tightly into that of resolution and the reduction of computational effort. Large-scale investigations of energy supply systems like, for example, the European transfer grid often limit the number of viable tools significantly. This topic is, of course, not independent of the available hardware. Therefore the question of scalability is also subject to the individual use case and available hardware infrastructure.
It can often be challenging to estimate a tool's scalability beforehand. Possible sources of information can be the author's intent and conducted and published studies. These do not rule out differing scopes, but hint at a magnitude of scale others deemed sensible for that particular tool. The software tool's documentation about scalability-related topics also usually provides useful information (see also the reported suitability category of the scalability criterion in Table 4.1)

When assessing the scalability of energy supply system modelling software tools, it is recommended to remember that scaling up is often more challenging than scaling down. To build on the example of the European transfer grid: Using a tool designed for that particular purpose may very well be used to model a small-scale isolated supply system. However, the possibility exists that a far more detailed analysis might be feasible, desirable and, in fact, possible by using a more appropriately scaled tool. On the other hand, using a tool designed to model the behaviour of a district's energy supply system is most likely incapable of modelling the whole European continent. Be it due to the interface or the infeasible number of components and time steps necessary for performing the optimisation (see also the subjective impression category of the scalability criterion in Table 4.1).

Supposing information concerning key-criterion three is available, scalability is the primary concern. The list of candidates can be reduced significantly, applying it as a selective criterion.

4.2.2.4. Key-Criterion 4: Techniques for the Reduction of Computational Effort

Another primary indicator of facilitating the selection process is the potential to reduce overall computational effort by the most prevalent means of aggregation, intermittency and rolling horizon.

Again, this is tightly linked with the questions of resolution and scalability. The reduction in computational effort (ROCE) might allow a particular modelling tool to be used outside of its detailed scope on a potentially much larger scale using some or all of the aforementioned techniques. This ROCE may alleviate the scalability constraints discussed before.

Generally, two types of aggregation techniques can be applied within the domain of energy supply system modelling. The first is, representing several individual components of similar or identical type as one unified component. Which usually reduces the amount of data in two ways (see also the spatial aggregation category of the reduction of computational effort criterion in Table 4.1):

1. Fewer components often imply fewer constraints and, thus, less computational effort.
2. By representing many physical components within one model component, multiple constraints can be lifted. Prominent examples are up and down times (non-linear) and load

gradients (linear). The underlying argumentation is that within the large number of physical actors represented by such a component, a subset can always be found that fulfils the solver's current solution. A typical example is the aggregation of all fossil-fuelled power plants of the same fuel within one region into a singular model component.

The second type of aggregation is that of time steps. By reducing the temporal resolution and averaging the subsequent demand and production constraints, the overall real-time needed for computation can be decreased. Since this also reduces the temporal accuracy, a balance has to be maintained using this technique (see also the temporal aggregation category of the reduction of computational effort criterion in Table 4.1).

Another form of reducing the computational effort is to reduce the number of time steps by employing intermittency approaches (see also the intermittency category of the reduction of computational effort criterion in Table 4.1). This reduction is achieved by introducing periodic production and demand constraints to approximate the whole time frame as a set of periodic frames so that the solution to one period is a solution to all periods of the same type. An oversimplified example would be dividing an entire year into four seasons, each represented by a singular weak. The problem of optimising the ESSM within a time span of 8760 hours (one year) gets reduced to optimising 672 hours (4 weeks) [67, 68, 69].

A similar strategy to reduce the overall data per optimisation process is that of the rolling horizon (see also the rolling horizon category of the reduction of computational effort criterion in Table 4.1). Instead of optimising a time span of 8760 hours at once, the span is divided into several smaller horizons (typically around 1 to 3 days). Each horizon is then optimised on its own, and the results of the $n-th$ horizon represent the starting values of horizon $n+1$ (see also Subsection 2.1.6).

Supposing scalability is of primary interest, key criterion four alongside key-criterion three should serve as the main selective criteria.

4.2.2.5. Key-Criterion 5: Modelling Approach of CHP Components

When modelling combined heat and power plants (CHP) following modelling approach differences were identified among the investigated software tools:

1. No pre-built-in support is provided for relating one inflow and two outflows of separate energy carriers to each other using conversion factors (see also the pre-built-in CHP component template category of the CHP components criterion in Table 4.1).

2. The software tool exclusively supports constraining the energy inflow in conjunction with different conversion factors for heat and power outflows (see also the inflow constraint only category of the CHP components criterion in Table 4.1).

3. The software tool allows describing a feasible field of operation using a heat-and-power diagram with subsequent power loss coefficients and minimum and maximum load, as well as a backpressure line (see also the pre-built-in field of operation category of the CHP components criterion in Table 4.1).

These different modelling approaches were found to impact the results severely. If the modelling task is centred around CHP technologies and thus dependent on specific modelling approaches, key-criterion five should be considered heavily during software selection.

4.2.2.6. Key-Criterion 6: Modelling Approach of Energy Storage Components

Another significant modelling-approach-related difference between investigated software tools was found within the parameterisation of storage components, expressed in the following questions:

1. Can an initial state-of-charge (SOC) greater than zero be provided? (See also the initial SOC category of the energy storage components criterion in Table 4.1.)
2. Are the storage-related emissions allocated to in- or outflow as opposed to initial and final SOC? (See also the type of emission allocation category of the energy storage components criterion in Table 4.1.)
3. Is there a mandatory cyclic SOC constraint enforced by the software tool? (See also the mandatory cyclic SOC constraints category of the energy storage components criterion in Table 4.1.)
4. Can some storage components within the same ESSM have a cyclic SOC constraint while others do not? (See also the individual cyclic SOC constraints category of the energy storage components criterion in Table 4.1.)

The investigations conducted in Reimer [63] show the influence of those differences on the optimisation results. In his study, he showed that the storage behaviour significantly impacted the overall costs, especially when investigating an emission-constraint expansion problem.

The differences, however, are evaluated as not so severe as to rule out any model based on them since the behaviour is hard to predict without actual testing. Instead, it is recommended not to overvalue the accuracy of emission-constrained expansion problem results and to consider using several models to circumvent the issue by creating a range of feasible solutions. Key-criterion six should, therefore, only be used to rule out the obviously unfit candidates.

4.2.2.7. Key-Criterion 7: Modelling Approach of Energy Transport Utilities

The final key criterion found to be highly decisive is how to approach energy transportation modelling. This approach encompasses modelling losses during transportation and potentially sophisticated physical carrier-dependent values like potential level (pressure, temperature, voltage (angle)) and other "secondary" values like velocity or current. Within the realm of optimising ESSMs, the issue of energy transportation can be roughly grouped into two categories:

1. **Active** components:

   Energy transportation losses are modelled using an efficiency value. The solver can **actively** decide whether to use a transportation component or not within the range of flow rate constraints (see also the active category under the energy transport criterion Table 4.1). The amount of transported energy is calculated independently of any physical values (except the efficiency or loss factor).

1. **Passive** components:

   Losses are modelled using physical values like friction or electrical resistance (reactance). The amount of energy transported is subject to differences in potential levels like pressure or voltage (angle). Which are, in turn, subject to the beforenamed physical, resistance-like parameters. This dependency implies that if an adjacent bus has energy flowing through it, a subsequent flow through the transportation component establishes. Hence the solver

has **no active** control over the transportation component, only indirect (if any) control via the adjacent bus's neighbouring components (see also the passive category under the energy transport criterion in Table 4.1).

The decisive weight of this key criterion has to be taken into serious consideration because, in grid-like structures, passive energy transportation is rarely modelled for all relevant energy carriers (power, hot water/steam, gas).

Key-criterion seven should serve as the primary selective criterion in cases where a passive model approach is desired, and further numerical analysis on the physical transportation aspect is mandatory. However, using several unique software tools for these in-depth analyses might still be necessary. Therefore, the lack of passive modelling capabilities can often be neglected in broad-scale or initial-approach investigations, especially when commitment or emission constraint expansion problems are of primary interest.

4.2.3. Preselection-Step 3: Factsheet-Based ESSMOS Tool Selection

The target of preselection-step three is gathering relevant facts about the leftover candidates without actually utilising them to accumulate enough valuable information to make an educated decision about which tools to investigate and compare further. The actual work consists of collecting data to fill out a specifically designed tabular template that streamlines the decision which tools to use.

The main difference between the key-criteria and the factsheet-based comparison is that the later covers more aspects in greater detail and is significantly more technical. It also involves utilising the corresponding software tool documentation as a user would and generally requires more expertise than the execution of preselection-step two. It is, therefore, only utilised when deemed necessary or beneficial.

An in-depth explanation of how and why this fact sheet was developed based on contemporary scientific literature alongside the resulting tabular template can be found in the work of Körber [66]. The factsheet created as part of the case study in Chapter 5 is given in Appendix Tables A.4 to A.6 and consists entirely of the concept developed by Körber.

4.3. CM-Step 2: Modelling and Optimising Suited Model Scenario Combinations (MOSMSC)

CM-step 2 comprises the actions necessary for modelling and optimising suited model scenario combinations (MOSMSC). The aim is to obtain one set of results for each tool regarding the same initial MSC, for all prior selected combinations. This procedure enables a result-based comparison while reducing potential deviations due to input data harmonisation and providing similar or equal presuppositions.

This part of the Comparative Method can be omitted in case implied by decision factors outside this thesis' scope. However, following the MOSMSC-steps recommended below is deemed sensible for the case study conducted in Chapter 5 and most of the anticipated use cases.

4.3.1. MOSMSC-Step 1: Selecting Suited MSCs

MOSMSC-step 1 directly addresses Goal 2b. Its primary target is to choose or create one or several ESSM to be optimised as part of one or multiple scenarios.

Depending on the comparison's initial goal, the underlying MSCs can significantly impact the results' reliability and (re)useability. The presented method utilises two distinct ESSM representations as part of six scenarios to create repeatable and transparent results. Since the MSCs were created using the developed software framework, they are presented and discussed in Subsections 3.8.1 and 3.8.2.

These six MSCs were specifically created for their sole purpose of comparing modern ESSMOS tools and to investigate the limits of their modelling approaches. And although their design is inspired by actual modern modelling task, they are not intended to model an actual existing energy system or serve any application or project-oriented research.

4.3.2. MOSMSC-Step 2: Initial Generic System Visualisation (GSV)

After choosing or creating suited MSCs, it is recommended to visualise the corresponding ESSMs. This initial visualisation is best done using the GSV described in Paragraph 2.2.2.1. This visualisation serves two primary purposes. The first is a simple plausibility check to see if all components are connected the way intended. The second is to get a visual impression of the complexity of the chosen ESSMs to potentially limit the focus of the detailed analysis on certain parts or aspects of the ESSM. Hence, visualising the system models is often tightly interwoven with the process of creating model scenario combinations.

4.3.3. MOSMSC-Step 3: Optimising the Selected MSCs

As a final endeavour of CM-step 2, optimisation results are to be generated based on the selected MSCs. The ESSMOS tools used typically provide an interface to initiate the optimisation process. Apart from interacting with this interface, no further user-initiated action is to be taken. The process of transforming the model scenario combinations into a set of solver software readable commands is typically hidden from the user via several layers of abstraction, as discussed in Subsection 2.1.8.

4.4. CM-Step 3: Preliminary Result Analysis (PRA) of the Optimisation Results

The recommended PRA is an attempt to reduce the amount of work to be done by manually comparing the readily available result visualisations in order to get an initial estimate on the following two aspects:

1. How similar are the obtained results?

   The comparison result-extremes very similar and very different, are easily detected using the PRA. Very similar results make a comparative result analysis unnecessary. For very different results key differences can often be identified by just comparing the installed capacities or summed-up loads, allowing this step to be omitted.

2. What result range is to be expected?

   If the actual results are of greater interest than the comparison between ESSMOS, the PRA is well-suited for providing the most important results and their subsequent differences. Thus, the PRA yields information on the amount of result deviation to be expected when utilising different software options.

   Suppose the PRA is the only analysis to be performed. In that case, it is advisable to additionally calculate relative deviations using the results of one software as a reference, highlighting more minor differences that are otherwise difficult to detect.

The main objective of the PRA as CM-step 3 is, thus, a reduction in the amount of data to be analysed in detail. This is achieved by comparing the most easily accessible results and identifying those model scenario combinations worth investigating further.

4.4.1. PRA-Step 1: Performing the Developed Result Data Assignment Technique

The main priority of PRA-step 1 is to make the calculated load $P(s,n,t)$ and installed capacity $C(s,n,t)$ for all ESSMOS s, all nodes n and all time steps t readily available. This result-allocation is achieved by following the proposed graph-oriented post-processing approach discussed in Subsection 2.2.3.

In practice, this means developing a post-processing algorithm that evaluates the optimisation results to sort and assign them to their respective node and edge representations. *Tessif*'s transform subpackage (Paragraph 3.4.1.1) already incorporates such routines. The implemented algorithms cover a large spectrum since the supported ESSMOS tools vary significantly in their internal representation. These subpackages should therefore serve as useful templates in case a personal approach is necessary or desired. Technical details regarding these ESSMOS-specific algorithms are discussed in Schnute [8] and Reimer [70] regarding *FINE* and *Calliope*, respectively. Information on *PyPSA* and *oemof* can be found in *Tessif*'s online documentation [61] concerning the *es2mapping* functionality.

4.4.2. PRA-Step 2: Inspecting the High-Priority Results (HPR)

PRA-step 2 aims at comparing the most relevant or high-priority results (HPR). The main priority of inspecting the HPR is to identify components and result categories worth investigating further. Four central result values have proven to be the first to be referred to. These include the **total costs**, the **total OPEX**, the **total CAPEX** and the **total emission** as listed in the result value column of Table 4.2. Within the developed software framework and the conducted case study, these results are aliased as shown in the alias column of Table 4.2.

Table 4.2.: Identified High-Priority Results (HPR).

Result Value	Alias
total costs	total costs
total OPEX	OPEX
total CAPEX	CAPEX
total emissions	emissions

4.4.3. PRA-Step 3: Inspecting the Summed-Up Loads Results

Goal of PRA-step 3 is to inspect and compare the summed-up loads results $\sum P(t)$, as described in Table 2.6 in Subsection 2.2.3. These results are recommended to be the second entry point of investigation after inspecting the high-priority results. They are much quicker to comprehend than the actual load profile results, especially in cases of extended optimisation time spans. In a pure commitment scenario, the summed-up loads results may often be the only result category necessary to be inspected to assess the degree of similarity confidently.

To quickly compare the summed-up loads results, even for large numbers of components, a horizontal bar plot was found to be the most practical. It has proven to be the best compromise between accessibility, complexity and extensibility and is therefore utilised throughout this work.

4.4.4. PRA-Step 4: Inspecting the Installed Capacity Results

PRA-step 4 aims at inspecting and comparing the installed capacity results P_{Cap}. The installed capacities are thereby only subject to solver modification in the context of expansion scenarios. Within those scenarios, however, they are typically of the most interest since they have the highest cost impact.

As with the summed-up loads results, a horizontal bar plot clustered by component for each ESSMOS was found to be the most practical compromise. In cases a detailed ESSMOS comparison is the primary investigation target, plotting installed capacity results relative to a singular ESSMOS is advisable.

Since expansion scenarios are often created in conjunction with emission constraints, investigating the total emissions caused results, as stated below, is recommended.

4.4.5. PRA-Step 5: Inspecting the Total Emissions Caused Results

PRA-step 5 comprises the inspection and comparison of the total emissions caused results $\sum e(t)$. Emission-constrained expansion problems are highly prioritised in current research for which this result value is often of utmost relevance. During the modelling work done for this thesis, emission allocation was found to be one of the major differences between the investigated ESSMOS. These differences can lead to large result deviations, making the total emissions caused results a critical category to investigate.

To plot the total emissions caused results in order to perform the initial manual comparison, the same procedure as with the summed-up loads and the installed capacity results is advised.

4.4.6. PRA-Step 6: Evaluating the PRA-Results for Further Analysis Selection

The main goal of the PRA as CM-step 3 is to reduce the amount of data investigated during the following detailed comparative result analysis (CRA) and thus reduce the amount of manual work. This reduction mainly includes sorting out those model scenario combinations generating equal enough results. Occasionally, however, this can also mean identifying specific components differing significantly, narrowing down the potential root causes.

The Goal of PRA-step six is, therefore, evaluating PRA-steps two to five, to identify those MSCs that require further analysis. Thereby, the HPR typically give the first significant indication in the form of the most relevant result category, usually corresponding to the result deviating the most.

The exact procedure usually differs slightly between the commitment and expansion scenario archetypes. The former is rarely emission constrained in a way impacting the entire system. At the same time, the actual load results of the latter are usually only subject to different capacity expansions and thus only a side effect. However, the essential goal remains the same between both archetypes, in that those components are tried to be excluded from further investigations that do not significantly impact observed deviations.

4.5. CM-Step 4: Comparative Result Analysis (CRA) of Selected Optimisation Results

Selecting an ESSMOS based only on CM-step one, the preselection, as described in Section 4.2, can be sufficient. Especially in cases of tightly scheduled projects. For large-scale investigation of significant impact, financially, responsibly or otherwise, the proposed CRA, executed as CM-step 4, can be of significant benefit. This usefulness mainly lies in the increased probability of avoiding a software change mid-project and increasing the confidence that the selected pool of candidates is sufficiently well-suited.

The following subsections divide the result comparison into five CRA-steps, of which the first four are guided and partly automated. The fifth and final CRA-step consists of a case-specific detailed analysis. It may vary significantly between investigations and depends on the individual problem formulated. Hence CRA-steps one to four can be seen as an elaborate preparation to reduce the time-consuming and complex final analysis and assessment.

4.5.1. CRA-Step 1: Inspecting the High-Priority Results (HPR)

Inspecting the HPR is the first recommended CRA-step. If no PRA was performed, the same procedure is advised, as discussed in Subsection 4.4.2. CRA-step 1 can be skipped in case the HPR were already analysed in PRA-step 2.

4.5.2. CRA-Step 2: Creating the Advanced System Visualisation (ASV)

Utilising the ASV, as described in detail in Paragraph 2.2.2.2, allows for rapidly comparing the optimisation results of potentially complex ESSMs and multiple ESSMOS. By visually encoding the most relevant optimisation results, differences often get highlighted clearly, simplifying the process of selecting aspects for further analysis. Sometimes, this kind of comparison can be the only one necessary. Often, for example, when implementing new ESSMOS tools or ESSM components due to the simplicity of detecting differences during testing.

4.5.3. CRA-Step 3: Identifying Significant Differences (ISD)

The main objective of CRA-step three is to identify significant differences (ISD) between the compared ESSMOS. This process can be decomposed into two aspects. First, define what a

significant difference is and, second, identify all occurrences of these differences. Paragraphs 4.5.3.1 and 4.5.3.2 address these issues by proposing a distinct set of generally applicable techniques based on the leveraged statistical tools (Section 2.3) implemented into *Tessif* (Paragraph 3.4.1.4).

4.5.3.1. ISD-Step 1: Detecting and Isolating Components of Interest

The first ISD-step is to isolate the components of particular interest. Often but not always, some components have already been identified as potentially interesting by the PRA. In these cases, detecting the components as described below can be seen as a backup and reassurance mechanism. However, to avoid making this a requirement and to allow the identification procedures to be adapted independently of this work, these processes are laid out as though no particular interest was formulated beforehand.

There are two different approaches introduced below. The first is a manual visual inspection based on the ASV developed in Paragraph 2.2.2.2, best suited for small to medium size ESSMs. The second is a statistical-aided concept supporting the first while also suited for large-scale ESSMs.

It is essential to note that a thorough preliminary result analysis can replace both. However, for an experienced user, manually inspecting the ASV results is much faster than manually executing each PRA-step. On the other hand, the statistical-aided identification of significant differences is fully automated, well-scaled and performed within seconds. Therefore, both techniques serve as very valuable assets even though a thorough preliminary result analysis can achieve the same outcome.

Visual Comparison: Before executing any automated or statistical-aided identification routines, researchers usually like to develop an idea or concept of the general nature and scope of the solver-found results, allowing them to validate the plausibility of any other machine-generated results. The ASV technique offers the possibility to do so by densely encoding the found results in a visual representation, therefore, allowing quick and simple visual identification of differing component results. Formulating sophisticated, potential root causes using the ASV observations in combination with the high-priority results is often possible since they usually determine which aspects to be compared.

The limitations of the visual comparison are those of the ASV technique itself: Large or complex systems potentially make additional user interaction necessary, and inexperienced users are often overwhelmed by the amount of information conveyed. Using this technique as it is right now is, therefore, only strictly recommended for small to medium-scale ESSMs. However, experienced users may also find it applicable and helpful beyond the scope of this recommendation. Especially when creating an interactive and infinitely scalable ASVs inside a browser application using the `tessif.visualize.dcgraph` drawing utility.

Statistical Identification: Several statistical error values can be used to calculate average result differences to help identify components of interest. A relevant excerpt of these is discussed in Subsection 2.3.3. Remains the issue of setting a threshold on which a difference is to be interpreted as insignificant instead of significant in the sense of value for further investigation. A low threshold may lead to insurmountable amounts of data for detailed analysis or data not of interest. On the other hand, a threshold set too high might suggest quite similar results,

whereas, in reality, a manual comparison would leave the analyst under the impression that notable differences exist and are worth further investigation.

Two approaches are proposed based on whether the results are a time series (as in component loads) or a static/singular value result (as in installed capacity) to identify component results that differ significantly between the compared ESSMOS.

For **identifying significant differences** in **time series results**, an evaluation of interest is proposed based on using a correlation coefficient in combination with a normalised error value. Section 2.3, provides the corresponding theoretical background.

The standard use case recommends using the **Pearson correlation coefficient** (PCC; Sbusection 2.3.2) combined with the **normalised mean average error** (NMAE; Subsection 2.3.4), calculated for each component, being compared across ESSMOS. Comparing a simple energy system of, e.g., three components between two ESSMOS would thus result in six PCC values and six NMAE values. For an initial run, a PCC threshold of 0.7 and an NMAE threshold of 10% are recommended.

It is essential to note that all four parameters, correlation coefficient and error value used, as well as their respective thresholds, are subject to the user's choice. They are thus designed to be adapted to different use cases or preferences. **This default parameterisation of the standard use case was found to be best suited for detecting slight to medium deviations. However, it proved to be less helpful in cases many large differences exist since they all just get flagged as of high interest. In these cases, the PRA of CM-step 3 (Section 4.4), has proven to be more helpful.** Alternatively, lowering the PCC threshold and increasing the $NMAE$ threshold could more easily differentiate large from very large deviations.

Utilising the combination of two clustering parameters (PCC and NMAE in this case) leads to four different categories of varying interest/ significance as illustrated in Table 4.3.

Since the PCC and $NMAE$ are always only calculated between two datasets, either an additional set of results has to be introduced during calculation, representing the arithmetic mean of all results, or one of the result sets is used as a reference. PCC and $NMAE$ values are thus to be interpreted in reference to the corresponding result set. The *oemof* results are used as a reference throughout this thesis since *Tessif*'s ESSM resembles that of *oemof* the most.

Table 4.3 summarises the proposed component-identifying technique and the estimated interest ratings.

Table 4.3.: Thresholds and Degrees of Interest for Identifying Components of Significant Difference.

	$PCC < 0.7$	$PCC \geq 0.7$
$NMAE < K = 10\%$	medium interest	low interest
$NMAE \geq K = 10\%$	high interest	medium interest

Three different interest ratings are proposed based on four different combinations of PCC and $NMAE$:

1. $PCC < 0.7$ and $NMAE > K = 10\%$; High interest / significantly different:

   This case represents the most relevant or interesting differences to detect between the ESSMOS. A PCC of less than 0.7 implies differing dynamic component behaviour. An

$NMAE$ of greater equal 10% indicates that on top of the differing dynamics, the resulting deviations are worth noting.

2. $PCC < 0.7$ and $NMAE < K = 10\%$; Medium interest / borderline significantly different:

   This case may be relevant when isolating relevant differences between the ESSMOS. However, it is often but not exclusively a side effect. Like above, the PCC value less than 0.7 indicates differing dynamics. In contrast to the case above, these differences only amount to less than 10% on average.

3. $PCC > 0.7$ and $NMAE > K = 10\%$; Medium interest / borderline significantly different:

   Combination three also describes a case only of potential interest. A PCC value greater than 0.7 indicates a somewhat synchronous dispatch pattern. In conjunction with a relatively high $NMAE$ of greater than 10%, however, one component provides significantly more energy than the other. This occurrence is often just a side effect of less emitting or cheaper components being more heavily constrained (direct or indirect) within some ESSMOS.

4. $PCC > 0.7$ and $NMAE < K = 10\%$; Least interesting / most likely of no interest:

   This case marks the least interesting one. A PCC value greater than 0.7 again implies a relatively synchronous dispatch pattern. Combined with a low $NMAE$, this often translates to similarly behaving components between the compared ESSMOS.

Table 4.4 shows example results taken from the CompE comparative result analysis with PPC and NMAE results, as well as corresponding interest levels as calculated by `tessif.identify`.

Table 4.4.: Example Compilation of the Statistical Timeseries Identification Results. Part of the CompE ISD Analysis; *oemof* Results Used as Reference. Results Calculated Automatically by `tessif.identfy`.

#	Flow Source	Flow Target	Parameter	*Calliope*	*FINE*	*oemof*	*PyPSA*
1	Battery	Powerline	Interest	low	low	low	high
			PCC	1.00	1.00	1.00	0.03
			NMAE	0.01	0.00	0.00	1.26
2	Combined Cycle PP	Powerline	Interest	low	low	low	high
			PCC	1.00	1.00	1.00	0.23
			NMAE	0.02	0.00	0.00	39.15
3	Hard Coal CHP	Heatline	Interest	None	None	None	high
			PCC	0.00	0.00	0.00	0.00
			NMAE	None	None	None	inf
4	Hard Coal PP	Powerline	Interest	None	None	None	None
			PCC	0.00	0.00	0.00	0.00
			NMAE	None	None	None	None

Entry one and two represent normal (non-edge-case) examples, where the *Caliope*, *FINE* and *oemof* results are equal/very similar ($PCC = 0$ and $NMAE \leq 0.02$) and the *PyPSA* results are marked as highly interesting with regard to *oemof*'s. The None and inf values in entries three and four correspond to a set of possible edge-cases occurring when using equations 2.9 and 2.18 to calculate the PCC and NAME values as described:

1. NMAE = None:

When reference series and result series are zero at all entries, normalisation calculation for the NMAE can be expressed as zero divided by zero, which is, in turn, calculated to None by the underlying software. This happens for unused components, especially in emission constraint expansion problems where high-emitting components often have to remain unused. A corresponding example can be seen in Table 4.4, entry 4. The `Hard Coal PP` component remains unused, and thus, the time series results are zero for all entries among all ESSMOS and their corresponding NMAE results are evaluated to None.

2. NMAE = inf:

   In cases where the reference time series is all zeroes, but the result is not, the NMAE normalisation can be expressed as non-zero divided by zero, which gets evaluated to infinity by the underlying software. A respective example can be seen in Table 4.4, entry 3, where the `Hard Coal CHP` NMAE results are inf for *PyPSA*. For the other tools the NMAE result is None as described above.

3. All PCC = 0:

   The PCC value for all result sets is evaluated to zero in cases where the reference time series is all zeroes. As can be seen, however, by the example results in Table 4.4, entries 3 and 4, the corresponding $NMAE$ edge cases still allow distinguishing between potential and no interest.

For **identifying significant differences** in **static** or **singular value** component results, an interest evaluation is proposed based on a simple comparison of relative deviations. To calculate the relative deviation Δ_{rel}, the reference result REF and the compared result value RES are utilised as shown in equation 4.1. Therefore, the relative deviation Δ_{rel} is interpreted as the relative difference between a calculated result and a selected reference.

$$\Delta_{rel} = \frac{RES - REF}{REF} \tag{4.1}$$

A relative deviation value of 0.0 corresponds to no difference between REF and RES. $\Delta_{rel} = 0.5$ means the selected reference is 50% larger than the calculated result. A relative deviation of -0.8 indicates that the calculated result is 80% smaller than the reference. Table 4.5 lists the assumed interest levels in the first column and their respective default threshold ranges in the second column.

Table 4.5.: Thresholds and Interest Levels for Identifying Static Differences Using Equation 4.1.

Interest Level	Threshold
high	$0.30 \leq \text{abs}(\Delta_{\text{rel}})$
medium	$0.10 \leq \text{abs}(\Delta_{\text{rel}}) < 0.30$
low	$\text{abs}(\Delta_{\text{rel}}) < 0.10$

Table 4.6 shows example results taken from the CompE CRA with both relative deviation results and corresponding interest level as calculated by `tessif.identify`.

Entry one of Table 4.6 represents a normal (non-edge-case) example where the *Caliope*, *FINE* and *oemof* results are equal ($\Delta_{rel} = 0$) and the *PyPSA* results are 95% larger than *oemof*'s. The None and inf entries two, three and four correspond to a set of possible edge-cases occurring when using equation 4.1 to calculate the relative deviation as described:

Table 4.6.: Example Compilation of the Statistical Static Identification Results.
Part of the CompE ISD Analysis; *oemof* Results Used as Reference.
Results Calculated Automatically by `tessif.identfy`.
Rel. Dev. Corresponds to Δ_{rel} in Equation 4.1.

#	Component	Parameter	*Calliope*	*FINE*	*oemof*	*PyPSA*
1	Battery	Interest	low	low	low	high
		Rel. Dev.	0.0	0.0	0.0	0.95
2	Gas Station	Interest	low	low	low	None
		Rel. Dev.	0.01	0.0	0.0	None
3	Hard Coal Supply	Interest	None	None	None	high
		Rel. Dev.	None	None	None	inf
4	Lignite Supply	Interest	None	None	None	None
		Rel. Dev.	None	None	None	None

1. $\Delta_{rel}(REF) = 0$ and $\Delta_{rel}(RES) =$ None; entry two in Table 4.6:

   The component used for calculating the relative deviation is present within the reference results but not within the results for which the relative deviation is to be calculated. In this case, the relative deviation of the reference result is calculated to zero, whereas the missing components result is set to None. This occurrence can be observed in Table 4.6 entry two, where the reference deviation of *oemof* is calculated to zero and the *PyPSA* deviation to None, since *PyPSA*'s ESSM does not include the `Gas Station` component.

2. $\Delta_{rel}(REF) =$ None and $\Delta_{rel}(RES) =$ inf; entry three in Table 4.6:

   The utilised reference result REF equals zero, whereas the result RES for which the relative deviation is to be calculated does not equal zero. The overall calculation can then be expressed as zero divided by zero for REF and non-zero divided by zero for RES, which the underlying software calculates to None for REF and to infinity for RES . Such an occurrence can be observed in Table 4.6, entry three. The results for the `Hard Coal Supply` component of *Calliope*, *FINE* and *oemof* equal zero and the *PyPSA* result does not equal zero. The relative deviation for *Calliope*, *FINE* and *oemof* can thus be expressed as zero divided by zero, which gets evaluated to None. *PyPSA*'s result on the other hand unequals zero and the relative deviation can subsequently be expressed as non-zero divided by zero, which gets evaluated to inf.

3. $\Delta_{rel}(REF) = \Delta_{rel}(RES) =$ None; entry four in Table 4.6:

   The utilised reference results REF, as well as the results RES for which the relative deviation is to be calculated, are both infinite. For example, this can occur when a commodity component's installed capacity is unconstrained and therefore modelled as infinite. The overall calculation to be performed then equals infinity divided by infinity, which is calculated to None by the underlying software. Table 4.6, entry four, shows such occurance, where the `Lignite Supply` is modelled as an infinite commodity for all ESSMOS.

4.5.3.2. ISD-Step 2: Detecting and Isolating Differing Timeframes

The goal of identifying differing timeframes is to isolate those time steps from many potential time steps, causing the differences and thus reducing the amount of data needed to be investigated. Different timeframes are usually of secondary interest when performing the method as recommended for pure comparative reasons like in this thesis since root cause hypothesis formulation can typically be done independently. However, manual inspection of the isolated differences, and thus that of differing timeframes, is often of significant interest for actual application-oriented modelling tasks. This is particularly the case when focusing on time series results like dispatched power or resulting state of charge profiles.

The subsequent measurement is proposed to identify relevant timeframes for the components selected in Paragraph 4.5.3.1. For components exceeding a calculated NMAE that surpasses the error value threshold, timeframes are isolated by extracting all continuous sequences where such exceedance occurs, as described by the following equation:

$$\{t \mid \left|\frac{REF(t) - RES(t)}{REF(t)}\right| \geq K\}. \tag{4.2}$$

Manually selected components or components statistically identified as medium interest can have an average time series deviation less than K (see Table 4.3). Isolating specific timeframes using the relation in equation 4.2 can still be done. In this case, a good initial threshold value for K is the arithmetic mean of the reference timeseries results $REF(t)$. In case of unsatisfactory results, K can be increased to sort out smaller deviations or decreased to include smaller deviations.

Figure 4.2, sourced from the *Tessif* online documentation[1] [61], illustrates an example where a timeframe of medium interest power flow is identified and visualised. It is important to note that, by default, only deviations exceeding the threshold value K are plotted as such. All other deviations are disregarded, and the corresponding reference results are used (*oemof* in this example). Consequently, no *Calliope* step plots results are depicted in Figure 4.2, as no above-threshold deviations were detected within the particular timeframe. However, Calliope is included in the comparison, as indicated by its legend entry.

[1] Online documentation of *Tessif* concerning the identification of differences https://tessif-phd.readthedocs.io/en/latest/getting_started/examples/application/identification.html

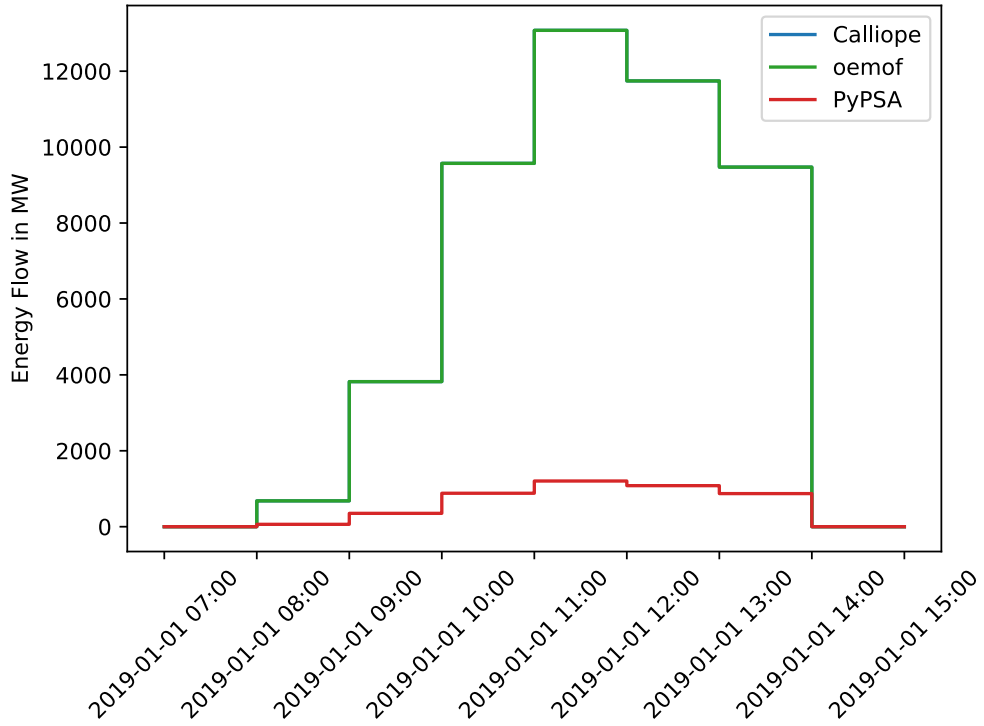

Figure 4.2.: Identified Differing Timeframe Example: Taken From *Tessif*'s Example Library. Created Using the `tessif.identify` Module.
Displaying Only the Above-Threshold Time-Varying Results Compared to *oemof*.

4.5.3.3. ISD-Step 3: Selection for Further Analysis

The further analysis selection as the final ISD-step aims to formulate hypotheses on potential root causes of the observed differences. Subsequently, deriving a set of questions aimed to be answered by CRA-step four. This outcome is typically achieved by inspecting the high-priority results in combination with the preliminary result analysis in case it is performed. While closely investigating the components isolated during the ISD analysis. This usually implies interpreting calculated PCC and NMAE values and comparing them to the respective result visualisations. It was found that deviations often stem from differing cost or emission allocation approaches, which are reliably detected using this method, enabling accurate hypotheses formulation.

4.5.4. CRA-Step 4: Analysis for Identifying Root Causes (IRC)

Subsection 4.5.4 proposes a set of distinctive investigations suited to identify potential root causes (IRC) of differences isolated prior. These investigative techniques consist of a tabular parameter comparison (Paragraph 4.5.4.2), a so-called parameter check (Paragraph 4.5.4.3), and a plausibility check (Paragraph 4.5.4.4). The IRC analysis begins with an investigative scenario formulation (Paragraph 4.5.4.1) and ends with a final assessment (Paragraph 4.5.4.5).

The IRC analysis focuses on specific components or parameter subsets and is typically quite abstract compared to the preceding CRA-steps. **The main objective of the IRC analysis is to verify falsify the plausibility of the formulated hypotheses.** The proposed procedure is, therefore, to be performed separately for each hypothesis and component. This thoroughness potentially introduces substantial work, emphasising the benefit of a carefully executed preliminary and

comparative results analysis in preparation.

4.5.4.1. IRC-Step 1: Investigative Scenario Formulation

The main objective of the investigative scenario formulation is to derive actionable tasks from the questions posed during the further analysis selection phase of the identifying significant differences analysis. This procedure guides and streamlines the remaining root cause identification process by abstracting the process as much as possible from the actual optimisation results to create an independent and manageable set of tasks.

4.5.4.2. IRC-Step 2: Performing a Tabular Parameter Comparison

The first main question for identifying potential root causes when using an input data harmonisation process like, for example, that of *Tessif* most likely resembles the following.

Does the harmonised data input provide all parameters expected, and do all ESSMOS offer corresponding implementations?

A systematic comparison of the respective component parameters and input data is advised in order to give an educated answer to this question.

The most practical approach is to create a set of tables for each component stating if and how the relevant input parameters are respected. *Tessif*'s online documentation [61] provides an extensive compilation of component clustered tables comparing all parameters of each ESSMOS supported.

The advantage of this method is that once the tables are established, they rarely have to be updated, and the tabular parameter comparison (TPC) is conducted within minutes, simplifying the manual comparison tremendously. If the observed differences stem from different implementations or parameter interfaces, they are most likely identified in this way. However, the initial amount of work is quite substantial, and it might not be feasible to create those tables in cases where only rough estimations are required. They, however, help familiarise oneself with technical software details, potentially bringing additional secondary benefits, requiring balancing out gain and effort. In the process of conducting the underlying research and developing *Tessif*, these tabular parameter comparison tables have proven invaluable and worth the effort of creating.

4.5.4.3. IRC-Step 3: Performing a Parameter Check

With the tabular parameter comparison verifying the existence of adequate parameter interfaces in both data input and ESSMOS, the next logical step is to check the actual parameter passing and parsing capabilities. **The parameter check's main objective is thus verifying the intended parameter passing and parsing.** This goal is achieved by employing introspection of the input data interface (like *Tessif*) and the respective ESSMOS, implying the necessity of consulting the corresponding ESSMOS documentation or code bases, thus requiring some programming experience and skill.

In addition to the parsing information gathered, sometimes complementary knowledge about the internal relations of the inspected parameters can be gained during this introspection phase. This guided research potentially results in additional key revelations.

4.5.4.4. IRC-Step 4: Performing a Plausibility Check

With the tabular parameter comparison and the parameter check verifying the existence and intended parameter handling, two central questions for comparing individual components between ESSMOS remain:

What is the expected behaviour, and does it differ between ESSMOS?

Lüdcke [71] developed a systematic analysis technique to answer these. **This technique isolates the individual components into small functional ESSMs to expose them to carefully designed scenarios provoking component-specific behaviours known beforehand. All parameters, components and constraints in these MSCs are explicitly set for this sole purpose.** Thus revealing the ESSMOS' defacto implementations. This technique is called plausibility check since the formulated description could be expressed in more simple terms as: "Is the component doing what it is supposed to do for how it is parameterised?"

Using this technique, different component behaviours due to implementation or parameterisation reasons are often easily identified. It is also of great utility during the integration process of new ESSMOS into an existing means of an input data harmonisation effort like that of *Tessif*. The respective transformation, passing and parsing of data can often be cumbersome and error-prone in these contexts. Using the plausibility check, however, potential implementation errors are identified quickly and reliably. *Tessif*'s components described in Section 3.2 were checked for plausibility this way. A comprehensive tabular overview of the results can be found in in the work of Lüdcke [71].

4.5.4.5. IRC-Step 5: Conducting a Final Assessment

The last step of the proposed identifying root causes analysis is to utilise each finding of the checks performed beforehand to make a final assessment. Apart from verifying or falsifying prior formulated hypotheses, a coherent chain of argumentation is to be built to facilitate sharing of the findings. In addition, an estimation is to be given whether additional model scenario combinations are to be investigated and how to handle observed differences and the respective root causes.

4.5.5. CRA-Step 5: Performing a Detailed Individual Analysis (DIA)

The final step of the recommended comparative result analysis involves a manual detailed individual analysis (DIA) to address any remaining questions. Occasionally, case and result-specific questions persist, which can be formulated similarly to the following:

Now, with precise knowledge of software and component internal parameter handling, do the observed results align as expected?

Checking residual ISD hypotheses ties specifically into the investigative scenario formulation and is highly case-dependent. Often, such checks are beneficial for connecting the abstract IRC analysis to the inspected model scenario combination results. It is frequently required for extensive or complex ESSMs or when assessing specific factors, such as the dynamic behaviour of certain components. Techniques such as examining load profiles and load duration curves can be beneficial and may be employed using *Tessif*'s visualise module.

When necessary, the information collected through the comparative result analysis should provide valuable starting points for additional investigations. In cases where the sources of observed differences remain unclear, several elements may be excluded from the manual inspection due to the preceding analysis. These include:

- Differing ESSM formulations, which would be detected using the advanced system visualisation
- Differing cost or emission allocations, which would be detected using the high-priority results in combination with the advanced system visualisation
- Differing component implementations, which would be detected using the analysis for identifying root causes

This understanding typically suggests that potential causes may include:

- Non-apparent differences in scenario formulations, encompassing secondary global constraints such as emissions or land usage
- Human-induced errors in the comparison process
- Non-apparent implementation errors within the comparative framework or ESSMOS

It is recommended to examine these factors initially before undertaking further analysis.

5. Conducted Case Study to Showcase the Use of the Comparative Method

Chapter 5 conducts an exemplary case study to showcase the entirety of the developed comparative method. It directly addresses Goal 3.

The objectives of this chapter are:

1. Combining the theoretical foundation of Chapter 2 with the created software framework of Chapter 3 to execute the developed method of Chapter 4
2. Showcasing how to leverage the developed preselection procedure to identify suitable ESSMOS based on limiting the research effort to the most relevant facts
3. Demonstrating how to exploit the result analysis techniques of the comparative-method-steps 3 and 4 to quickly sift through large amounts of result data to isolate key differences and to quickly identify their root causes

The chapter does NOT aim at achieving the following points:

1. Evaluating the suitability of the developed energy supply system models (ESSMs) to model existing energy systems
2. Analysing the results of the used model scenario combinations (MSCs) as if they were designed to model existing energy supply systems

5.1. Free and Open-Source (FOS) Energy Supply System Modelling and Optimisation Software (ESSMOS) Compared in this Case Study

All of the compared ESSMOS are currently integrated into *Tessif*, thus facilitating an automated comparison process involving these tools. Additional, detailed information beyond this brief introduction can be found in *Tessif*'s online documentation [61] and the documentation of the individual ESSMOS as marked in their respective subsections. The case study was conducted using the PhD version 0.0.1 of *Tessif* [62].

All compared ESSMOS function similarly regarding the general steps necessary to formulate and optimise an ESSM, following the general structure discussed in Subsection 2.1.8. The individual ESSMOS provide a unique programmatic interface to create their ESSM by providing predefined building blocks or components similar to what is described in Section 3.2 of how *Tessif* is constructed. The subjectively perceived degree of similarity varies between very similar for *Tessif* compared to *oemof* to quite different for *Tessif* compared to *FINE*.

5.1.1. Calliope

Calliope is a Python-based multi-scale FOS ESSMOS, originally developed at the Department of Environmental Systems Science, part of the ETH Zürich [72]. Related publications can be found on the *Calliope* website [73].

To create an ESSM within *Calliope*, a collection of YAML (YAML Ain't Markup Language) and CSV (comma-separated value) files must be provided. With this attempt to separate ESSM data input and code, *Calliope* distinguishes itself from all other ESSMOS compared here. In its core, six central components, labelled as tech groups within the framework, are offered to build the model: `supply`, `supply_plus`, `demand`, `conversion`, `conversion_plus`, and `transmission`. In addition, a very convenient method is provided for reusing, expanding or pairing a particular ESSM to a collection of (different) scenarios. All in all, *Calliope* is to be seen as reasonably well-documented [74]. However, it was found that *Calliope* provides comparatively few online tutorials/examples and thus offer comparatively little help in familiarising oneself with the software.

Calliope distinguishes itself from the other ESSMOS by providing a built-in mechanism to reuse a given (cost) optimal system by calculating additional, near-optimal results, maximising the use of different technologies, thus providing a powerful feature for real-world modelling applications. Another uncommon but potentially helpful feature is a built-in time resolution adjustment functionality, albeit relatively simple. In addition, *Calliope* provides a pre-built-in option for using a rolling or receding horizon approach.

A comprehensive analysis of *Calliope* and its *Tessif* integration was done by Reimer [70], evaluating *Calliope* as a suitable and functional asset in the context of *Tessif* as the fourth ESSMOS included. This case study was conducted using *Calliope* version 0.6.6.post1 [75].

5.1.2. Framework for Integrated Energy Assessment (*FINE*)

FINE [76] is a FOS ESSMOS written in the Python programming language. It was developed at the *Jülich Research Center* within the *Energy System 2050* project as part of the *Helmholtz Energy Computing Initiative* (*HECI*) [77, 78].

FINE provides a programming interface for creating ESSMs consisting of five base components: `Source`, `Sink`, `Storage`, `Conversion` and `Transmission`. Additionally, the total emitted CO2 can be constrained. *FINE*'s programming interfaces features a well-structured, relatively easy-to-comprehend interface to create and adjust the ESSM and its underlying components. It is relatively well documented [79], and comprehensive examples are given to facilitate first contact. These include sector coupling as well as hydrogen integration, among others.

Parameterisation of *FINE* components allows for (in the context of the ESSMOS compared) unusually detailed cost definitions in the sense of per-default integration of the equivalent-annual-cost method. On the technical side of component parameterisation, *FINE* distinguishes itself from others by offering partwise linearisation [31], a built-in interface to use their in-house developed time series aggregation module (*tsam*) [80], and several additional tools to reduce the computational effort [81, 82, 83].

A detailed analysis of *FINE* and its integration into *Tessif* was done by Schnute [8], resulting in the assessment of *FINE* being a suitable and functional asset in the context of *Tessif* as the third tool to be incorporated. This case study was conducted using *FINE* version 2.2.1 [84].

5.1.3. Open Energy Modelling Framework (*oemof*)

The `solph` library of *oemof* is a Python-based FOS ESSMOS for optimising multi-regional ESSMs considering power, heat, and mobility [85, 86]. It was primarily developed at the *Center for Sustainable Energy Systems Flensburg*, the *Reiner Lemoine Institute Berlin* and the *Otto-von-Guericke-Universität, Magdeburg* [87]. Several related publications can be found on the *oemof* website [88].

Oemof provides a detailed, highly modular, customisable programming interface for creating ESSMs, offering five core components: `Bus`, `Source`, `Sink`, `Transformer` and `Storage` to do so. In addition, multiple custom components and experimental-state components like specialised CHP and storage components, as well as distinct electrical transport components. *oemof* offers an extensive example library, facilitating first usage significantly. Furthermore, *oemof* provides detailed documentation [89]. Due to several programmatic abstraction layers and project restructuring processes, however, noticeable amounts of faulty links and often convoluted descriptions have to be dealt with, not taking away any of *oemof*'s utility but requiring patience during initial contact.

Oemof distinguishes itself most notably by focusing on an open science, collaborative approach while offering (by far) the most detailed component models.

As part of the ESSMOS integrations, *oemof* was the first ESSMOS to be included in *Tessif* and therefore had a significant impact on its initial design. This case study was conducted using *oemof.solph* version 0.4.4 [90].

5.1.4. Python for Power System Analysis (*PyPSA*)

PyPSA is a Python-based FOS ESSMOS that provides a programmatic environment for state-of-the-art energy system modelling. It was initially developed by members of the *Frankfurt Institute of Advanced Studies* (*FIAS*), the *Karlsruhe Institute of Technology* (*KIT*) and the *Technical University of Berlin* (*TUB*) [91, 92].

PyPSA provides a python programming interface offering built-in components for ESSM creation, heavily focusing on electric grids. In addition, well-structured documentation [93] is provided, including a decent-sized example library, considerably helping in getting to know the framework. Despite its open-source structure and good documentation, however, extending the core functionalities is not part of *PyPSA*'s inherent design, making it difficult, for example, to add custom components. Consequently, the author subjectively perceives it as less open and collaborative than, for example, *oemof*.

PyPSA distinguishes itself from the other ESSMOS compared here by heavily focusing on electrical grid components, even offering the possibility of reusing the optimisation problem formulated system ESSM to perform a load flow analysis.

As part of the ESSMOS integrations, *PyPSA* was the second tool to be integrated into *Tessif*. This case study used version 0.19.3 [94].

5.2. Executing CM-Step 1: Preselection of Suited FOS ESSMOS

Performing comparative-method-step one, the preselection, consists of three preselection-steps. The first is creating a list of possible candidates (Subsection 5.2.1). The second is comparing these canditates' key criteria (Subsection 5.2.2) to narrow down the number of sensible choices. The third and final preselection-step is conducting a detailed a priori comparison using the developed factsheet (Subsection 5.2.3).

5.2.1. Executing Preselection-Step 1: Creating a List of Potential Candidates

The list of candidates used in this case study consists of the following four ESSMOS (listed in alphabetical order):

- *Calliope*
- *FINE*
- *oemof*
- *PyPSA*

Since the list of candidates in this case study is predefined, the preselection phase is somewhat redundant. It is illustrated here, nonetheless, to demonstrate this method's proper usage.

5.2.2. Executing Preselection-Step 2: Key-Criteria-Based Reduction in the Number of Candidates

Tables 5.1 to 5.3 illustrate the result of executing preselection-step two as developed in Subsection 4.2.2. **The specific assembly of the most relevant facts as key-criteria in these tables is considered as the primary result in this context**. For an inexperienced ESSMOS user, collecting all key-criteria information takes about one to three hours per ESSMOS. The data displayed exemplifies this process and demonstrates this technique's effectiveness by drawing comprehensive conclusions based on the tabular results. However, these are only considered as beneficial side-effects within this thesis since developing and showcasing the Comparative Method itself is the primary goal and since the tools used for the comparison are pre-defined by *Tessif*'s currently supported ESSMOS tools.

1. If a general-purpose tool is necessary, employing *oemof* is advised, as it emphasises extensibility and broad applicability rather than concentrating on a specific field of application (see the first and fourth category of the author's intent criterion in Table 5.1).
2. *Calliope* is recommended if many scenario formulation variations are to be performed on the same ESSM. *Calliope* provides a pre-built-in interface for separating model and scenario formulation with the option to automate this process (see the fourth category of the author's intent criterion in Table 5.1).
3. Suppose the primary need is to extend an existing framework when adding a few specialised components, for example in a short project. Then the use of *oemof* is recommended since it is the most open and extensible (see the fourth category of the author's intent criterion in Table 5.1).

4. The use of *oemof* is also recommended in case a community-driven software tool is to be used or supported, for example in the context of a non-profit organisation (see categories two to four of the author's intent criterion in Table 5.1).
5. All compared ESSMOS can be used equally, independent of the modelling requirements for specific temporal or spatial resolution (see the resolution criterion in Table 5.1).
6. *Fine* is the only compared tool that provides automated spatial aggregation capabilities (see the spatial aggregation category of the reduction of computational effort criterion in Table 5.2).
7. FINE with its built-in intermittency functionalities is recommended for optimising a large time span with numerous time steps, (see the intermittency category of the reduction of computational effort criterion in Table 5.2). Alternatively, *Calliope*'s time resolution adjustment feature may be employed (refer to the temporal aggregation category of the reduction of computational effort criterion in Table 5.2).
8. To investigate dispatch optimisation characteristics of a rolling horizon, in, e.g., day-ahead market optimisations, *Calliope* or *FINE* are recommended (see the rolling horizon category of the reduction of computational effort criterion in Table 5.2).
9. For large-scale investigations, *FINE* is recommended due to its extensive suit of built-in routines for reducing computational effort (see the scalability criterion in Table 5.2).
10. In case a detailed comparison is to be performed about the benefits of different means of reduction in computational effort, *FINE* is recommended for the same reason (see the reduction of computational effort criterion in Table 5.2).
11. In case detailed commitment problems are to be formulated using combined heat and power plants or storage systems, *oemof* might be the best choice since it offers the highest amount of detail (see the CHP and energy storage components criteria in Table 5.3) and is the easiest to expand (see the fourth category of the author's intent criterion in Table 5.1).
12. In case modelling detailed electrical energy transportation is of primary interest, then *PyPSA*, followed by *oemof*, are the most promising candidates (see the passive category of the energy transport criterion in Table 5.3).
13. If the modelled storage components must not have a cyclic state of charge constraint, then *FINE* should not be used (see the cyclic SOC constraint category of the energy storage components criterion in Table 5.3).
14. If the modelled storage components necessitate a non-zero initial state of charge, the use of *FINE* is not advised (see the initial SOC constraint category of the energy storage components criterion in Table 5.3).
15. *Calliope* should not be used in cases where an ESSM comprises storage components with varying cyclic SOC constraints (see the individual cyclic SOC constraint category of the energy storage components criterion in Table 5.3).

Table 5.1.: Key Criteria Collection of the Compared ESSMOS — Part One.
Resulting Table as Output of the Performed Preselection-Step 2 Developed in Subsection 4.2.2.

Criterion	Category	*Calliope*	*FINE*	*oemof*	*PyPSA*
	1	Suited for planning energy systems at scales ranging from urban districts to entire continents	Suited for large scale energy system modelling	Suited for all-purpose energy system modelling	Suited for simulating and optimising modern power systems
The Author's Intent	2	Provides clear separation of framework (code) and model (data)	Provides multiple locations and commodities	Aims at being community driven	Provides combining alternating and direct current networks
	3	Provides distinct flexibility	Provides distinct ROCE techniques	Provides a distinct modular design	Provides support for large scale power system analysis
	4	Provides possibility to execute many runs based on the same base model	Offers a very scientific approach to its ESSM design	Aims at being highly extensible	
Reported Suitability		Provides high temporal and spatial resolution	Provides high temporal and spatial resolution	Provides high temporal and spatial resolution	Provides high temporal and spatial resolution
Resolution	Temporal	Allows user defined time steps	Allows user defined time steps	Allows user defined time steps	Allows user defined time steps
	Spatial	Unconstrained/ use case dependent	Unconstrained/ use case dependent	Unconstrained/ use case dependent	Unconstrained/ use case dependent

Table 5.2.: Key Criteria Collection of the Compared ESSMOS — Part Two.
Resulting Table as Output of the Performed Preselection-Step 2 Developed in Subsection 4.2.2.

Criterion	Category	*Calliope*	*FINE*	*oemof*	*PyPSA*
Scalability	Reported suitability	Provides rudimentary straightforward ROCE techniques	Provides multiple ROCE techniques and is designed to scale well	Provides no distinct pre-built-in features	Provides alternative problem formulations to reduce the computational effort
	Subjective impression	Perceived as good	Perceived as excellent	Perceived as decent	Perceived as good
Reduction Of Computational Effort	Temporal aggregation	Provides pre-built-in time-resolution-adjustment	Achieved manually by adjusting time step length	Achieved manually by adjusting time step length	Achieved manually by adjusting time step length
	Spatial aggregation	Achieved manually by representing multiple plants within one component	Provides pre-built-in functionalities	Achieved manually by representing multiple plants within one component	Achieved manually by representing multiple plants within one component
	Intermittency	Is not pre-built-in but realisable via user-coded extensions	Provides pre-built-in tsam interface	Is not pre-built-in but realisable via via user-coded extensions	Is not pre-built-in but realisable via user-coded extensions
	Rolling horizon	Is pre-built-in	Provides pre-built-in myopic forecast approach	Is not pre-built-in but realisable via user-coded extensions	Is not pre-built-in but realisable via user-coded extensions

Table 5.3.: Key Criteria Collection of the Compared ESSMOS — Part Three.
Resulting Table as Output of the Performed Preselection-Step 2 Developed in Subsection 4.2.2.

Criterion	Category	*Calliope*	*FINE*	*oemof*	*PyPSA*
CHP Components	Pre-built-in CHP component template	Requires user construction using other existing component templates	Requires user construction using other existing component templates	Provides multiple pre-built-in CHP component templates	Requires user construction using other existing component templates
	Inflow constraint only	Yes	Yes	In- and outflow constraining possible	Yes
	Pre-built-in field of operation	-	-	Yes	-
Energy Storage Components	Initial SOC	Can unequal zero	Must be zero	Can unequal zero	Can unequal zero
	Type of emission allocation	Emission allocation to outflow	Emissions allocation to outflow	Emission allocation to outflow	Emission allocation to differences in initial and final SOC
	Cyclic SOC constraint	No mandatory cyclic SOC constraint	Mandatory cyclic SOC constriant	No mandatory cyclic SOC constraint	No mandatory cyclic SOC constraint
	Individual cyclic SOC constraint	Must be the same for all storage components	Can be individually set by each storage component	Can be individually set by each storage component	Can be individually set by each storage component
Energy Transport	Passive	-	-	Experimental, electrical components	Very detailed electrical components
	Active	Carrier-independent transport components	Carrier-independent transport components	Carrier-independent transport components	Carrier-independent transport components

5.2.3. Executing Preselection-Step 3: Factsheet-Based ESSMOS Selection

The main goal of preselection-step three is to further reduce the list of candidates if required. **Since no further reduction is desired here, preselection-step three is executed solely for demonstrative purposes.** The results of this factsheet-based comparison are listed in Tables A.4 to A.6 in Appendix Section A.3. An excerpt of the additionally gained highest-value information is summarised in Table 5.4 to draw the following conclusions:

1. In case using an up-to-date ESSMOS is of primary importance (when comparing modern ESSMOS tools, for example), all compared tools are to be recommended. By the time of writing this thesis, no update was older than two months (see the basic information category in Table 5.4).
2. If a non-Pyomo solver interface is necessary, for instance when comparing Pyomo-interfaced and non-Pyomo-interfaced computational resources, *PyPSA* is recommended due to its option for using its own solver interface in addition to the commonly shared Pyomo interface (refer to the solver interface parameter in Table 5.4 under the programmatic formulation category).
3. For cases prioritising simplicity of data input, such as in student courses, *Calliope* is advised due to its deliberate separation of model-building and input data via a YAML interface for input data (refer to the input data parameter in Table 5.4 under the programmatic formulation category).
4. On the other hand, *PyPSA* is recommended when requiring high flexibility of input data formats (refer to the variety of supported input data format types in Table 5.4 under the programmatic formulation category).
5. When aiming to utilise non-linear problem formulations (NLP), *FINE*, *PyPSA* or *oemof* are recommended since they all offer some built-in linearisation options (refer to the mathematical description category in Table 5.4).
6. For inexperienced programmers, having a larger selection of predefined, specialised component templates is often helpful. In this case, *oemof* and *PyPSA* are recommended since they provide the most pre-built-in and specialised component templates (compare the number of distinct component template names in the modelling approach category in Table 5.4).
7. If supply profiles are to be generated, *Calliope*, *PyPSA* and *oemof* are likely candidates. They provide tools to assist in supply profile synthesis independently of the actual ESSMOS. However, since they are developed specifically to be used with their ESSMOS, synergies are to be expected, especially in the form of reduced learning curves due to similar programming and documentation (refer to the supply profile generation parameter in Table 5.4 under the renewable energies category).
8. If demand profiles are to be generated, the same applies to *FINE* and *oemof* since they provide in-house tools to assist in load profile synthesis (refer to the demand profile generation parameter in Table 5.4 under the renewable energies category).
9. If the advancement of uniform scientific terminology is of interest, e.g., for open science communities or committees, then the use of *oemof* is recommended since they are already undergoing the effort (refer to the collaborative properties category in Table 5.4).

Table 5.4.: Excerpt of the Additional Most Valuable Information Acquired by the Factsheet Comparison. Resulting Table as Output of the Performed Preselection-Step 3 Presented in Subsection 4.2.3. Factsheet-Based Comparison Originally Developed by KÖRBER [66]. Full Results Listed in The Appendix Section A.3.

Category	Parameter	*Calliope*	*FINE*	*oemof*	*PyPSA*
Basic Information	Last Update	2023-01-18	2023-01-17	2023-02-03	2023-02-16
Programmatic Formulation	Input Data	YAML	Python code	Python code	Python code CSV, HDF5, nCDF
	Solver Interfaces	Pyomo	Pyomo	Pyomo	Pyomo & Own
Mathematical Description	Optimisation Problem Formulation	LP	LP, MILP, NLP	LP, MILP, NLP	LP, MILP, NLP
Modelling Approach	Native Components	Conversion, Conversion-plus Demand Storage Supply, Supply-plus Transmission	Conversion Source Sink Storage Transmission	Bus, ELine, ExCHP, GenStor, GenCAES, OffTran, Sink, SiDSM Source, Storage, Tran	Bus, Generator, Line, Link, Load, ShuntImpedance, StorageUnit, Store Transformer
Renewable Energies	Supply Profile Generation	Yes (*solar-and-wind-potentials*)	No	Yes Yes (*feedinlib*)	Yes Yes (*atlite*)
	Demand Profile Generation	No	Yes (LoadProfile Generator)	Yes Yes (*demandlib*)	No
Collaborative Properties	Terminological Uniformity	Medium	High	Very High	High

5.3. Executing CM-Step 2: Modelling and Optimising Suited Model Scenario Combinations (MOSMSC)

Performing CM-step 2, the modelling and optimisation of suited model scenario combinations (MOSMSC), consists of three MOSMSC-steps. The first is to create or choose suited MSCs (Subsection 5.3.1). The second is visualising the ESSM as a generic graph to perform a simple plausibility check of the model structure and to get a visual impression of its complexity (Subsection 5.3.2). MOSMSC-step 3 optimises the selected MSCs for all compared ESSMOS tools (Subsection 5.3.3).

5.3.1. Executing MOSMSC-Step 1: Selecting Suited MSCs

Table 5.5 gives an overview of the selected MSCs which are discussed in detail in Subsections 3.8.1 and 3.8.2. The decision to add combination 2a, the Modified CompE, was made while performing the root cause analysis in Subsection 5.5.1 during the case study. It is listed here already to maintain coherence and facilitate comprehension.

Table 5.5 helps navigate this work by providing backlinks to the GSVs of the underlying ESSMs (column *(6)*) and linking the respective execution of CM-steps three and four (columns *(7)* and *(8)*). It furthermore summarises the chosen model scenario combinations in column *(5)*, and clarifies how the respective alias is constructed using the ESSM name (column *(3)*) and problem formulation archetype (column *(4)*). Additionally, the design focus is stated in column *(1)*, and an MSC-number is assigned in column *(2)*.

5.3.2. Executing MOSMSC-Step 2: Initial Generic System Visualisation (GSV)

The GSVs of the used MNSCs are shown in Figures 3.4 to 3.6 of Section 3.8, close to where the MSCs are first introduced.

5.3.3. Executing MOSMSC-Step 3: Optimising the Selected MSCs

MOSMSC-step 3 encompasses the optimisation of the chosen MSCs using each ESSMOS to be compared. Within the scope of this work, this is achieved by either using the `tessif.optimize` functionalities directly (see also Paragraph 3.4.1.2) or by automating the entire comparison process and using the corresponding `tessif.compare` capabilities (Paragraph 3.4.1.3).

Table 5.5.: Selected Model Scenario Combinations (MSCs) as the Result of Performing MOSMSC-Step 1.
The General System Visualisation of the Underlying ESSMs are Depicted as Referenced in Column *(6)*.
The MSCs are Used in this Case Study to Perform Comparative-Method-Steps 3 and 4 as Referenced in Columns *(7)* and *(8)*.
All MSCs Were Specifically Designed for Comparing ESSMOS Tools.
Component-Focused MSCs Motivated, Created and Evaluated by Reimer [63].
Grid-Focused MSCs Motivated, Created and Evaluated by Hanke [64].

Design Focus *(1)*	**MSC-Nmbr.** *(2)*	**ESSM Name** *(3)*	**Problem Formulation Archetype** *(4)*	**Combination Alias** *(5)*	**General System Visualisation** *(6)*	**CM-Step 3 The Preliminary Result Analysis** *(7)*	**CM-Step 4 The Comparative Result Analysis** *(8)*
Component	1	Component	Commitment	Component-Commitment (**CompC**)	Figure 3.4	Subsection 5.4.2	-
	2	Component	Expansion	Component-Expansion (**CompE**)		Subsection 5.4.3	Subsection 5.5.1
	2a	Modified Component	Expansion	Modified Component-Expansion (**Modified CompE**)		-	Subsection 5.5.2
Grid	3	Lossless	Commitment	Lossless-Commitment (**LossLC**)	Figure 3.5	Subsection 5.4.4	-
	4	Transformer-Sinks	Commitment	No-Congestion Transformer-Commitment (**No-Congestion TransC**)	Figure 3.6	Subsection 5.4.5	-
	5	Transformer-Sinks	Commitment	Congestion Transformer-Commitment (**Congestion TransC**)		Subsection 5.4.6	-
	6	Transformer-Sinks	Expansion	Transformer-Expansion (**TransE**)		Subsection 5.4.7	-

5.4. Executing CM-Step 3: Preliminary Result Analysis (PRA) of the Optimisation Results

Performing CM-step 3, the PRA, consists of six PRA-steps. The result discussion of these PRA-steps is clustered around each investigated model scenario combination to facilitate comprehension.

5.4.1. Executing PRA-Step 1: Performing The Developed Result Data Assignment

The result data assignment is realised as an automated process in *Tessif*. After successful optimisation, the respective post-processing algorithms are called to execute PRA-step 1 without requiring any user-initiated actions.

5.4.2. Preliminary Result Analysis of the Component-Commitment (CompC) MSC

The CompE MSC, listed as MSC-number two in Table 5.5, is a non-emissions-constraint commitment problem. Therefore, the installed capacities are known before the optimisation, and the emission results do not impact the optimal solution. Hence PRA-steps four and five are skipped in this preliminary result analysis.

5.4.2.1. Executing PRA-Step 2: Inspecting the CompC High-Priority Results (HPR)

Figure 5.1 shows the CompC HPR of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are listed in Appendix Table A.7. The horizontal axis depicts the high-priority results discussed in Subsection 4.4.2. The vertical axis shows the respective numerical result values in relation to *oemof* as coloured bars representing the individual ESSMOS results.

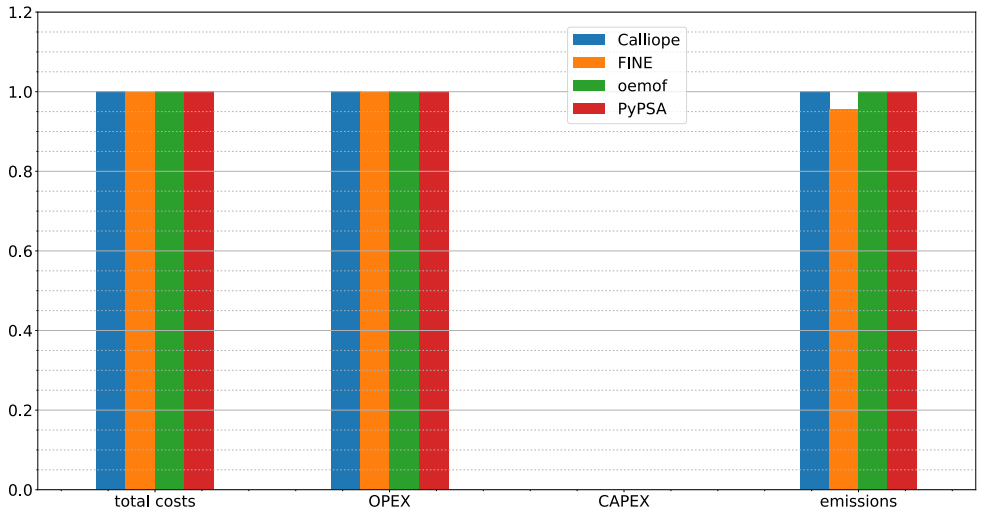

Figure 5.1.: HPR of the Compared ESSMOS Relative to *oemof* for the CompC MSC. The Depicted Numerical Results Are Listed in Appendix Table A.7.

The relevant aspects to note are:

- None of the ESSMOS calculate CAPEX results, as the CompC-MSC is of the commitment problem formulation archetype (no bars are drawn for the CAPEX results in Figure 5.1).
- The total cost and OPEX results are identical for all ESSMOS (compare all coloured bar lengths of the respective high-priority results in Figure 5.1).
- The *FINE* emission results are approximately 0.95 those of the other ESSMOS (compare the orange bar length of the emissions results to the differently coloured bar lengths of the same category in Figure 5.1).

5.4.2.2. Executing PRA-Step 3: Inspecting the CompC Summed-Up Loads Results

Figure 5.2 shows the CompC summed-up loads results of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are shown in Appendix Table A.8. The vertical axis depicts the edges of the respective flows as (`source`, `target`). The horizontal axis shows the numerical result values relative to *oemof* as coloured bars representing the individual ESSMOS results. Figure 5.2 only depicts edges with differing results.

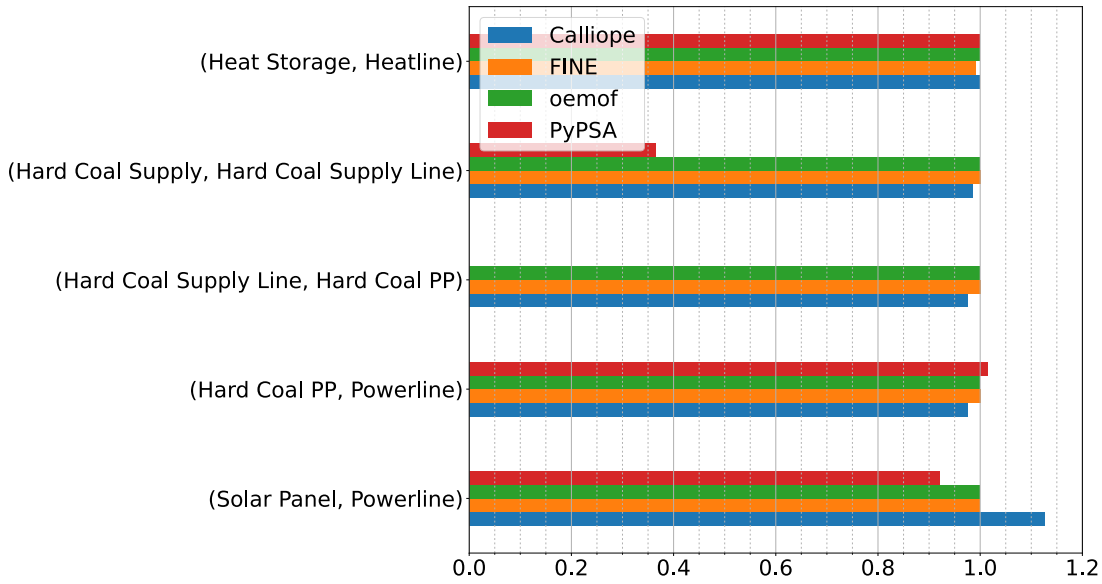

Figure 5.2.: Summed-Up Loads Results of the Compared ESSMOS Relative to *oemof* for the CompC MSC. Only the Differing Results are Visualised.
The Depicted Numerical Results Are Listed in Appendix Table A.8.

The relevant observations to make are:

- The total energy stored inside the `Heat Storage` component deviates less than 5% between the ESSMOS (compare the bar lengths of the `Heat Storage` to `Heat Line` edge in Figure 5.2).
- The total amount of electrical energy provided by the `Hard Coal PP` component deviates less than 5% between the ESSMOS (compare the bar lengths of the `Hard Coal PP` to `Powerline` edge in Figure 5.2).
- The total amount of electrical energy the `Solar Panel` component provides is about 1.1 times higher for *Calliope* and about 0.9 times lower for *PyPSA* (compare the bar lengths of the `Solar Panel` to `Powerline` edge in Figure 5.2).

Reimer [63] was able to show in his analysis that this observed difference originates from the specific energy supply costs for the `Solar Panel` and the `Hard Coal PP` component. These equal cost parameters were chosen to compare components having equal specific costs but different specific emissions.

- The total amount of chemically bound energy provided to the *Calliope* `Hard Coal Supply Line` and `Hard Coal PP` components is about 0.975 times lower compared to the other ESSMOS (compare the bar lengths of the `Hard Coal Supply Line` edges in Figure 5.2). This difference directly corresponds to the above-observed difference in *Calliope*'s `Hard Coal PP` component.
- The total amount of chemically bound energy provided to the *PyPSA* `Hard Coal Supply` component is about 0.35 times lower than the corresponding amount of *FINE* and *oemof* (compare bar lengths of the `Hard Coal Supply` to `Hard Coal Supply Line` edge in Figure 5.2). *PyPSA* models chemically bound energy transformers with only one outflow fundamentally different from *Tessif*. These singular output *PyPSA* transformers resemble *Tessif*'s `Source` component much closer than *Tessif*'s `Transformer` component in that no actual inflow is constrained (revisit Section 3.2 to put the technical terms of these *Tessif* component template names into context). These *PyPSA* transformer components are called `Generator` components and only declare their input as an arbitrary string. Therefore, the *PyPSA* `Hard Coal PP` component can be loosely interpreted here as a source component, not accounting for any inflows. This modelling approach starkly contrasts the `Hard Coal PP` components of *Tessif*, *Calliope*, *FINE* and *oemof*. Since the *PyPSA* `Hard Coal PP` component requires no inflow, *PyPSA*'s `Hard Coal Supply` component must provide correspondingly less chemically bound energy.

5.4.2.3. Executing PRA-Step 6: Evaluating the CompC PRA-Results for Further Analysis Selection

The two most relevant findings of the above-performed CompC-PRA-steps two and three concerning the question for further analysis are:

1. The total cost and overall OPEX results of the CompC MSC are the same between all ESSMOS (Paragraph 5.4.2.1). Therefore, all relevant high-priority results are equal and warrant no further investigation.
2. The observed differences in the summed-up load results are either irrelevantly minor or originating from different modelling approaches, not significantly impacting the overall system behaviour (Paragraph 5.4.2.2).

The PRA of the CompC MSC thus shows that no further comparative analysis is needed. The results are evaluated as very similar between all compared ESSMOS and warrant no further investigation.

5.4.3. Preliminary Result Analysis of the Component-Expansion (CompE) MSC

The CompE MSC, listed as MSC-number two in table 5.5, is an emissions-constraint expansion problem. The installed capacity of all controllable and volatile renewable energy transformer components, of all storage components and of the power to heat component are subject to optimisation. These components have a specific emissions value allocated to their outflows to

model fuel-dependent emissions (if applicable), as well as their respective life-cycle-assessment-dependent emissions.

The following preliminary result analysis thus executes all PRA-steps. Performing it in this detail requires significant amounts of time. It is, however, the only sensible option here since the observed differences are too large for CRA-steps one to three. This opportunity is thus leveraged to demonstrate how to properly use the preliminary result analysis as a standalone investigation to identify significant differences.

5.4.3.1. Executing PRA-Step 2: Inspecting the CompE High-Priority Results (HPR)

Figure 5.3 shows the CompE HPR of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are shown in Appendix Table A.9. The horizontal axes depict the content-result-categories discussed in Subsection 4.4.2. The vertical axes show the respective result values in relation to *oemof* as coloured bars representing the individual ESSMOS results. The figure is split into two subplots of differently scaled vertical axes to facilitate the visual comparison of the individual high-priority results. The depicted legend applies to both subplots.

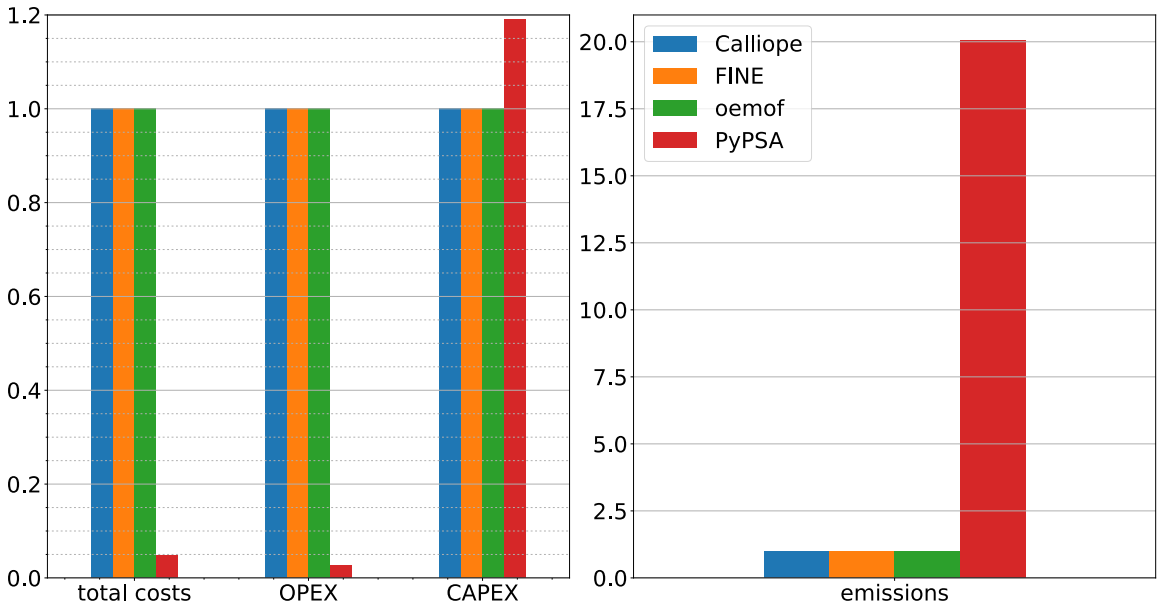

Figure 5.3.: HPR of the Compared ESSMOS Relative to *oemof* for the CompE MSC. The Depicted Numerical Results Are Listed in Appendix Table A.9.

The essential observations are:

- The high-priority results of *Calliope* and *FINE* equal those of *oemof* (compare the blue and orange coloured bar lengths of each high-priority result to the green bar length in Figure 5.3).
- The total cost result of *PyPSA* corresponds to about 0.05 times the amount of the other tools (compare the red bar length of the total cost result to the respective lengths of the differently coloured bars in Figure 5.3).
- *PyPSA*'s OPEX result is about 0.025 times the amount of the other ESSMOS (compare the red bar length of the OPEX result to the respective lengths of the differently coloured bars in Figure 5.3).

- The *PyPSA* CAPEX results correspond to about 1.2 times the amount of the other ESSMOS (compare the red bar length of the CAPEX content-result-category to the respective lengths of the differently coloured bars in Figure 5.3).
- The emission results of *PyPSA* are about twenty times the amount of the other tools (compare the red bar length of the emissions content-result-category to the respective lengths of the differently coloured bars in Figure 5.3).

The overall high-priority results of *PyPSA* thus differ substantially from the other tools, implying significant discrepancies within the ESSM formulation of *PyPSA*.

5.4.3.2. Executing PRA-Step 3: Inspecting the CompE Summed-Up Loads Results

Figure 5.4 shows the CompE summed-up loads results of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are shown in Appendix Table A.10. The vertical axis depicts the edges of the respective flows as (`source`, `target`). Only those edges are shown where the summed-up loads are greater than zero. The horizontal axis shows the numerical result values in MWh as coloured bars representing the individual ESSMOS results.

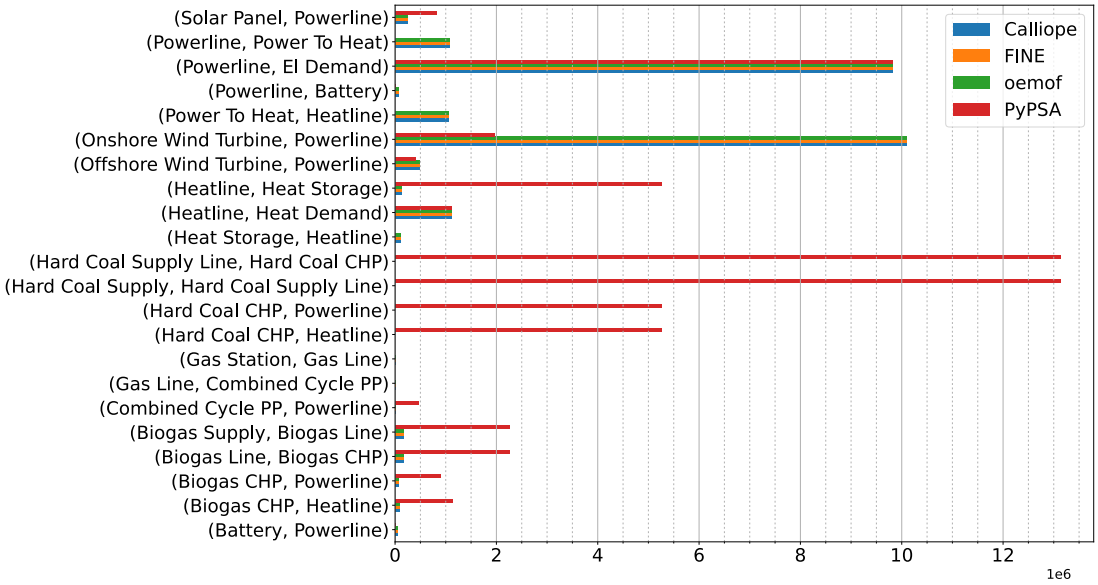

Figure 5.4.: Summed-Up Loads Results in MWh of the Compared ESSMOS for the CompE MSC. Depicted Numerical Results Are Listed in Appendix Table A.10.

The most relevant aspects to note here are:

- The summed-up load results of *Calliope* and *FINE* are the same as those of *oemof* (see all blue, orange, and green bars in Figure 5.4).
- For *PyPSA*, all summed-up load results differ from *oemof* except for the predefined power and heat demand flows (compare all red and green bars in Figure 5.4).
- For *PyPSA*, the amount of energy flowing inside the `Heat Storage` Component is roughly 40 times higher than the other ESSMOS (compare the red bar length of the `Heatline` to `Heat Storage` edge to the length of the other bar length of the same edge in Figure 5.4). The total energy flowing into the *PyPSA* `Heat Storage` amounts to five times the overall

heat demand (compare the red bar length of the `Heatline` to `Heat Storage` edge with the `Heatline` to `Heat Demand` edge in Figure 5.4).

- Only *PyPSA* utilises the `Hard Coal CHP` component and its subsequent commodity supply line (only red bars of edges containing `Hard Coal` in their specifier can be seen in Figure 5.4). The *PyPSA* `Hard Coal CHP` component provides about half of the electrical energy required by the `El Demand` component (compare the red bar length of the `Hard Coal CHP` to `Powerline` edge to the `Powerline` to `El Demand` edge in Figure 5.4).
- *PyPSA* utilises the `Biogas CHP` component and its subsequent commodity supply lines more than ten times as much as the other tools (compare the red bar length of edges containing `Biogas` in their specifier to the length of the other bars in Figure 5.4).
- *PyPSA* only utilises about a third of the amount of the volatile renewable energies (`Solar Panel`, `Offshore Wind Turbine` and `Onshore Wind Turbine` components) compared to the other tools (compare the red bar lengths of the respective `Powerline` edges to the differently coloured bars of the same edges in Figure 5.4).
- *PyPSA* does not use the `Power To Heat` component compared to the other ESSMOS, which provide about 80% of the heat demand using the `Power To Heat` component (compare the bar lengths of the `Power To Heat` to `Heatline` edge to the bar lengths of the `Heatline` to `Heat Demand` edge in Figure 5.4).

5.4.3.3. Executing PRA-Step 4: Inspecting the CompE Installed Capacity Results

Figure 5.5 shows the CompE installed capacity results of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are shown in Appendix Table A.11. The vertical axis depicts the component names. Only those components are shown where the installed capacity is greater than zero and not infinite. The horizontal axis depicts the numerical result values in GWh for the `Battery` and `Heat Storage` component and in GW for the remaining components as coloured bars representing the individual ESSMOS results. The axis is divided into two differently scaled parts to improve the visibility of the lower-value results. The relevant aspects to observe are:

- The installed capacity results of *Calliope* and *FINE* are the same as those of *oemof* (see all blue, orange and green bars in Figure 5.5).
- The installed capacity of the *PyPSA* `Heat Storage` component is about ten times higher than the other ESSMOS (compare the red bar length value (crossing the broken horizontal axis) of the `Heat Storage` component to the length values of its differently coloured bars in Figure 5.5).
- The *PyPSA* `Hard Coal CHP` component has twice the installed capacity compared to the other ESSMOS (compare the red bar length of the `Hard Coal CHP` component to the length of its differently coloured bars in Figure 5.5).
- The *PyPSA* `Biogas CHP` component has only about half the installed capacity compared to the other ESSMOS (compare the red bar lengths of the `Biogas CHP` component to the length of its differently coloured bars in Figure 5.5). At the same time, however, the other tools only utilise less than a tenth of the `Biogas CHP` heat and power output (see PRA-Step three in Paragraph 5.4.3.2). This observation implies *PyPSA* is reducing the unused amount of installed capacity of the `Biogas CHP` component and is the only tool able to do so.

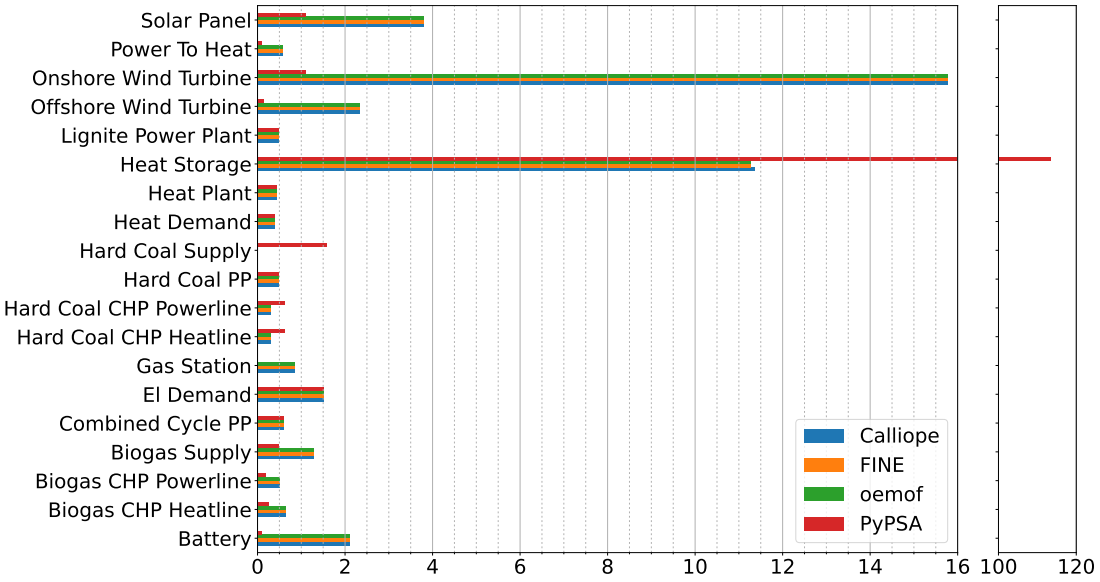

Figure 5.5.: Installed Capacity Results in GW or GWh of the Compared ESSMOS for the CompE MSC. A Broken X-Axis is Utilised to Improve the Visibility of Low-Value Results. The Visualised Numerical Results Are Listed in Appendix Table A.11.

- The *PyPSA* installed capacity of volatile renewable energies in form of the `Solar Panel`, `On-` and `Offshore Wind Turbine` components is about ten times less than the other tools (compare the red bar lengths of the `Solar Panel`, `Offshore Wind Turbine` and `Onshore Wind Turbine` components to the lengths of the differently coloured bars of these components in Figure 5.5).

5.4.3.4. Executing PRA-Step 5: Inspecting the CompE Total Emissions Caused Results

Figure 5.6 shows the CompE total emissions caused results of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are shown in Appendix Table A.12. The vertical axis depicts the edges of the respective flows as (`source`, `target`). Only edges are shown where the emissions caused results are greater than zero. The horizontal axis shows the numerical result values in $\mathrm{Mt_{CO2\text{-}eq}}$ as coloured bars representing the individual ESSMOS results. The axis is divided into two differently scaled parts to improve the visibility of the lower-value results.

With a maximum global emission constraint of $0.25\,\mathrm{Mt_{CO2\text{-}eq}}$, the relevant aspects to note here are:

- The *PyPSA* `Hard Coal CHP` component alone emits more than $4.5\,\mathrm{Mt_{CO2\text{-}eq}}$, thus exceeding the maximum allowed emissions by a factor of 20 (refer to the red bar length values of the `Hard Coal CHP` to `Powerline` edge (crossing the broken horizontal axis) and the `Hard Coal CHP` to `Heatline` edges in Figure 5.6).
- The *PyPSA* `Biogas CHP` component alone emits about $0.25\,\mathrm{Mt_{CO2\text{-}eq}}$, thus corresponding to the maximum allowed emissions. (refer to the red bar length of the `Biogas CHP` to `Powerline` and `Biogas CHP` to `Heatline` edges in Figure 5.6).
- For all other ESSMOS, only the volatile renewable energy components (`Solar Panel`, `Offshore Wind Turbine` and `Onshore Wind Turbine`) and the `Biogas CHP` component are causing emissions, respecting the maximum emission limit of $0.25\,\mathrm{Mt_{CO2\text{-}eq}}$ (refer to

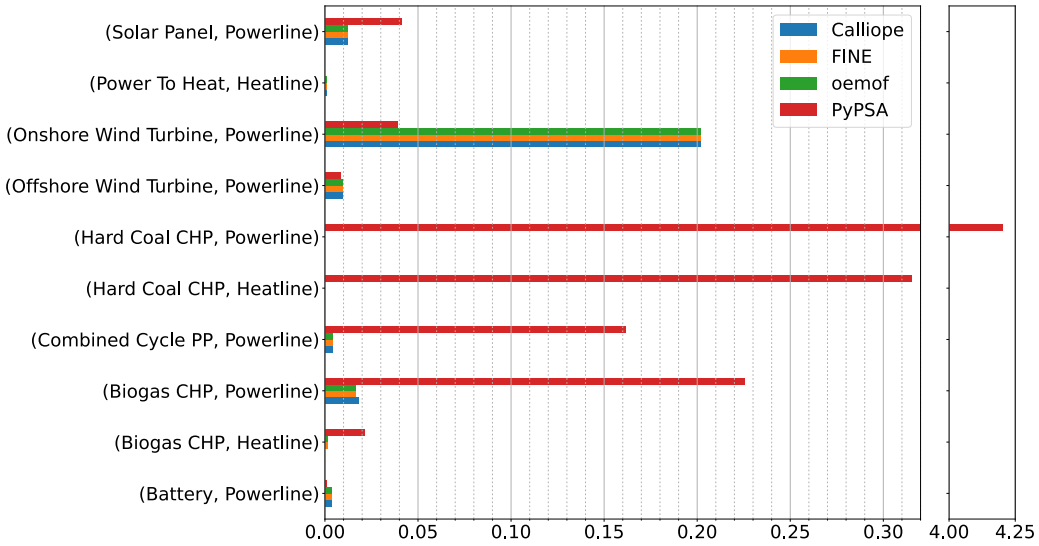

Figure 5.6.: Total Emissions Caused Results in $Mt_{CO2\text{-}eq}$ of the Compared ESSMOS for the CompE MSC. A Broken X-Axis is Utilised to Improve the Visibility of Low-Value Results. The Visualised Numerical Results Are Listed in Appendix Table A.12.

the blue, orange and green bar lengths of the `Solar Panel`, `Offshore Wind Turbine`, `Onshore Wind Turbine` and `Biogas CHP` component in Figure 5.6).

- *Calliope* allocates all `Biogas CHP` emissions to the `Powerline` outflow and none to the `Heatline` outflow in contrast to the other ESSMOS (compare the blue-coloured bar lengths to the differently coloured bars of the `Biogas CHP` edges in Figure 5.6).

5.4.3.5. Executing PRA-Step 6: Evaluating the CompE PRA-Results for Further Analysis Selection

Summarising the results of CompE-PRA-Steps two to five highlights three core differences relevant to the question of further analysis:

1. *PyPSA* does not respect the emission constraint of $0.25\,Mt_{CO2\text{-}eq}$, and it solves the optimisation problem at much lower total costs (Paragraph 5.4.3.1).
2. For *PyPSA*, the total energy flowing into the `Heat Storage` component is about five times the overall heat demand (Paragraph 5.4.3.2), while the installed capacity of *PyPSA*'s `Heat Storage` component is approximately ten times higher the other ESSMOS (Paragraph 5.4.3.3). Both observations are indicative of large amounts of unused thermal energy .
3. The *PyPSA* components causing the most emissions are the CHP components and their commodity supply lines (Paragraph 5.4.3.4). At the same time, both CHP components combined provide about 60% of the required electrical energy (see Paragraph 5.4.3.4).

These three aspects combined indicate that *PyPSA* does not interpret CHP component emission allocation as designed by *Tessif*. This likely results in the preferred use of the overall more energy-efficient cogeneration, in combination with the `Heat Storage` component having non-cyclic state of charge constraints and comparatively low expansion costs.

The PRA of the CompE MSC thus identifies three significantly different components in both the CHP components and the `Heat Storage` component. It furthermore identifies a likely root

cause in the unexpected or different emission allocation behaviour of *PyPSA* CHP components. The comparative result analysis of the CompE MSC can therefore skip CRA-steps one to three and start with identifying potential root causes (CRA-step four) focusing on the CHP-emissions relationship.

5.4.4. Preliminary Result Analysis of the Lossless Commitment (LossLC) MSC

The LossLC MSC is a non-emissions-constraint commitment problem. Therefore, the installed capacities are known before the optimisation, and the emission results do not impact the optimal solution. Hence PRA-steps four and five are skipped in this analysis.

5.4.4.1. Executing PRA-Step 2: Inspecting the LossLC High-Priority Results (HPR)

Figure 5.7 shows the LossLC HPR of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are listed in Appendix Table A.18. The horizontal axis depicts the high-priority results. The vertical axis shows the respective numerical result values in relation to *oemof* as coloured bars representing the individual ESSMOS results. The essential observations to make here are:

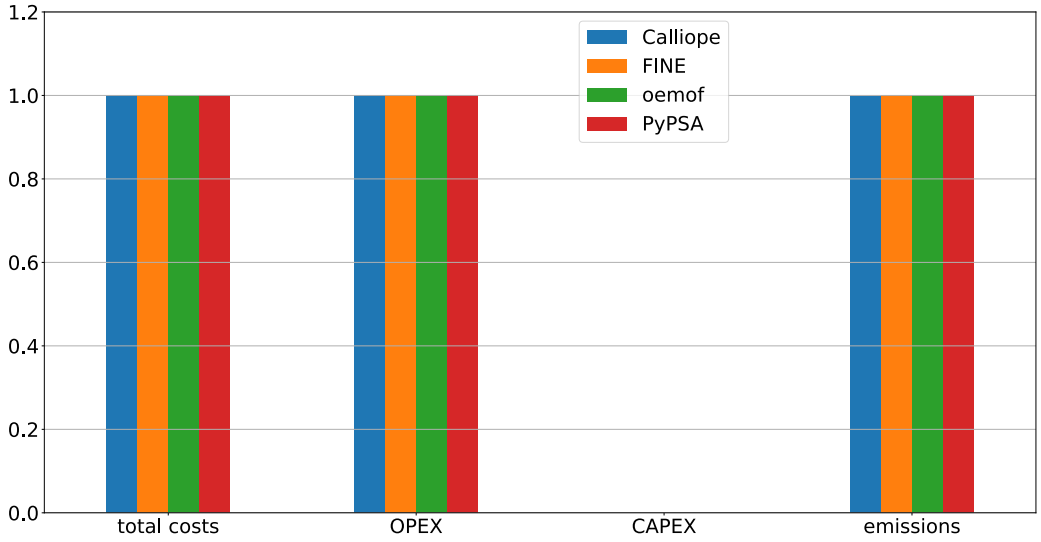

Figure 5.7.: HPR of the Compared ESSMOS Relative to *oemof* for the LossLC MSC. The Depicted Numerical Results Are Listed in Appendix Table A.18.

- For all ESSMOS, no CAPEX results are calculated since the LossLC-MSC is of the commitment problem formulation archetype (no bars are drawn for the CAPEX content-result-category in Figure 5.7).
- The remaining high-priority results are identical or deviate by less than 1% (compare all coloured bar lengths in Figure 5.7).

The LossLC high-priority results are, thus, to be evaluated as highly similar. At this point, the analyst usually terminates the investigation since the observed results are deemed equal enough. However, to showcase that equal or very close to equal global results do not necessarily imply an identical system behaviour, LossLC-PRA-step three is performed in addition.

5.4.4.2. Executing PRA-Step 3: Inspecting the LossLC Summed-Up Loads Results

Figure 5.8 shows the LossLC summed-up loads results of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are shown in Appendix Table A.19. The vertical axis depicts the edges of the respective flows as (`source`, `target`). The horizontal axis shows the numerical result values relative to *oemof* as coloured bars representing the individual ESSMOS results. Figure 5.8 only depicts edges with differing results.

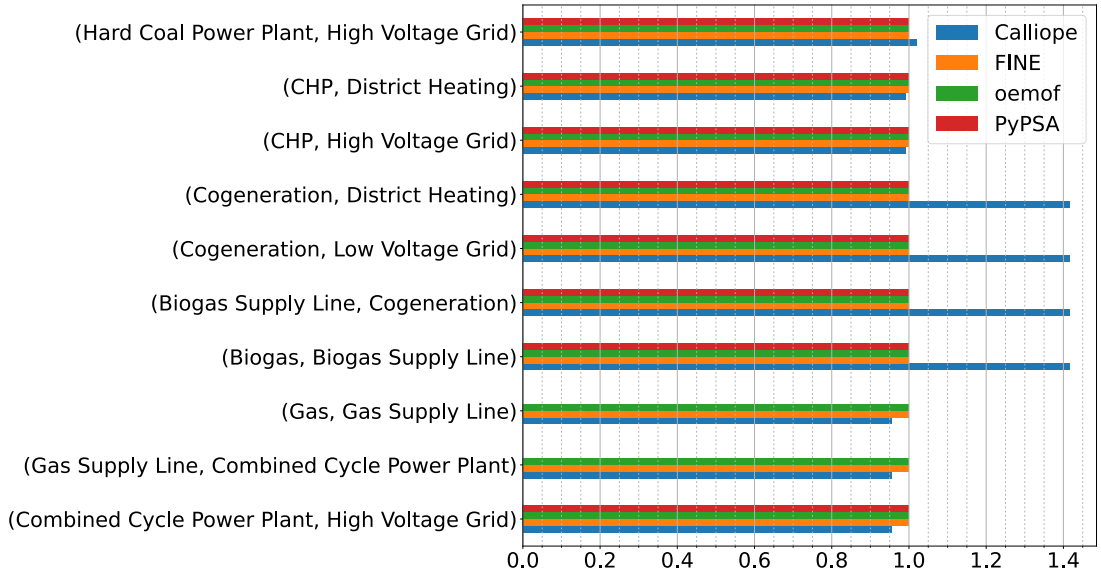

Figure 5.8.: Summed-Up Loads Results of the Compared ESSMOS Relative to *oemof* for the LossLC MSC. Only the Differing Results are Visualised.
The Depicted Numerical Results are Listed in Appendix Table A.19.

The relevant observations to make are:

- For *PyPSA*, the amount of chemically bound energy flowing into the `Combined Cycle Power Plant` component equals zero (no red bars are drawn for the (`Gas`, `Gas Supply Line`) and the (`Gas Supply Line`, `Combined Cycle Power Plant`) edges in Figure 5.8). This PyPSA-specific difference is caused by *PyPSA*'s approach of modelling chemically bound energy transformers as components with no inflow (see also the final observation point stated in Paragraph 5.4.2.2).
- For *Calliope*, the `CHP` component provides about 0.99 times the amount of power and heat compared to the other ESSMOS (compare the blue bar lengths of the `CHP` edges to the length of their differently coloured bars in Figure 5.8).
- The *Calliope* `Combined Cycle Power Plant` component provides about 0.95 times the amount of electrical energy compared to the other ESSMOS (compare the blue bar lengths of the `Gas Supply Line` and `Combined Cycle Power Plant` edges to the length of their differently coloured bars in Figure 5.8).
- For *Calliope*, the `Cogeneration` component provides about 1.4 times the amount of power and heat compared to the other ESSMOS (compare the blue bar lengths of the `Biogas Supply Line` and the `Cogeneration` edges to the length of their differently coloured bars in Figure 5.8).

Figure 5.9 displays the numerical difference between the summed-up load results of the `Hard`

`Coal Power Plant`, `CHP`, `Cogeneration` and `Combined Cycle Power Plant` components compared to *oemof*. The underlying numerical results are shown in Appendix Table A.20. This figure helps to contextualise the observed deviations in *Calliope*'s dispatch of these components. The vertical axis depicts the outflow edges of the before-mentioned components as (`source`, `target`). The horizontal axis shows the summed-up load differences in MWh compared to *oemof* as coloured bars representing the individual ESSMOS results.

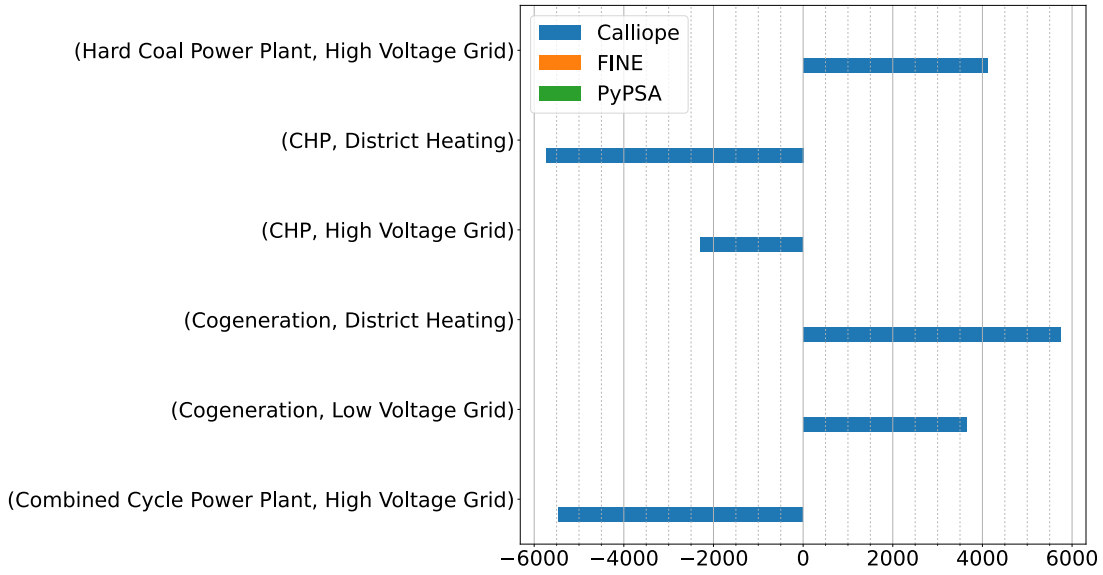

Figure 5.9.: Summed-Up Load Result Differences to *oemof* in MWh of the Compared ESSMOS for Selected Components of the LossLC MSC. The Depicted Numerical Results Are Listed in Appendix Table A.20.

The relevant aspects to observe are:

- Of the edges depicted, only those of *Calliope* differ (no orange or red bars are present in Figure 5.9; *FINE* and *PyPSA* legend entries are shown to indicate their inclusion in the comparison).
- The heat supplied by *Calliope*'s `CHP` component is less than *oemof*'s and this difference is provided by *Calliope*'s `Cogeneration` component (compare the blue bar lengths of the (`CHP`, `District Heating`) and the (`Cogeneration`, `District Heating`) edge).
- The power that the *Calliope* `CHP` and `Combined Cycle Power Plant` components supply less than those of *oemof* is instead provided by *Calliope*'s `Hard Coal Power Plant` and `Cogeneration` components (compare the blue bar lengths of the combined (`CHP`, `High Voltage Grid`) and (`Combined Cycle Power Plant`, `High Voltage Grid`) edges to the combined length of the (`Hard Coal Power Plant`, `High Voltage Grid`) and (`Cogeneration`, `District Heating`) edges in Figure 5.9).

5.4.4.3. Executing PRA-Step 6: Evaluating the LossLC PRA-Results for Further Analysis Selection

The most relevant findings of the above-performed LossLC-PRA-steps two and three concerning the decision for further analysis are:

1. The high-priority results of the LossLC MSC are the same between all ESSMOS or deviate by less than 1% and warrant no further investigation (Paragraph 5.4.4.1).
2. The observed deviations in the summed-up load results originate from differently dispatched components resulting in the same overall energy amount supplied (Paragraph 5.4.4.2). However, these deviations do not impact the relevant global results. Since the total emission results are the only global result differing slightly (see Appendix Table A.18), the observed dispatch differences likely originate from minor deviations in how the ESSMOS allocate specific emissions.

Further analyses would be recommended if the LossLC MSC were a standalone investigation focused on the actual optimisation results rather than their comparison. Since this is not the case and the overall global results are close to equal, no additional effort is undertaken.

The PRA of the LossLC MSC thus shows that no further comparative analysis is needed. The results are evaluated as very similar between all compared ESSMOS and warrant no further investigation.

5.4.5. Preliminary Result Analysis of the No-Congestion Transformer-Commitment (TransC) MSC

Figure 5.10 shows the No-Congestion TransC HPR of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are listed in Appendix Table A.21. The horizontal axis depicts the high-priority results. The vertical axis shows the respective numerical result values in relation to *oemof* as coloured bars representing the individual ESSMOS results.

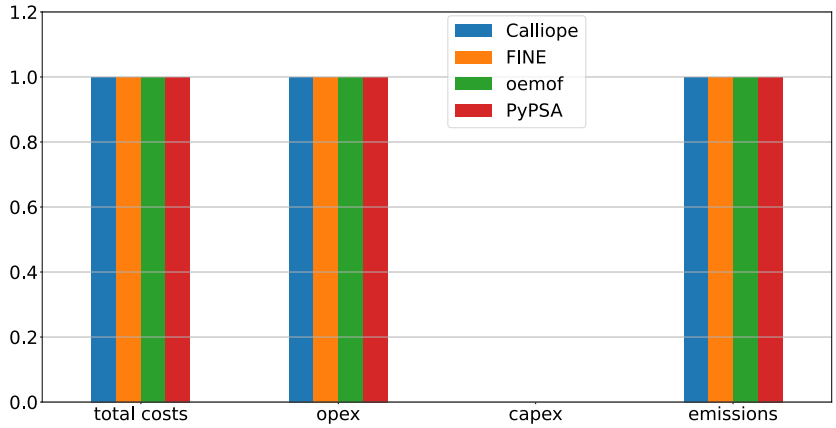

Figure 5.10.: HPR of the Compared ESSMOS Relative to *oemof* for the No-Congestion TransC MSC.// The Depicted Numerical Results Are Listed in Appendix Table A.21.

The relevant aspects to note are:

- For all ESSMOS, no CAPEX results are calculated since the CompC-MSC is of the commitment problem formulation archetype (there are no bars drawn for the CAPEX result in Figure 5.10).
- The total cost, OPEX and emission results are identical for all ESSMOS (compare all coloured bar lengths of the respective results in Figure 5.10).

Since no HPR deviations are observed, the investigation is stopped here. No additional valuable findings are expected to be uncovered by further comparing this MSC.

The PRA of the No-Congestion TransC MSC thus concludes that no further comparative analysis is needed.

5.4.6. Preliminary Result Analysis of the Congestion Transformer-Commitment (TransC) MSC

The Congestion TransC MSC is also a non-emissions-constraint commitment problem. Therefore, the installed capacities are known before the optimisation, and the emission results do not impact the optimal solution. Hence PRA-steps four and five are skipped in the following analysis.

5.4.6.1. Executing PRA-Step 2: Inspecting the Congestion TransC High-Priority Results (HPR)

Figure 5.11 shows the Congestion TransC HPR of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are listed in Appendix Table A.22. The horizontal axis depicts the high-priority results. The vertical axis shows the respective numerical result values in relation to *oemof* as coloured bars representing the individual ESSMOS results.

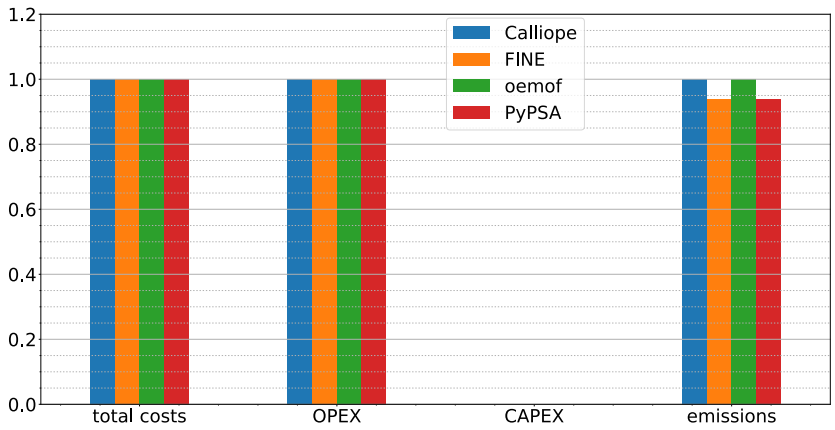

Figure 5.11.: HPR of the Compared ESSMOS Relative to *oemof* for the Congestion TransC MSC.
The Depicted Numerical Results Are Listed in Appendix Table A.22.

The relevant aspects to note are:

- For all ESSMOS, no CAPEX results are calculated since the CompC-MSC is of the commitment problem formulation archetype (no bars are drawn for the CAPEX content-result-category in Figure 5.11).
- The total cost and OPEX results are identical for all ESSMOS (compare all coloured bar lengths of the respective content-result-categories in Figure 5.11).
- The *FINE* and *PyPSA* emission results are about 0.95 times the amount of the other ESSMOS (compare the orange and red bar lengths of the emission content-result-category to the blue and green bar lengths of the same category in Figure 5.11).

5.4.6.2. Executing PRA-Step 3: Inspecting the Congestion TransC Summed-Up Loads Results

Figure 5.12 shows the Congestion TransC summed-up loads results of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are shown in Appendix Table A.23. The vertical axis depicts the edges of the respective flows. The horizontal axis shows the numerical result values in MWh as coloured bars representing the individual ESSMOS results.

The only relevant observations to make here is that all summed-up load results of all compared ESSMOS are equal (compare the differently coloured bar lengths of each edge in Figure 5.11).

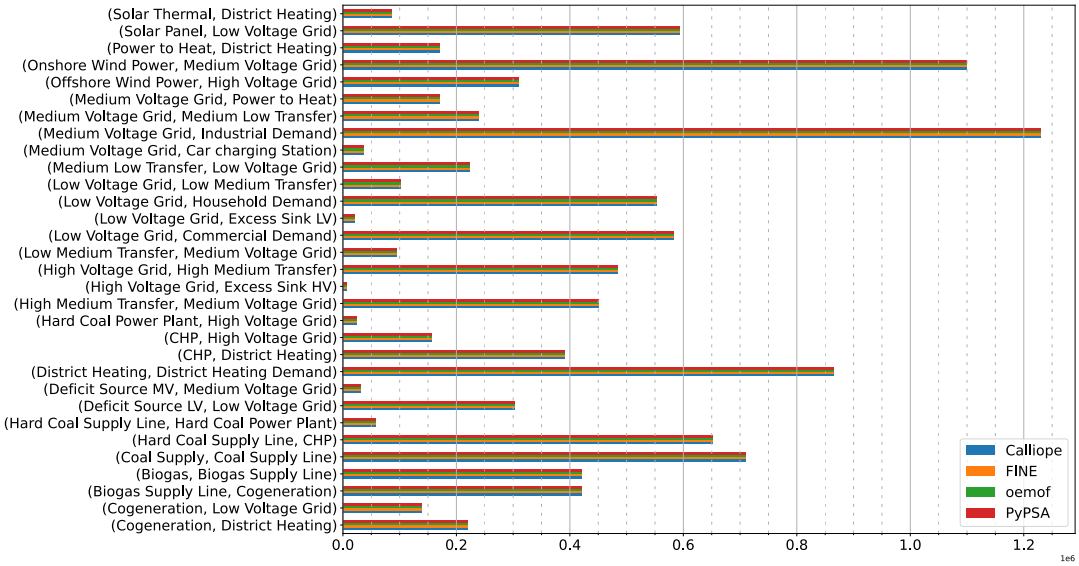

Figure 5.12.: Summed-Up Loads Results of the Compared ESSMOS in MWh for the Congestion TransC MSC. The Depicted Numerical Results Are Listed in Appendix Table A.23.

5.4.6.3. Executing PRA-Step 6: Evaluating the Congestion TransC PRA-Results for Further Analysis Selection

The decisive points of the above-performed Congestion-TransC-PRA-steps two and three concerning further analysis are:

1. The total cost and OPEX results of the LossLC MSC are the same for all ESSMOS. Thus, all relevant high-priority results are equal (Paragraph 5.4.6.1).
2. All summed-up load results for all ESSMOS are equal (Paragraph 5.4.6.2).
3. The *FINE* and *PyPSA* emission results are about 0.95 times the amount of the other ESSMOS (Paragraph 5.4.6.1). In the context of points two and three this difference is considered to be irrelevantly small.

The PRA of the Congestion TransC MSC thus indicates that no further comparative analysis is needed. The inspected relevant results for each ESSMOS are equal and warrant no further investigation.

5.4.6.4. Additional Remarks Concerning Observable Redispatch Events

The Congestion TransC MSC is designed to model grid congestion effects and the subsequent redispatch necessary to counteract these. This redispatch can be calculated by comparing the inflows of an `Excess Sink` component and the outflows of the respective `Deficit Source` component on the other side of the grid component having congestion issues (revisit Figure 3.6 in Paragraph 3.8.2.2 to put the component names mentioned here into visual context).

Within the optimised time span, two occurrences of redispatch happen. Both are listed in Table 5.6. This table numbers the occurrence in its left column, states the occurences' point in time in its middle column and quantifies the amount of redispatched power in MW in its right column. Both redispatch events happen at the `High Medium Transfer` component and thus in the power flow direction from the high voltage grid to the medium voltage grid.

Table 5.6.: Congestion and Redispatch Occurrences in the Congestion-TransC MSC.

Occurrence	Timestamp	Redispatch [MW]
1	2030-10-13 17:00:00	2096
2	2030-10-13 18:00:00	5368

This redispatch phenomenon is irrelevant to the comparative method developed in Chapter 4. However, it indicates that the Congestion-TransE MSC's approach is suited to model grid congestion-related issues and thus achieves the modelling objectives formulated in Paragraph 3.8.2.1.

5.4.7. Preliminary Result Analysis of the Transformer Expansion (TransE) MSC

The TransE MSC is a non-emissions-constraint expansion problem. Its transfer components have an expandable installed capacity which are subject to optimisation. Thus, all PRA-steps are performed for this MSC.

5.4.7.1. Executing PRA-Step 2: Inspecting the TransE High-Priority Results (HPR)

Figure 5.13 shows the Transformer-Expansion HPR of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are listed in Appendix Table A.24. The horizontal axis depicts the high-priority results. The vertical axis shows the respective numerical result values in relation to *oemof* as coloured bars representing the individual ESSMOS results.

The relevant aspects to note are:

- The total cost, OPEX and emission results are identical for all ESSMOS (compare all coloured bar lengths of the respective content-result-categories in Figure 5.13).
- The *FINE* CAPEX results are about 1.05 times the amount of the other ESSMOS (compare the orange bar lengths of the CAPEX content-result-category to the differently coloured bar lengths of the same category in Figure 5.13).

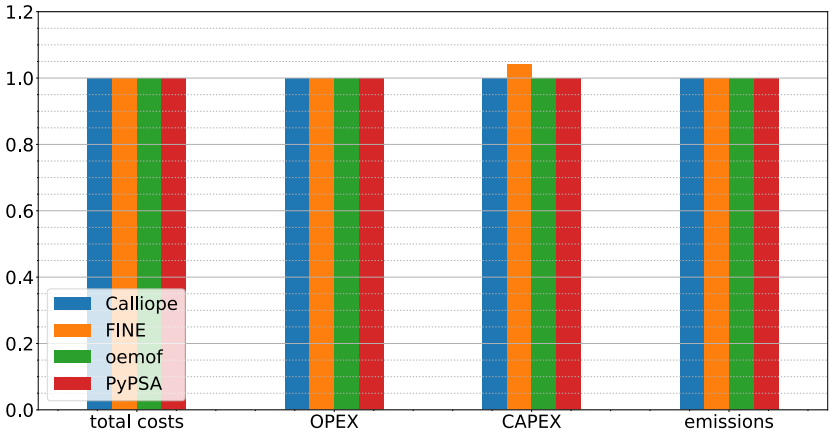

Figure 5.13.: HPR of the Compared ESSMOS Relative to *oemof* for the TransE MSC. The Depicted Numerical Results Are Listed in Appendix Table A.24.

5.4.7.2. Executing PRA-Step 3: Inspecting the TransE Summed-Up Loads Results

Figure 5.14 shows the TransE summed-up loads results of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are shown in Appendix Table A.27. The vertical axis depicts the edges of the respective flows as (`source`, `target`). The horizontal axis shows the numerical result values in MWh as coloured bars representing the individual ESSMOS results.

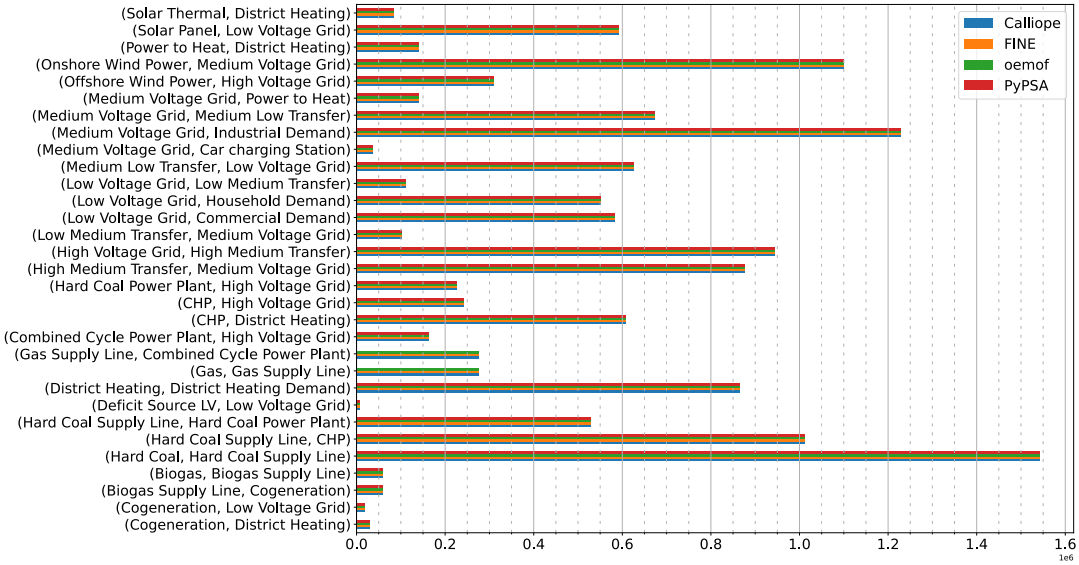

Figure 5.14.: Summed-Up Loads Results of the Compared ESSMOS in MWh for the TransE MSC. The Depicted Numerical Results Are Listed in Appendix Table A.27.

The only relevant observations to make here is that all summed-up load results of all compared ESSMOS are equal (compare the differently coloured bar lengths of each edge in Figure 5.14).

5.4.7.3. Executing PRA-Step 4: Inspecting the TransE Installed Capacity Results

Figure 5.15 shows the TransE installed capacity results of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are shown in Appendix Table A.25. The vertical axis depicts

the component names. Only the transfer grid components are shown since these are the only expandable components. The horizontal axis depicts the numerical result values in MW as coloured bars representing the individual ESSMOS results.

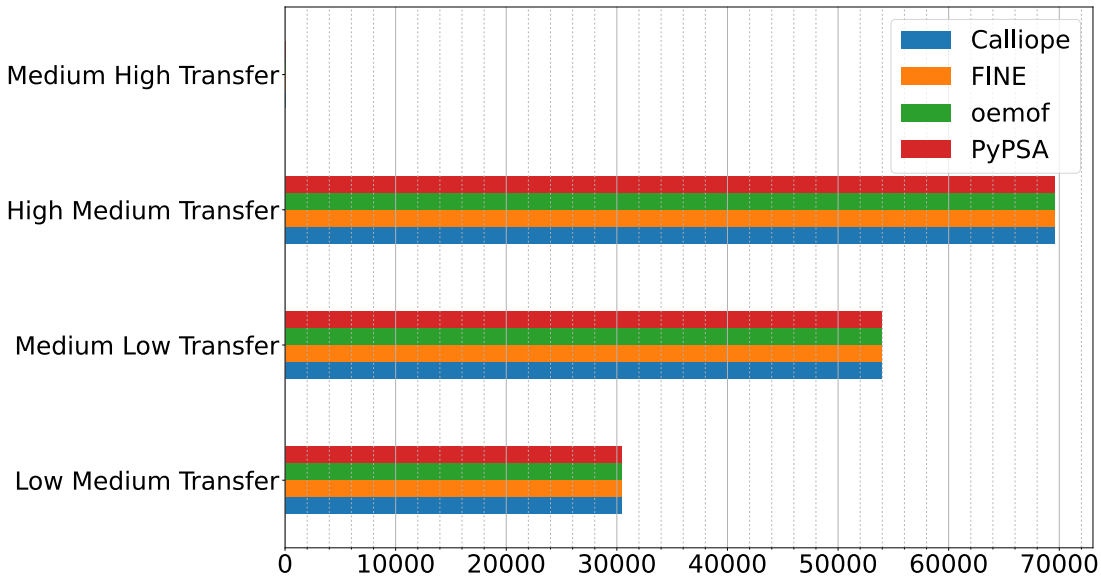

Figure 5.15.: Installed Capacity Results of the Compared ESSMOS in MW for the Expandable Components of the TransE MSC. The Visualised Numerical Results Are Listed in Appendix Table A.25.

The most noteworthy aspects shown are:

- None of the compared ESSMOS expands the `Medium High Transfer` component (there are no coloured bars for the `Medium High Transfer` component in Figure 5.15).
- The installed capacity of all expandable components is the same for all compared ESSMOS (compare the lengths of the differently coloured bars for each component in Figure 5.15). The expansion problem was thus solved equally among the tools.

5.4.7.4. Executing PRA-Step 5: Inspecting the TransE Total Emissions Caused Results

Figure 5.16 shows the TransE total emission caused results of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are shown in Appendix Table A.26. The vertical axis depicts the edges of the respective flows. Only edges are shown where the emissions caused results are greater than zero. The horizontal axis shows the numerical result values in $Mt_{CO2\text{-}eq}$ as coloured bars representing the individual ESSMOS results. The most relevant aspects to observe here are:

- *Calliope* only allocates emission to one outflow of multiple outflow components (compare the (`CHP`, `High Voltage Grid`) edge to the (`CHP`, `District Heating`) edge and the (`CHP`, `High Voltage Grid`) edge to the (`CHP`, `District Heating`) in Figure 5.16).
- For *Calliope*, the combined `CHP` component emissions equal the combined `CHP` component emission of the other ESSMOS (compare combined bar lengths of the non-blue bars of the `CHP` component to the blue bar length of (`CHP`, `District Heating`) edge in Figure 5.16).
- The same behaviour can be observed for the `Cogeneration` component.

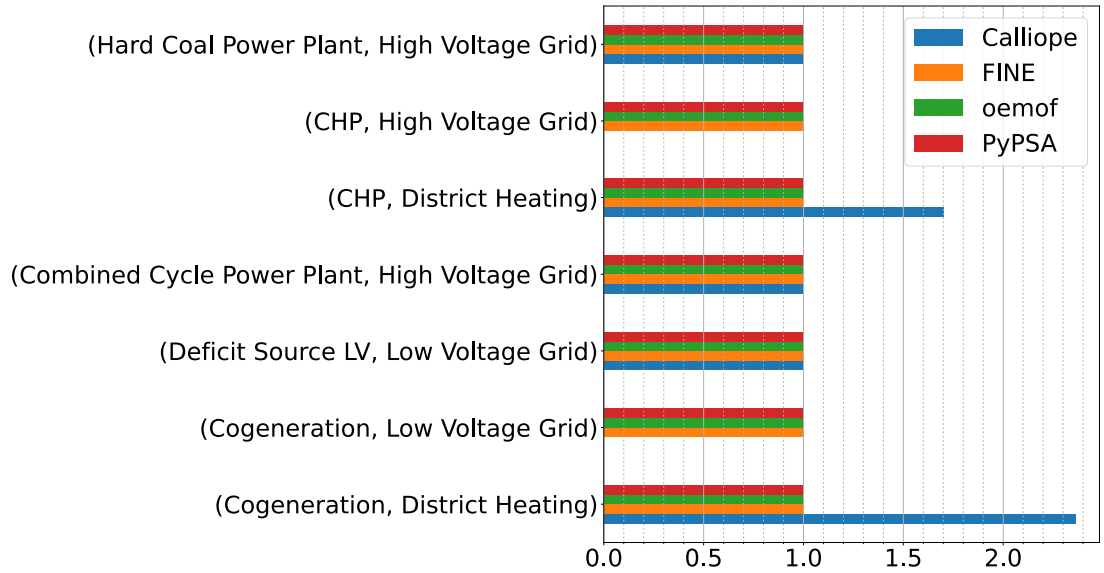

Figure 5.16.: Total Emissions Caused Results of the Compared ESSMOS Relative to *oemof* for the TransE MSC. The Visualised Numerical Results Are Listed in Appendix Table A.26.

- When taking the above-mentioned *Calliope* particularities into account, the emissions caused results for all ESSMOS are equal.

5.4.7.5. Executing PRA-Step 6: Evaluating the TransE PRA-Results for Further Analysis Selection

The key findings from the TransE-PRA steps two and three concerning further analysis include:

- The total cost, OPEX and emission results are identical across all ESSMOS, while *FINE*'s CAPEX results are approximately 1.05 times the amount of the other ESSMOS (Paragraph 5.4.7.1).
- The summed-up load results for all components are consistent between the compared ESSMOS (Paragraph 5.4.7.2).
- The emissions caused results for all ESSMOS are equal (Paragraph 5.4.7.4).
- The installed capacity for all components is equal between the compared ESSMOS (Paragraph 5.4.7.3). Consequently, *FINE*'s higher CAPEX results do not impact the overall system behaviour and likely stem from minor differences in internal CAPEX representation or the *Tessif* to *FINE* interface.

The PRA of the TransE MSC thus indicates that no further comparative analysis is needed. The inspected relevant results for each ESSMOS are equal and warrant no further investigation.

5.5. Executing CM-Step 4: Comparative Result Analysis (CRA) of Selected Optimisation Results

Performing the comparative result analysis (CRA), as CM-step 4, includes five CRA-steps. The PRA in Section 5.4 marks the CompE MSC to be inspected in greater detail during this CRA. However, it also particularly recommends skipping CRA-steps 1 to 3 for the CompE MSC since it isolates the CHP component emission relationship as the remaining investigation subject. Subsection 5.5.1 therefore starts with CRA-step 4 and continues analysing the CHP component by identifying underlying root causes for the observed result differences.

During the CRA of the CompE MSC several key findings are obtained leading to an additional, slightly altered MSC called "Modifed Component Expansion (Modified CompE)". In Subsections 5.5.2 to 5.5.5 all five CRA-steps are fully executed for this Modified CompE.

5.5.1. Executing CRA-Step 4 on the CompE MSC: Analysis for Identifying Root Causes

The CompE MSC identification of root causes (IRC) employs IRC-steps 1 to 5, as outlined in Paragraphs 4.5.4.1 to 4.5.4.5 and executed in Paragraphs 5.5.1.1 to 5.5.1.5.

5.5.1.1. Executing IRC-Step 1: The CompE Investigative Scenario Formulation

Following the selection for further analysis of ISD-step there (Paragraph 5.4.3.5), an investigative scenario formulation is derived focusing on this central question:

Do the *PyPSA* CHP components respect the allocated emissions as intended by *Tessif*, and how does it compare to the other ESSMOS?

Three **CHP-investigations** are conducted to address this question:

1. Perform a tabular parameter comparison of the component templates underlying the CompE MSC `Hard Coal CHP` and `Biogas CHP` components to ensure the presence of an emission parameter and its accurate allocation. These templates are grouped by the term "CHP-related" since exact template names differ between ESSMOS.
2. Confirm the correct allocation of emissions in *Tessif* and *PyPSA* through a parameter check.
3. Perform a plausibility check using a small ESSM focused around a singular CHP component, imposing an emission constraint forcing the solver to use more expansive but less emitting alternatives.

5.5.1.2. Executing IRC-Step 2: The Tabular Parameter Comparison of the CHP-Related Component Template

Table 5.7 displays an excerpt of the CHP-related component template tabular parameter comparison (TPC) between *Tessif* and *PyPSA*. The complete comparison for all compared ESSMOS is available in *Tessif*'s online documentation[1] [61]. The first row shows the CHP-related com-

[1] *Tessif*'s TPC of CHP-related component templates: https://tessif-phd.readthedocs.io/en/latest/usage/supported_models/tabular_parameter_comparison.html#transformer

ponent template name, and the second row indicates the internal interface for addressing the created component. The third row lists the component template's emission allocation parameter, indicating that *PyPSA*'s `Link` component template does not inherently provide this parameter.

Table 5.7.: Tabular Parameter Comparison Excerpt of *Tessif*'s and *PyPSA*'s CHP-related Component Templates. The Full Table is Listed in *Tessif*'s Online Documentation [61].

#		Tessif	PyPSA
1	Component Template	`Transformer`	`Link`
2	Identifier	`Uid`	`name`
3	**Energy Flow Specific Emissions**	`flow emissions`	`-`

The CHP component TPC shows that the *PyPSA* CHP component, in contrast to *Tessif*, does not have an inherent emission allocation parameter (refer to the highlighted row in table 5.7). *PyPSA*'s online documentation states [93]:

"Global constraints are added to OPF problems and apply to many components at once. Currently, only constraints related to primary energy (i.e. before conversion with losses by generators) are supported, the canonical example being CO2 emissions for an optimisation period. Other primary-energy-related gas emissions also fall into this framework." [2]

Tessif circumvents that by adding individual carrier objects to its *PyPSA* networks for each component having CO2 emissions allocated, as stated in *Tessif*'s online documentation [61]:

"Note how an extra Carrier object gets parsed to accommodate for the Link allocated emission constraints." [3]

The tabular parameter comparison thus confirms the capability to allocate emissions to CHP-related component templates, addressing CHP-investigation-1.

5.5.1.3. Executing IRC-Step 3: The CHP Component Parameter Check

During the CompE MSC creation, the `Hard Coal CHP` and `Biogas CHP` components are created by explicitly parameterising *Tessif*'s `Transformer` component template. The parameter check results in Table 5.8 overleaf show the relevant emission allocation parameters after the ESSM is created. These results are obtained by examining the created *Tessif* ESSM using its public interface. The specific emission values are expressed in $t_{CO2\text{-}eq}$/MWh.

The CompE MSC parameter check results show that the `Hard Coal CHP` and `Biogas CHP` components have specific emissions values allocated to their outflows but none to their inflows (refer to the parameter values column in Table 5.8). Comparing these parameter check results with the initial parameterisation stated in Appendix A.1 shows that *Tessif* allocates these correctly.

To optimise *Tessif*'s ESSM using *PyPSA*, it must be transformed into *PyPSA*'s ESSM representation. *Tessif*'s CHP components, created through the `Transformer` component template,

[2] Online documentation of *PyPSA*'s global emission constraint: https://pypsa.readthedocs.io/en/v0.19.3/components.html#global-constraints

[3] Online documentation of *Tessif*'s to *PyPSA* emission handling: https://tessif-phd.readthedocs.io/en/latest/api/transform/es2es/ppsa.html#tessif.transform.es2es.ppsa.create_pypsa_links_from_transformers

Table 5.8.: Emission Allocation Parameter Check of *Tessif*'s `Hard Coal CHP` and `Biogas CHP` Components of the CompE MSC. Results Obtained by Inspecting the Components After Their Creation. Specific Emission Values are Expressed in $t_{CO2\text{-}eq}$/MWh. The Full Parameterisation can be Found in *Tessif*'s Example Library [95].

Component	Parameter	Parameter Values
`Hard Coal CHP`	`flow_emissions`	`{"hard_coal": 0, "electricity": 0.8, "hot_water": 0.06}`
`Biogas CHP`	`flow_emissions`	`{"biogas": 0, "electricity": 0.25, "hot_water": 0.01875}`

are transformed into *PyPSA* CHP components using *PyPSA*'s `Link` component template. Table 5.9 lists the *PyPSA* CHP component parameter check results obtained by examining the component's public interface (first and second column). As *PyPSA* does not allocate emissions directly to the `Link` component (see also Paragraph 5.5.1.2), the corresponding `Carrier` objects are also examined (third and fourth column). The outflow-specific conversion factors are also given since they are used in calculating the inflow-specific carrier-allocated emissions.

Table 5.9.: CompE MSC Emission Allocation Parameter Check of *PyPSA*'s `Hard Coal CHP` and `Biogas CHP` Components. Results Obtained by Inspecting the Components After Their Creation. Emissions are Expressed in $t_{CO2\text{-}eq}$/MWh Inflow.

Component	Conversion Factors	Corresponding `Carrier` Object	Carrier Allocated Emission
`Hard Coal CHP`	`electricity: 0.4`	`Hard Coal CHP.carrier`	0.34
	`hot_water: 0.4`		
`Biogas CHP`	`electricity: 0.4`	`Biogas CHP.carrier`	0.11
	`hot_water: 0.5`		

The *PyPSA* parameter check results in the rightmost column of Table 5.9 indicate that the CompE MSC `Hard Coal CHP` and `Biogs CHP` corresponding `Carrier` objects have inflow specific emission values of 0.34 and 0.11 $t_{CO2\text{-}eq}$/MWh. These values represent the inflow-specific emission values e_{in} obtained by summing up the outflow emission values $e_{out,i}$ multiplied by the respective outflow efficiency $\eta_{out,i}$ ($e_{in} = \sum e_{out,i} \cdot \eta_{out,i}$).

The initial goal of allocating emissions to *PyPSA*'s `Battery` component was, therefore, successfully reached. The parameter check, hence, verifies the correct emission allocation, thus, addressing CHP-investigation-2. Nevertheless, the observed difference in the CompE emission results remains to be explained.

Upon inspecting *PyPSA*'s emission calculation method, it becomes evident that *PyPSA*'s `Link` component templates are not considered when calculating the global emissions. *PyPSA*'s online documentation states [93]:

"*Emissions can come from generators whose energy carriers have CO2 emissions and from stores and storage units whose storage medium releases or absorbs CO2 when it is converted.*" [4]

[4] https://pypsa.readthedocs.io/en/v0.19.3/optimal_power_flow.html#global-constraints

In contrast, *Tessif* enables CHP component emissions to be allocated to inflows or outflows. Thus, *Tessif*'s and *PyPSA*'s approaches to modelling CHP component emissions fundamentally differ.

5.5.1.4. Executing IRC-Step 4: The CHP Component Plausibility Check

To examine the effects of differing CHP-emission modelling approaches between *Tessif* and *PyPSA*, a plausibility check (PLC) is conducted, as outlined in Paragraph 4.5.4.4. A distinct ESSM is created using *Tessif*'s abstract component templates, deliberately designing each parameter, component identifier, interconnection, and overall emission constraint. The primary aim is to compel the solver to utilise the CHP component to the extent permitted by the emission constraint before switching to costlier alternatives, enabling the analysis of the CHP component-emission relationship as outlined in CHP-investigation-3. The system remains abstract and generic by supplying only the essential parameters, allowing for the isolation and comprehension of underlying mechanisms without delving into deeper optimisation problem formulation layers (refer to the ESSMOS-tool abstraction layers discussion in Subsection 2.1.8). The system uses abstract specifier names, as no actual energy system component is modelled within the context of this plausibility check. The underlying ESSM is available in *Tessif*'s example library[5] [40].

The generic system visualisation (GSV) of the ESSM created for this PLC is depicted in Figure 5.17, alongside the carefully and deliberately chosen parameterisation in Table 5.10. CU represents cost unit, EU denotes energy unit, and EMU signifies emission unit. To instigate the switch between using the `CHP Component` and using the `Power/Heat Source Component`, the following core design aspects were selected:

1. The optimisation encompasses four time steps.
2. The optimisation's objective is the minimisation of total costs.
3. The overall emission constraint is set to 54 EMU (3 time steps multiplied by 18 EMU).
4. The `Power Demand Component` requires 10 EU per time step and no specific costs or emissions are allocated to its inflow (refer to the power value column of the `Power/Heat Demand Component` entry in Table 5.10).
5. The same applies to the `Heat Demand Component` but with a requirement of 8 EU per time step instead.
6. The `CHP Component` enables power flow between 0 and 10 EU and heat flow between 0 and 8 EU per time step as determined by the optimisation process. Both outflows do not cause any costs but emission of 1 EMU/EU. (refer to the `CHP Component` entry in Table 5.10). Consequently, the `CHP Component` causes 18 EMU at maximum load (10 EMU due to the generated power and 8 EMU due to the generated heat).
7. The `Power Source Component` allows power flow from 0 to 10 EU per time step, depending on the optimisation results. While no specific emissions are allocated, the provided energy entails a cost of 1 CU/EU (refer to the power value column of the `Power/Heat Source Component` entry in Table 5.17).

[5] Code and access to the CHP-emissions ESSM: https://tessif-examples.readthedocs.io/en/latest/source/examples/plausibility/chp_emissions.html

Component	Parameter		Value Power	Value Heat
Power/ Heat Demand Component	flow_rates	$t=1:$	10EU	8EU
		$t=2:$	10EU	8EU
		$t=3:$	10EU	8EU
		$t=4:$	10EU	8EU
	flow_emissions		0EMU/EU	0EMU/EU
	flow_costs		0CU/EU	0CU/EU
CHP Component	float_rates	$t=1:$	$[0,10]$ EU	$[0,8]$ EU
		$t=2:$	$[0,10]$ EU	$[0,8]$ EU
		$t=3:$	$[0,10]$ EU	$[0,8]$ EU
		$t=4:$	$[0,10]$ EU	$[0,8]$ EU
	installed_capacity		10 EU	8 EU
	outflow_emissions		**1 EMU/EU**	**1 EMU/EU**
	flow_costs		0CU/EU	0CU/EU
Power/ Heat Source Component	flow_rates	$t=1:$	$[0,10]$EU	$[0,8]$EU
		$t=2:$	$[0,10]$EU	$[0,8]$EU
		$t=3:$	$[0,10]$EU	$[0,8]$EU
		$t=4:$	$[0,10]$EU	$[0,8]$EU
	flow_emissions		0EMU/EU	0EMU/EU
	flow_costs		**1 CU/EU**	**1 CU/EU**

Table 5.10.: Parameterisation of the CHP-Emissions PLC. The **Total Emissions are Limited to 54 MU**. The Optimisation Time Span Length is 4 Steps. The Parameterisation can also be Accessed Through *Tessif-Examples'* Online Documentation [95].

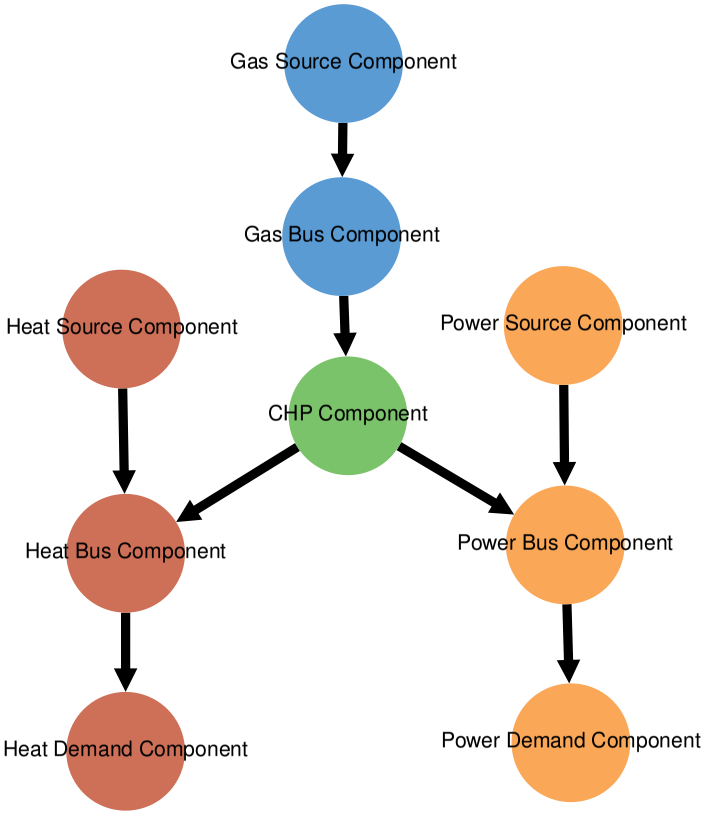

Figure 5.17.: GSV of the CHP-Emissions PLC. Relevant Component Parameters are Listed in Table 5.10 Using the Same Name. All Components and Parameters are Explicitly Designed for this PLC. The MSC is Available in *Tessif-Examples* [40].

8. The same applies to the `Heat Source Component` but instead with a maximum flow rate of 8 EU.

The following optimisation result expectations can be formulated based on the parameterisation above:

1. The `CHP Component` provides power and heat for three of the four time steps since it is the least expensive option.
2. At one of the four time steps, the global emission limit is reached due to the `CHP Component` emitting 18 EU at full load. The costlier yet less emitting `Power Source Component` and `Heat Source Component` meet the power and heat demand at this remaining time step. Consequently, the emission limit is fully exploited, resorting to the more expensive alternatives only when indispensable.
3. Given the above two expectations, the overall total cost results will be 18 CU (caused by the `Power Source Component` and `Heat Source Component`), and the emissions will amount to 54 EU (caused by the `CHP Component`).

Table 5.11 displays the CHP-Emissions plausibility check results. The table contains three main columns. The first column lists markers for identifying the rows' content, including the four optimisation time span steps. Column two represents the identical *Calliope*, *FINE*, and *oemof* results. Column three shows the deviating *PyPSA* results. Each of these two ESSMOS result columns is split into four columns displaying the `CHP Component` and `Power/Heat Source Component` results for their respective `Power` and `Heat` outflows. The first row lists the total caused outflow-specific emission and cost results. The remaining rows show the outflow results of time steps one to four, expressed in EU per time step.

Table 5.11.: Relevant results of the CHP-Emissions plausibility check.

ESSMOS	***Calliope***		***FINE***	***oemof***		***PyPSA***		
Component	`CHP`		`Source`		`CHP`		`Source`	
Power/Heat	`Power`	`Heat`	`Power`	`Heat`	`Power`	`Heat`	`Power`	`Heat`
CU/EMU	30 EMU	24 EMU	10 CU	8 CU	**40 EMU**	**32 EMU**	**10 CU**	**0 CU**
t=1	10	8	0	0	10	8	0	0
t=2	10	8	0	0	10	8	0	0
t=3	10	8	0	0	10	8	0	0
t=4	**0**	**0**	**10**	**8**	10	8	0	0

All examined ESSMOS, except for *PyPSA*, produce the expected results. *Calliope*, *FINE* and *oemof* use 30/24 EU from the `CHP Component` and 10/8 EU from the `Power/Heat Source Component` (refer to the t=... rows for *Calliope*, *FINE* and *oemof* in Table 5.11). Conversely, *PyPSA* utilises the `CHP Component` during all four time steps to meet the demand without using the `Power/Heat Source Component`. As a result, *PyPSA* solves the optimisation problem at lower costs but with higher emissions (refer to the *PyPSA* CU/EMU row in Table 5.11).

5.5.1.5. Executing IRC-Step 5: Conducting a Final Assessment of the CompE MSC IRC

The tabular parameter comparison in Paragraph 5.5.1.2 demonstrated that *Tessif* and *PyPSA* provide interfaces for allocating emissions to CHP-related component templates.

The parameter check in Paragraph 5.5.1.3 confirmed that a *Tessif* `CHP` component, created through a `Transformer` component template, could be transformed into a *PyPSA* `CHP` component using the `Link` component template while preserving the emission allocation parameter. However, the parameter check also uncovered that *PyPSA* does not consider `Link`-allocated emissions when calculating the overall emissions effectively ignoring *Tessif*'s emission allocation.

The plausibility check executed in Paragraph 5.5.1.4 showed that the modelling approach differences concerning CHP emissions could lead to *PyPSA* employing CHP-related components more extensively, in both full load hours and installed capacity, than the other tools, resulting in a more cost-efficient solution in the context of emission-constrained scenario formulations.

In the context of this analysis, the root cause of the different CompE MSC results has been successfully identified in the different modelling approaches of *Tessif*'s and *PyPSA*' emission allocation to CHP-related components.

It is crucial to emphasise that this does not imply that *PyPSA* is misfunctioning. Instead, the current *Tessif-PyPSA* interface cannot implement *Tessif*'s modelling approach for allocating emissions to CHP component outflows. In order to counteract this discrepancy, a reallocation of the emissions to the CHP components' feeding commodities is advised. To study the impact of this measure, an additional investigation, called Modified CompE, will be conducted, where emissions are allocated to the hard coal and biogas source components instead to the CHP components so that *PyPSA* can correctly interpret the intent of *Tessif*. This Modified CompE model scenario combination is listed as MSC-number 2a in Table 5.5.

5.5.2. Executing CRA-Step 1 on the Modified Component-Expansion (Modified CompE) MSC: Inspecting the High-Priority Results (HPR)

Figure 5.18 shows the Modified CompE HPR of *Calliope*, *FINE*, *oemof* and *PyPSA*. The underlying numerical results are shown in Appendix Table A.28. The horizontal axes depict the high-priority results discussed in Subsection 4.5.1. The vertical axes show the respective result values in relation to *oemof* as coloured bars representing the individual ESSMOS results.

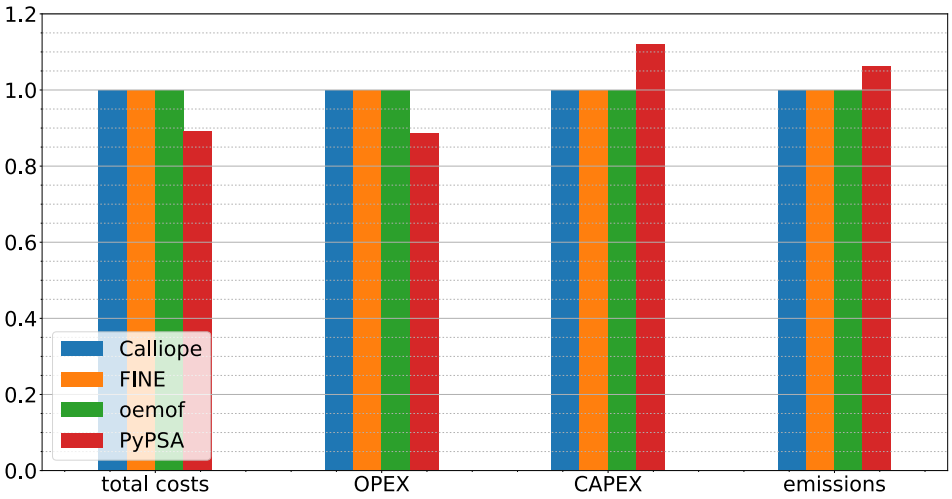

Figure 5.18.: HPR of the Compared ESSMOS Relative to *oemof* for the Modified CompE MSC.
The Depicted Numerical Results Are Listed in Appendix Table A.28.

The essential observations are:

- The high-priority results of *Calliope* and *FINE* equal those of *oemof* (compare the blue and orange coloured bar lengths of each result to the green bar length in Figure 5.18).
- The total cost result of *PyPSA* corresponds to about 0.9 times the amount of the other tools (compare the red bar length of the total cost result to the respective lengths of the differently coloured bars in Figure 5.18).
- *PyPSA*'s OPEX result is about 0.9 times the amount of the other ESSMOS (compare the red bar length of the OPEX result to the respective lengths of the differently coloured bars in Figure 5.18).
- The *PyPSA* CAPEX results correspond to about 1.1 times the amount of the other ESSMOS (compare the red bar length of the CAPEX result to the respective lengths of the differently coloured bars in Figure 5.18).
- The emission results of *PyPSA* are about 1.05 times the amount of the other tools (compare the red bar length of the emissions result to the respective lengths of the differently coloured bars in Figure 5.18).

5.5.3. Executing CRA-Step 2 on the Modified CompE MSC: Creating the Advanced System Visualisation

Figures 5.19 and 5.20 show the *oemof* and *PyPSA* Modified CompE advanced system visualisations (ASVs). The *FINE* and *Calliope* visualisations are identical to *oemof* and are, thus, omitted here. The shown ASVs depict the ESSM components as two concentric circles. The outer circle is visualised as a coloured perimeter, whereas the inner circle is drawn as a disk of equal colour. The allowed energy flow connections are depicted as greyscale-coloured arrows indicating the flow direction.

Figures 5.19 and 5.20 only depict components that provide or use energy (i.e. components with a summed-up load result greater than zero). The deployed `tessif.visualize` drawing utility automatically sorts out any components not providing or using any energy (i.e. components with a summed-up load result of zero).

The entire ASV-technique is developed, thoroughly explained and demonstrated in Paragraph 2.2.2.2. Of all the possible options discussed there, the following are applied here:

1. The outer node diameter is drawn proportional to the component's installed capacity.
2. The inner node diameter is drawn proportional to the component's characteristic value (see also Subsection 2.2.3). For energy-supplying components, this corresponds to the capacity factor. For energy storage components, this corresponds to the average state of charge divided by the installed storage capacity. For energy using-components, this represents the relation between the average amount of consumed energy and the maximum occurring load.
3. The arrow shaft width is drawn proportional to the summed-up load result of its underlying edge. The wider the edge, the more overall energy is transported.
4. The arrow greyscale is drawn proportional to the specific emissions of its underlying edge. The darker/blacker the arrow, the higher the edge's specific emission.

To facilitate the visual comparison between Figures 5.19 and 5.20, the following references are used:

- Fixed Node Size Reference:

  The `El Demand` component's installed capacity serves as an outer diameter reference and corresponds to one major or five minor the grid spacing widths. It is the same for all ESSMOS since it is externally constrained by the provided power demand profile. Consequently the depicted outer and inner diameters of the `El Demand` nodes are referring to the same numerical value in both figures.

- Fixed Arrow Shaft Width Reference:

  The (`Powerline`, `El Demand`) edge summed-up load serves as edge width reference and is the same for all ESSMOS since it is externally constrained by the provided power demand profile. The depicted shaft width of the (`Powerline`, `El Demand`) arrows is, thus, referring to the same numerical value in both figures.

- Fixed Grey-Scaling Reference:

  The (`Solar Panel`, `Powerline`) edge emissions caused result serves as a black colouring reference. All specific emission values greater or equal will cause a black-coloured arrow, and specific emissions lower than that will cause an arrow colour of lighter grey shades. The selected grey-scaling reference corresponds to the `Solar Panel` component's specific emission of $0.05\,\mathrm{t_{CO2\text{-}eq}}/\mathrm{MWh}$ as listed in the first row of Appendix Table A.1, which states the Component Commitment and Component Expansion (CompCnE) ESSM's underlying parameter assumptions.

The depicted ASVs are used here for two purposes. The first is to demonstrate the developed ASV-technique and its utility by making general observations, and the second is to identify significantly different components during the visual comparison of ISD-step one.

5.5.3.1. General Observation Results of Inspecting the Modified CompE ASVs

The following **general-ASV-observations** can be made:

1. All components depicted in Figures 5.19 and 5.20 use or provide energy, i.e., their summed-up load results are greater than zero, otherwise they would have been sorted out by the drawing algorithm.

2. The `Lignite PP`, `Hard Coal CHP`, and `Hard Coal PP` components are not depicted in Figures 5.19 and 5.20 despite having an installed capacity greater than zero (see entries six to nine in Appendix Table A.1 listing the underlying parameter assumptions). Hence, they provide no energy and are, thus, automatically sorted out by the drawing utility.

3. The Onshore Wind Turbine component supplies most of the energy (compare the shaft width of (`Onshore Wind Turbine`, `Powerline`) arrow to the shaft width of the (`Powerline`, `El Demand`) arrow in Figures 5.19 and 5.20).

4. The `Power To Heat` component supplies most heat (compare the shaft width of the (`Power To Heat`, `Heatline`) arrow to the shaft width of the (`Heatline`, `Heat Demand`) arrow in Figures 5.19 and 5.20).

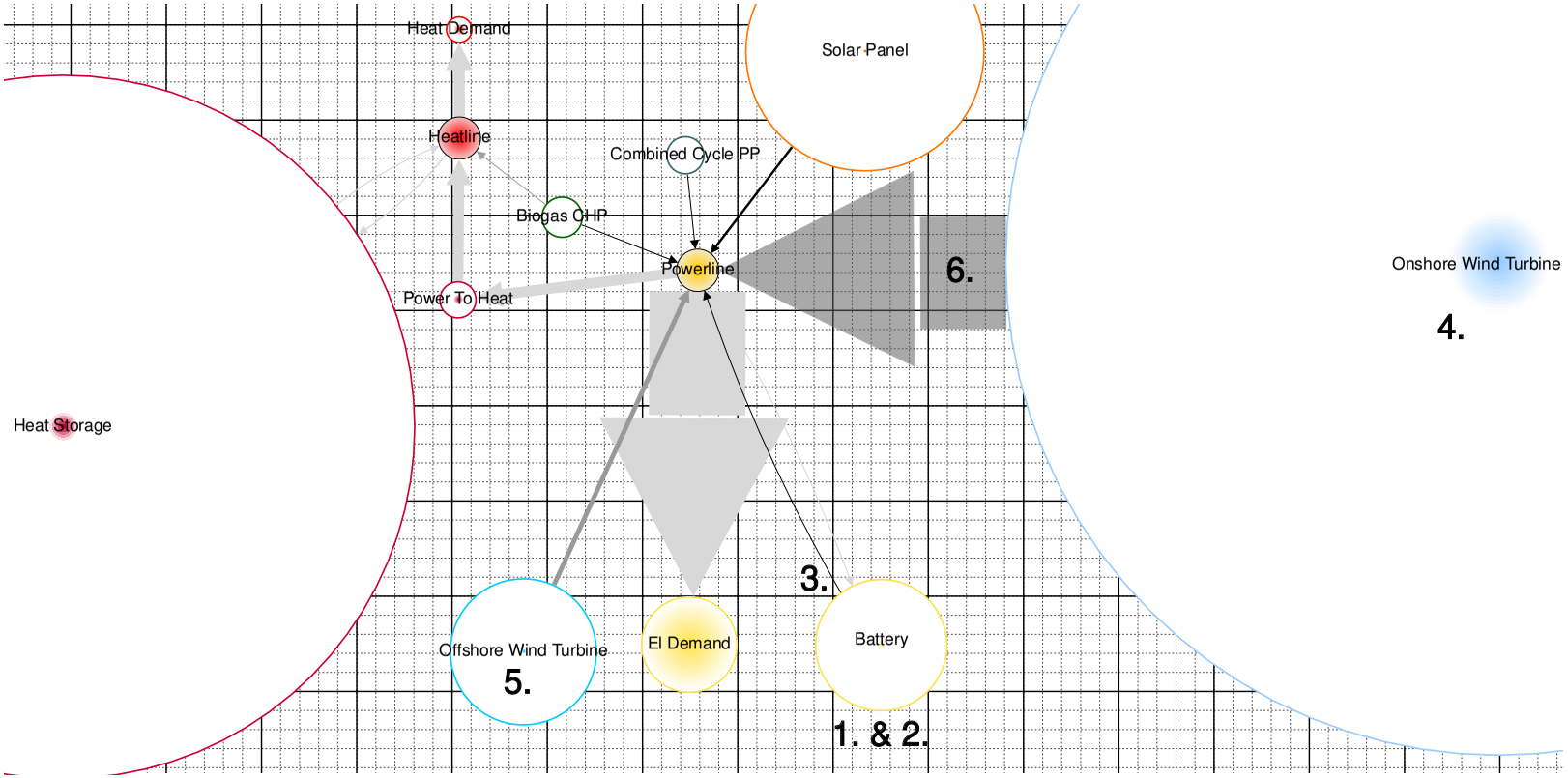

Figure 5.19.: Advanced System Visualisation of the *oemof* Modified CompE Results.
Technique Thoroughly Developed, Explained and Demonstrated in Paragraph 2.2.2.2.
Outer Node Diameter Scaling with Installed Capacity; Inner Node Diameter Scaling with Characteristic-Value.
Edge Width Scaling with Summed-Up Load Result; Edge Greyscale Darkening with Increase in Specific Emissions.
Numbers Correspond to the Comparative-ASV-Observations in Paragraph 5.5.3.2.

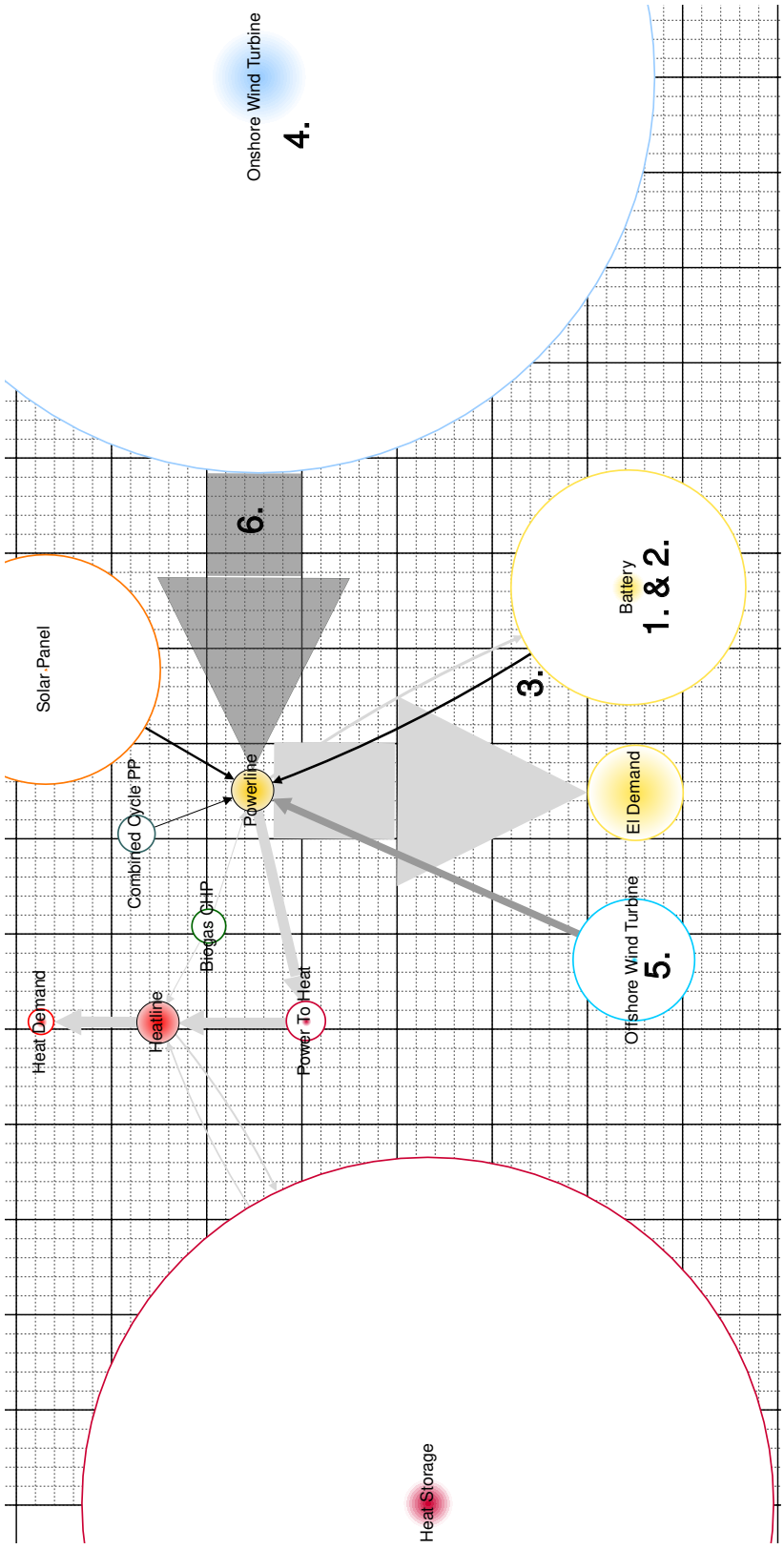

Figure 5.20.: Advanced System Visualisation of the *PyPSA* Modified CompE Results.
Technique Thoroughly Developed, Explained and Demonstrated in Paragraph 2.2.2.2.
Outer Node Diameter Scaling with Installed Capacity; Inner Node Diameter Scaling with Characteristic-Value.
Edge Width Scaling with Summed-Up Load Result; Edge Greyscale Darkening with Increase in Specific Emissions.
Numbers Correspond to the Comparative-ASV-Observations in Paragraph 5.5.3.2.

5. Of the `Powerline` feeding components, the `Battery`, the `Combined Cycle PP`, the `Biogas CHP` and the `Solar Panel` have the highest specific emission values allocated since the arrows leaving these comoponents are the blackest in Figures 5.19 and 5.20.

6. All energy-providing components have relatively large installed capacities compared to their comparatively low capacity factor. (Compare the large outer diameters of the energy-providing components to their comparatively small inner diameters in Figures 5.19 and 5.20).

7. In average, all energy storage components have their overall storing capacity (during the entire optimisation time span) utilised only by about 10%. (Compare the large outer diameters of the `Battery` and `Heat Storage` components to their comparatively small inner diameters in Figures 5.19 and 5.20).

8. The `Biogas CHP` component supplies only insignificantly low thermal and electrical energy. The `Biogas CHP` component is, nevertheless, depicted in Figures 5.19 and 5.20, implying it provides at least some energy since it was not automatically sorted out. However, the energy provided is so small that no inner visible `Biogas CHP` circle is depicted in Figures 5.19 and 5.20.

9. The same observation as in 8 applies to the `Combined Cycle PP` component.

5.5.3.2. Comparative Observation Results of Inspecting the Modified CompE ASVs

The following **comparative-ASV-observations** are made during the visual comparison of ISD-step one in Paragraph 5.5.4.1. Thereby, the listing number corresponds to the numbers depicted inside the ASVs. The observations are listed here to make them easier to follow, despite contextually belonging to ISD-step one. It is essential to note that, the points made only roughly quantify the observed differences since the primary purpose of the comparative-ASV-observations is to name and identify the differences rather than to precisely quantify them.

1. Except for the `Battery` and `Onshore Wind Turbine` component, no large striking differences can be immediately observed. The first visual impression when comparing Figures 5.19 and 5.20 leads to the conclusion of an overall seemingly similar distribution of node diameter sizes, arrow shaft widths and arrow colours. Additionally, the general observations listed in Subsection 5.5.3 apply to both Figures and, thus, to both the ESSMOS *oemof* and *PyPSA*. The Modified CompE optimisation results of *PyPSA* and *oemof* are, therefore, significantly more similar than their respective unmodified CompE MSC results (for context revisit the CompE preliminary result analysis in Subsection 5.4.3).

2. The *PyPSA* `Battery` component's installed capacity is about twice as large as *oemof*'s (more than twelve minor grid spacings for *PyPSA* and less than seven for *oemof* when comparing the outer `Battery` node diameters between Figures 5.19 and 5.20).

3. The amount of electrical energy flowing to and from the `Battery` component is roughly two times the amount for *PyPSA* than for *oemof* (compare the shaft widths of the arrows between the `Battery` and `Powerline` components in Figures 5.19 and 5.20).

4. The *PyPSA* `Onshore Wind Turbine` component's installed capacity corresponds to about 0.8 of the installed capacity of *oemof*'s `Onshore Wind Turbine` component (approximately four major grid spacings for *PyPSA* and five for *oemof* when comparing the outer `Onshore Wind Turbine` node radius between Figures 5.19 and 5.20).

5. The *PyPSA* `Offshore Wind Turbine` component's installed capacity is about 0.875 that of *oemof*'s (seven minor grid spacings for *PyPSA* and eight for *oemof* when comparing the outer `Offshore Wind Turbine` node diameters between Figures 5.19 and 5.20).
6. *PyPSA*'s `Offshore Wind Turbine` component provides about 1.2 times the amount of electrical energy compared to *oemof* (six minor grid spacings for *PyPSA* and five for *oemof* when comparing the shaft widths of the arrow between the `Offshore Wind Turbine` and `Powerline` component in Figures 5.19 and 5.20).

5.5.4. Executing CRA-Step 3 on the Modified CompE MSC: Identifying Significant Differences

All three ISD-steps are performed for the Modified CompE MSC. The obtained results are thereby discussed in great detail to facilitate understanding and to demonstrate the utility of the developed identification techniques. In an actual application, the typical process consists of a brief visual comparison using the ASVs combined with a quick confirmation utilising the automated statistical identification results. For an experienced user of this method, the entire identification process, including the figure generation and comparison, would be accomplished in less than twenty minutes.

5.5.4.1. Executing ISD Step 1: Detecting and Isolating Components of Interest

The Modified CompE optimisation result differences between *PyPSA* and the other ESSMOS are identified using the **visual comparison** and the **statistical identification**.

Visual Comparison: The **comparative-ASV-observations** relevant to this ISD-step are listed in Paragraph 5.5.3.2, closer to the advanced system visualisation figures.

Statistical Identification: The primary goal of the statistical identification process is to assist the analyst in isolating significantly different components that should be investigated further. **This statistical identification process is neither designed for nor capable of quantifying the isolated differences.** However, its main advantage is that it is fully automated, adjustable in its detection granularity, and yet very robust. **It can isolate significantly different components within seconds.** Three underlying result values are used to identify these differences:

1. The Pearson correlation coefficient *PCC* (see also equation 2.9 in Subsection 2.3.2):

   In this identification, the *PCC* value measures a component load profile's linear correlation between two ESSMOS. A high linear correlation in this context translates to a similar dispatch pattern of the same component between an inspected ESSMOS and *oemof*. Dispatch pattern, thereby, refers to **similar directions of a component's load profile slope** averaged over the entire optimisation time span. This relationship can be expressed in simpler words: If a component's load constantly changes in the same direction between the investigated ESSMOS and *oemof*, then the PCC value is calculated to one and the degree of linear relationship is, thus, very high.

   The underlying identification algorithm uses a PCC-threshold value of 0.70. This value marks the transition from a "very high" to a "high" degree of linear correlation (see also Table 2.7 in Subsection 2.3.2). A *PCC* result lower than 0.70 is, thereby, interpreted as

potentially interesting for further investigation. Revisit Table 4.3 in Paragraph 4.5.3.1 for a summary of assumed interest levels.

2. The normalised mean absolute error $NMAE$ (see also equation 2.18 in Subsection 2.3.4):

   In this identification, the $NMAE$ value measures the absolute deviation of a component's load profile between two ESSMOS averaged out and normalised over the entire optimisation time span. An $NMAE$ of zero indicates no mean absolute deviation for a component's load profile result between two ESSMOS, and the load profiles are thus identical. An $NMAE$ result of, e.g., 0.05 indicates a mean absolute deviation in the inspected ESSMOS component's load profile of 5% compared to *oemof*.

   The underlying identification algorithm uses an NMAE-threshold value of 0.1 or 10%. An $NMAE$ result greater than 0.1 is interpreted as potentially interesting for further investigation. Revisit Table 4.3 in Paragraph 4.5.3.1 for a summary of assumed interest levels.

3. The relative deviation value Δ_{rel} (see also the Δ_{rel} equation 4.1 in Paragraph 4.5.3.1):

   In this identification, the Δ_{rel} value measures the relative deviation of a component's installed capacity result compared to *oemof*. A relative deviation of 0.0 means that both components have the same installed capacity. A relative deviation of -0.02 translates to the compared component's installed capacity being 2% smaller than *oemof*'s. A Δ_{rel} value of 0.01 means that the compared ESSMOS component's installed capacity is 1% larger than that of *oemof*.

   The underlying identification algorithm utilises the two relative-deviation-threshold values 0.10 and 0.30. A relative deviation result equal to or greater than 0.30 is interpreted as a high interest level. A relative deviation less than 0.30 but equal to or greater than 0.10 is flagged as medium interest, and a result below 0.10 is marked as low interest. Revisit Table 4.5 in Paragraph 4.5.3.1 for a summary of assumed interest levels.

The entire statistical identification technique is thoroughly described in Paragraph 4.5.3.1, including all of the above result values, all their possibly occurring edge cases and two comprehensive examples used to explain this process in detail.

Tables 5.12 and 5.13 compile the most relevant Modified CompE statistical identification results. All results are given in Appendix Tables A.29 to A.33. In an application context, the desired output is usually only comprised of certain interest levels as provided through the `tessif.identify` interest keyword interface (e.g., `high` for "of-high-interest"). Tables 5.12 and 5.13, however, list all underlying result values to help understand *Tessif*'s automated recommendations better.

Table 5.12.: Compilation of the Statistical Modified CompE Capacity Identification Results. Threshold Evaluation Values are Listed in Table 4.5 in Paragraph 4.5.3.1. All Calculated Results are Listed in Appendix Tables A.32 and A.33.

#	Component	Parameter	*Calliope*	*FINE*	*oemof*	*PyPSA*
1	Battery	Interest	low	low	low	high
		Relative Deviation	0.00	0.00	0.00	0.79
2	Onshore Wind Turbine	Interest	low	low	low	medium
		Relative Deviation	0.00	0.00	0.00	-0.20
3	Offshore Wind Turbine	Interest	low	low	low	medium
		Relative Deviation	0.00	0.00	0.00	-0.17
4	Heat Storage	Interest	low	low	low	low
		Relative Deviation	0.01	0.00	0.00	-0.02

Table 5.13.: Compilation of the Statistical Modified CompE Load Profile Identification Results.
PCC Values Measure the Linear Correlation to *oemof* as Listed in Table 2.7.
NMAE Values Measure the Average Deviation to *oemof*.
Threshold Evaluation Values are Listed in Table 4.3 in Paragraph 4.5.3.1.
All Calculated Results are Listed in Appendix Tables A.29 to A.31.

#	Flow Source	Flow Target	Parameter	*Calliope*	*FINE*	*oemof*	*PyPSA*
1	Battery	Powerline	Interest	low	low	low	high
			PCC	1.00	1.00	1.00	0.59
			NMAE	0.00	0.00	0.00	3.17
2	Powerline	Battery	Interest	low	low	low	high
			PCC	1.0	1.0	1.0	0.17
			NMAE	0.00	0.00	0.00	4.40
3	Onshore Wind Turbine	Powerline	Interest	low	low	low	low
			PCC	1.0	1.0	1.0	0.93
			NMAE	0.0	0.0	0.0	0.05
4	Offshore Wind Turbine	Powerline	Interest	low	low	low	medium
			PCC	1.0	1.0	1.0	0.73
			NMAE	0.0	0.0	0.0	0.77
5	Heatline	Heat Storage	Interest	low	low	low	high
			PCC	1.00	1.00	1.00	0.61
			NMAE	0.00	0.00	0.00	1.17
6	Heat Storage	Heatline	Interest	low	low	low	medium
			PCC	1.00	1.00	1.00	0.80
			NMAE	0.00	0.00	0.00	0.53

Using the above-explained result values, the following **comparative-statistical-observations** can be made:

1. All *Calliope* and *FINE* interest levels are identified as low, and all underlying result values are either the same as *oemof*'s or insignificantly different (compare the *Calliope*, *FINE* and *oemof* columns in Tables 5.12 and 5.13).

2. Only four high interest level occurrences are identified. Three of these concern the `Battery` component (see the interest results of entry one in Table 5.12 and of entries one and two in Table 5.13). The remaining occurrence concerns the `Heat Storage` component inflow (see the interest result of entry four in Table 5.12).

   In contrast, the statistical identification results of the CompE MSC detect 18 occurrences

of high interest for the inspected load profiles (Appendix Table A.13) and 12 occurrences for the inspected installed capacity results (Appendix Table A.16).[6]

This observation matches comparative-ASV-observation-1 in that the Modified CompE optimisation results of *PyPSA* and *oemof* are significantly more similar than their respective unmodified CompE MSC results.

3. The *PyPSA* `Battery` component's installed capacity is 79% larger than *oemof*'. This observation matches comparative-ASV-observation-2 and is flagged with an interest level of high (refer to the interest level and relative deviation results in entry one of Table 5.12).

4. The discharging load profile of *PyPSA*'s `Battery` component exhibits a "high" degree of linear correlation with *oemof*, as indicated by a PCC result of 0.59. However, the mean absolute deviation between *PyPSA*'s and *oemof*'s `Battery` component's discharging load profile is 317%, indicating a substantial discrepancy. Consequently, the identification algorithm classifies the *PyPSA* `Battery` component's load profile as highly interesting (refer to the PCC, $NAME$ and interest level results in entry one of Table 5.13).

   Comparative-ASV-observation-3 identifies a higher total amount of energy flowing from the `Battery` component to the `Powerline` component for *PyPSA*. Combining these three observations indicates that more energy is discharged from *PyPSA*'s `Battery` component while the "high" degree of linear correlation ($PCC = 0.59$) implies that the discharge timings and the general gradient directions of the discharging load profiles are comparatively similar between *oemof* and *PyPSA*.

5. With a PCC result of 0.17, the charging load profile of *PyPSA*'s `Battery` component is evaluated as having a "low" degree of linear correlation compared to oemof. The mean absolute deviation of *PyPSA*'s `Battery` component's charging load profile is 440% compared to *oemof*, and, thus, very high. Overall, the load profile is classified as highly interesting by the identification algorithm (refer to the PCC, $NMAE$ and interest level results of entry two in Table 5.13).

   Comparative-ASV-observation-3 identifies a higher total amount of energy flowing from the `Powerline` component to the `Battery` component for *PyPSA*. Combining these observations shows that *PyPSA*'s `Battery` component is charged with more energy than *oemof*'s. In addition, the "low" degree of linear correlation ($PCC = 0.17$) indicates that the charging timings and the general gradient directions of the charging load profiles differ significantly between the two tools.

   Considering comparative-statistical-observation-4, it can be concluded that *PyPSA*'s `Battery` component is generally charged more at different times compared to *oemof*'s. In contrast, the discharge timings of the `Battery` component are similar between the tools, albeit more energy is discharged from *PyPSA*'s `Battery` component than from *oemof*'s.

6. The *PyPSA* `Onshore Wind Turbine` component's installed capacity is 20% smaller than *oemof*'s. This observation matches the estimation of comparative-ASV-observation-4 and is flagged with an interest level of medium (see the relative deviation and interest level results in entry two of Table 5.12).

   With a PCC result of 0.93, the load profile of *PyPSA*'s `Onshore Wind Turbine` component is evaluated as having a "very high" degree of linear correlation with *oemof*. The mean

[6] The CompE statistical identification result tables are not shown here for brevity since the number count of identified high interest levels is the only result used in this analysis.

absolute deviation of *PyPSA*'s `Onshore Wind Turbine` component's load profile is 5% compared to *oemof*, and, thus, very small. Consequently, the identification algorithm classifies the load profile as lowly interesting (see the PCC, $NMAE$ and interest level results of entry three in Table 5.13).

The observed differences in *PyPSA*'s installed capacity of the `Onshore Wind Turbine` component compared to *oemof* is likely a side effect of the higher installed capacity of *PyPSA*'s `Battery` component and the associated more favourable time shift of the volatile renewable energy consumption.

7. The *PyPSA* `Offshore Wind Turbine` component's installed capacity is 17% smaller than *oemof*'s. This observation matches the estimation of comparative-ASV-observation-5 and is flagged with an interest level of medium (see the relative deviation and interest level results in entry three of Table 5.12).

   With a PCC result of 0.73, the load profile of *PyPSA*'s `Offshore Wind Turbine` is evaluated as having a "very high" degree of linear correlation compared to *oemof*. However, the mean absolute deviation of *PyPSA*'s `Offshore Wind Turbine` component's load profile is 77% compared to *oemof*, and, thus, comparatively large. Overall, the identification algorithm classifies the *PyPSA* `Offshore Wind Turbine` component's load profile as medium interesting (see the PCC, $NMAE$ and interest level results of entry four in Table 5.13).

   Comparative-ASV-observation-6 identifies a higher total amount of energy flowing from *PyPSA*'s `Offshore Wind Turbine` component to the `Powerline` component. Combining these two observations shows that more energy is provided by *PyPSA*'s `Offshore Wind Turbine` component than from *oemof*'s. However, the "very high" degree of linear correlation implies that the points in time of dispatch and the general gradient directions of the load profiles must be comparatively similar.

   The observed differences in *PyPSA*'s installed capacity and load profile of the `Offshore Wind Turbine` component compared to *oemof* are likely a side effect of the higher installed capacity of *PyPSA*'s `Battery` component and the associated more favourable time shift of the volatile renewable energy consumption.

8. The relative deviation of *PyPSA*'s `Heat Storage` component's installed capacity is -2%. The interest level is classified as low (see the relative deviation and interest level results in entry four of Table 5.12).

   The load profiles of *PyPSA*'s `Heat Storage` component are classified as highly interesting for the inflow and medium interesting for the outflow. With PCC results of 0.61 for the inflow and 0.80 for the outflow, the degree of linear relationship to *oemof*'s results is evaluated as "high" for the inflow and "very high" for the outflow. The mean absolute deviation of *PyPSA*'s `Heat Storage` load profile results is 117% for the inflow and 53% for the outflow compared to *oemof* (refer to the PCC, $NAME$ and interest level results of entries five (inflow) and six (outflow) in Table 5.13).

   The combination of high linear correlation and high mean average deviation indicates similar load profiles but different magnitudes. However, the visual ASV comparison yielded no results on the different amounts of energy flowing from and to the `Heat Storage` component between *oemof* and *PyPSA* because the arrow shafts are very thin (see the (`Heat Line`, `Heat Storage`) and (`Heat Storage`, `Heat Line`) arrow shaft widths in Figures 5.19 and 5.20). Therefore, the amount of energy flowing in and out of the `Heat Storage` component is comparatively little and the resulting impact is evaluated as low.

5.5.4.2. Executing ISD Step 2: Detecting and Isolating Differing Timeframes

Since the `Battery` component emerges as the most significantly different, its charging load profile is inspected further to analyse distinct dispatch behaviours between the ESSMOS. This analysis involves isolating markedly different timeframes within the `Battery` component's inflow results using ISD-step two, as detailed in Paragraph 4.5.3.2.

Figure 5.21 depicts four of the 152 identified timeframes associated with the `Battery` component's charging load profile. The visualisation exclusively comprises the above-threshold time-varying result differences compared to *oemof*. These differences are calculated for each time step and ESSMOS using equation 4.2 with a 10% threshold. When the computed difference falls below this threshold, the differing value is replaced with *oemof*'s result. Conversely, if the calculated deviation exceeds the threshold, the ESSMOS result is retained. The algorithm is also able to isolate different sequences of these above-threshold deviations as distinct timeframes.

Figure 5.21 displays four such automatically detected timeframes, each represented as an individual step plot. The horizontal axes indicate the separate timestamps, while the vertical axes show the `Battery` component's inflow megawatts. Upon examining Figure 5.21, the following observations can be made:

1. No differences are evident among *Calliope*, *FINE* and *oemof* as indicated by the absence of blue and orange steps.
2. The increased installed capacity of *PyPSA*'s `Battery` component is leveraged in timeframes one, three and four (compare the maximum of the red steps and the green steps in subplots one, three and four of Figure 5.21). This observation is in line with comparative-ASV-observation-2 and comparative-statistical-observation-3.
3. The energy stored within *PyPSA*'s Battery component is significantly greater in timeframes one, three and four (compare the area under the curve for the red and green steps in subplots one, three and four of Figure 5.21). This observation is in line with comparative-ASV-observation-3 and comparative-statistical-observation-5.
4. The duration of *PyPSA* charges the `Battery` component exceeds that of *oemof* in all four timeframes (compare the number of subsequent timestamps between the lowest non-zero flow results for the red and green steps in Figure 5.21). This observation matches the conclusions drawn in comparative-statistical-observation-6 and 7 in that *PyPSA*'s increased `Battery` component capacity leads to an overall more cost-efficient time shift and use of volatile renewable energy.
5. *PyPSA* begins charging its `Battery` prior to *oemof* (compare the timestamps of the continuous positive shifts between two consecutive steps for the red and green lines in Figure 5.21). A similar conclusion is drawn in comparative-statistical-observation-5.

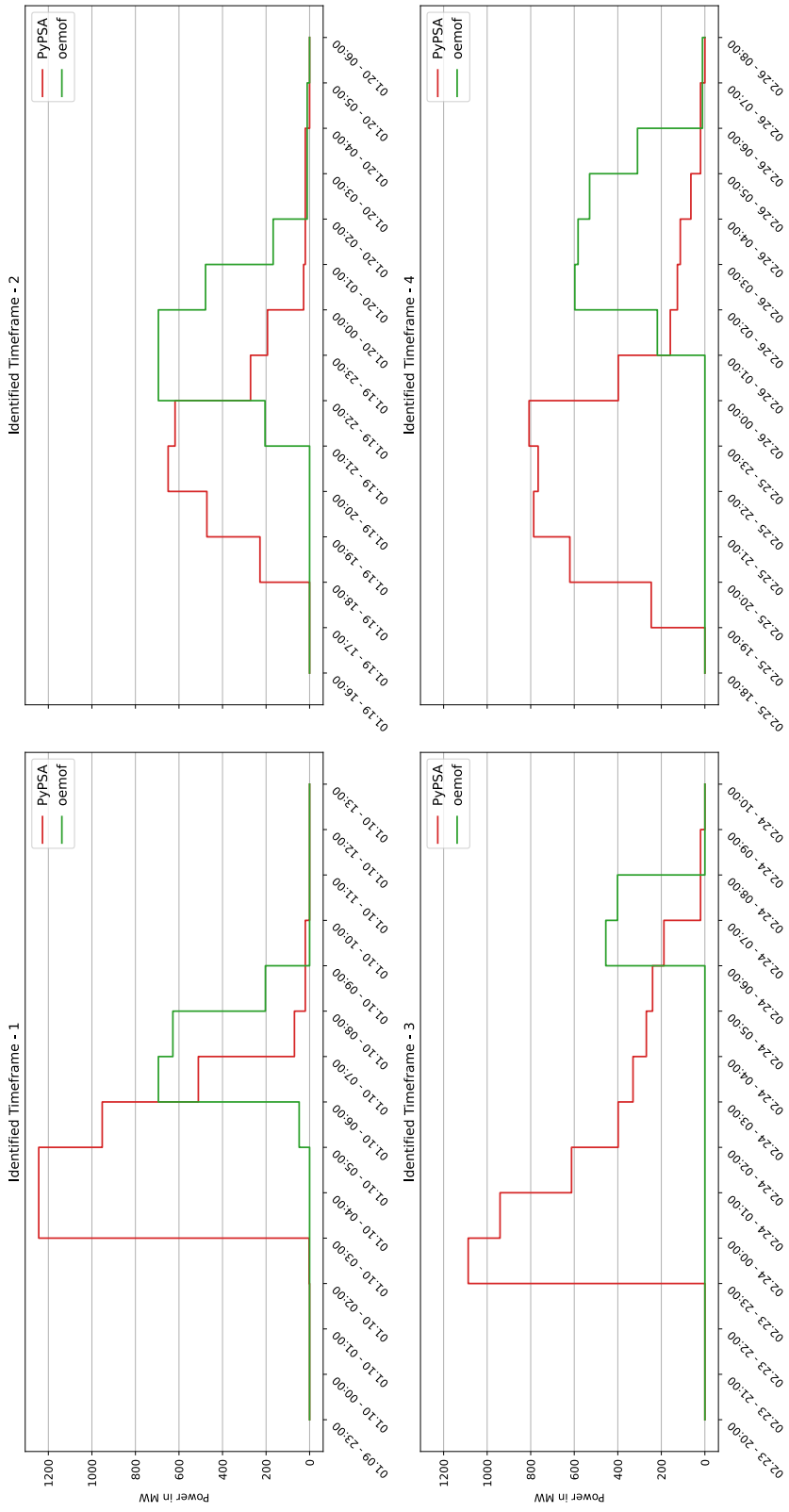

Figure 5.21.: Modified CompE Battery Charging Load Profile Result Timeframes Significantly Different from *oemof*.
Only the Above-Threshold Time-Varying Result Differences are Shown; Threshold = 10%.
No Significant Differences Were Detected between *Calliope*, *FINE* and *oemof*.
The Timeframes are Isolated Based on Equation 4.2 Using the `tessif.identify` Module.
The Technique of Isolating Significantly Different Timeframes is Explained in Paragraph 4.5.3.2.

5.5.4.3. Executing ISD Step 3: Selection for Further Analysis

Upon identifying different components and timeframes, both visually and with the assistance of `tessif.identify`, the subsequent conclusions are drawn regarding potential root causes requiring further investigation.

- *PyPSA* resolves the Modified CompE MSC at reduced total costs but with higher overall emissions than *Calliope*, *FINE*, and *oemof* (refer to Figure 5.18 in Subsection 5.5.2).
- The `Battery` component has comparatively high specific emissions allocated to its outflow for all ESSMOS (refer to general-ASV-observation-5 in Paragraph 5.5.3.1).
- *PyPSA* expands the `Battery` component installed capacity further than other ESSMOS and utilises this additional capacity (refer to comparative-ASV-observation-2 in Paragraph 5.5.3.2, to comparative-statistical-observation-3 in Paragraph 5.5.4.1, and to timeframe-observation-2 and 3 in Paragraph 5.5.4.2).
- *PyPSA*'s `Battery` component exhibits distinct charging patterns compared to other ESSMOS, storing more energy for potentially extended durations (see comparative-statistical-observation-4 and 5 in Paragraph 5.5.4.1 and timeframe-observations-4 and 5 in Paragraph 5.5.4.2).
- The installed capacity of *PyPSA*'s volatile renewable energies is lower than that of other ESSMOS (refer to comparative-ASV-observations-5 and 6 in Paragraph 5.5.3.2, and to comparative-statistical-observation-6 and 7 in Paragraph 5.5.4.1).

It is highly plausible that *PyPSA* interprets storage-allocated emissions differently from other ESSMOS, including *Tessif*. This interpretation allows *PyPSA* to expand and utilise the `Battery` component more cost-effectively, storing increased energy for longer periods. This difference results in the more efficient use of available volatile renewable energy, enabling *PyPSA* to require less installed volatile renewable capacity.

Consequently, a root cause analysis will be conducted, concentrating on the interaction between storage components and allocated emissions within an emission-constrained problem formulation.

5.5.5. Executing CRA-Step 4 on the Modified CompE MSC: Analysis for Identifying Root Causes

The Modified CompE MSC identification of root causes (IRC) employs IRC-steps 1 through 5, as outlined in Paragraphs 4.5.4.1 to 4.5.4.5.

5.5.5.1. Executing IRC-Step 1: The Modified CompE Investigative Scenario Formulation

Following the selection for further analysis of ISD-step three (Paragraph 5.5.4.3), an investigative scenario formulation is derived, focusing on this central question:

Does the abstract *PyPSA* storage-related component template interpret the allocated emissions as intended by *Tessif*, and how does this compare to the other ESSMOS?

Three **storage-investigations** are conducted to address this question:

1. Perform a tabular parameter comparison of the component templates underlying the Modified CompE MSC `Battery` and `Heat Storage` components to ensure the presence of an emission parameter and its accurate allocation. These templates are grouped by the term "storage-related" since the exact template names differ between ESSMOS.
2. Confirm the correct allocation of emissions in *Tessif* and *PyPSA*, through a parameter check.
3. Perform a plausibility check by creating a small ESSM centred around a singular storage component with outflow allocated emissions and a non-cyclic state of charge constraint. This storage component should cover the energy demand until an emission threshold is reached. The remaining demand should then be met by a more expansive but less emitting source component.

5.5.5.2. Executing IRC-Step 2: The Tabular Parameter Comparison of the Storage-Related Component Template

Table 5.14 briefly compares the storage-related component template parameters between *Tessif* and *PyPSA*. The complete comparison for all assessed ESSMOS is available in *Tessif*'s online documentation[7] [61]. The fist row displays the storage-related template name, and the second row indicates the internally used interface for addressing the created component. The third row lists the ESSMOS's component template's emission allocation parameter and indicates that *PyPSA*'s `StorageUnit` component template inherently lacks such parameter.

As discussed in Paragraph 5.5.1.2, *PyPSA* employs separate `Carrier` objects for assigning emissions to predefined (primary) energy carrier specifiers. *Tessif* employs these specifiers to create a component-specific labelled `Carrier` object with allocated emissions. Consequently, *Tessif* generates a label termed `storage_component_name.carrier`, enabling *PyPSA* to compute the storage component's emissions with a component-specific energy carrier.

Thus, the tabular parameter comparison, confirms the capability to allocate emissions to storage-related component templates, addressing storage-investigation-1.

Table 5.14.: Tabular Parameter Comparison Excerpt of *PyPSA*'s and *Tessif*'s Storage Component Template. The Full Table is Listed in *Tessif*'s Online Documentation [61].

#		*Tessif*	*PyPSA*
1	Component Template	Storage	StorageUnit
2	Identifier	Uid	name
3	**Energy Flow Specific Emissions**	`flow emissions`	-

5.5.5.3. Executing IRC-Step 3: The Storage Component Parameter Check

During the Modified CompE MSC creation, the `Battery` and `Heat Storage` components are created by explicitly parameterising *Tessif*'s `Storage` component template. The parameter check results in Table 5.15 show the relevant emission allocation parameters after the ESSM

[7] *Tessif*'s TPC of storage-related component templates: https://tessif-phd.readthedocs.io/en/latest/usage/supported_models/tabular_parameter_comparison.html#storage

is created. These results are obtained by examining the created *Tessif* ESSM using its public interface. The specific emission values are expressed in $t_{CO2\text{-}eq}$/MWh.

Table 5.15.: Emission Allocation Parameter Check of *Tessif*'s `Battery` and `Heat Storage` Components of the Modified CompE MSC. Specific Emission Values are Expressed in $t_{CO2\text{-}eq}$/MWh. The Full Parameterisation can be Found in *Tessif*'s Example Library [40].

Component	Parameter	Parameter Values
`Battery`	`flow_emissions`	`{"electricity": 0.06}`
`Heat Storage`	`flow_emissions`	`{"hot_water": 0}`

The parameter check results indicate that the Modified CompE MSC `Battery` component has a specific emission value of $0.06\,t_{CO2\text{-}eq}$/MWh. In contrast, the `Heat Storage` component has an allocated emission value of $0\,t_{CO2\text{-}eq}$/MWh (see parameter values column in Table 5.15). Comparing these parameter check results with the initial parameterisation stated in Appendix Table A.1 shows that *Tessif* allocates these correctly.

To optimise *Tessif*'s ESSM using *PyPSA*, it must be transformed into *PyPSA*'s ESSM representation. *Tessif*'s `Storage` components (i.e. the `Battery` and the `Heat Storage` component) are transformed using *PyPSA*'s `StorageUnit` component template. Table 5.16 presents the *PyPSA* `StorageUnit` component parameter check. As *PyPSA* does not allocate emissions directly to the `StorageUnit` component (see also Paragraph 5.5.5.3), the corresponding `Carrier` objects are also examined.

Table 5.16.: Emission Allocation Parameter Check of *PyPSA*'s `Battery` and `Heat Storage` Component of the Modified CompE MSC.

Component	Corresponding `Carrier` Object	Carrier Allocated Emission
Battery	`Battery.carrier`	0.06
Heat Storage	-	-

The *PyPSA* parameter check results indicate that the Modified CompE MSC `Battery` component corresponding `Carrier` object has a specific emission value of $0.06\,t_{CO2\text{-}eq}$/MWh. The `Heat Storage` component has no corresponding `Carrier` object, and no emissions allocated.

The initial goal of allocating emissions to *PyPSA*'s `Battery` component was, therefore, successfully reached. The parameter check, hence, verifies the correct emission allocation, thus, addressing storage-investigation-2. Nevertheless, the observed difference in the Modified CompE emission results remains to be explained.

Upon inspecting *PyPSA*'s emission calculation method, it becomes evident that only the difference between the final and initial state of charge is considered when calculating storage-component-related emissions. *PyPSA*'s online documentation states [93]:

"*Emissions can come from generators whose energy carriers have CO2 emissions and from stores and storage units whose storage medium releases or absorbs CO2 when it is converted. Only stores and storage units with non-cyclic state of charge that is different at the start and end of the simulation can contribute.*" [8]

[8] Online documentation of *PyPSA*'s emission calculation: https://pypsa.readthedocs.io/en/v0.19.3/optimal_power_flow.html#global-constraints

In contrast, *Tessif* allocates emissions to the storage component's inflow or outflow, modelling specific emissions per energy unit charged or discharged. Thus, *Tessif*'s and *PyPSA*'s approaches to modelling storage component emissions fundamentally differ.

5.5.5.4. Executing IRC-Step 4: The Storage Component Plausibility Check

To examine the effects of differing storage emissions modelling approaches between *Tessif* and *PyPSA*, a plausibility check (PLC) is conducted, as outlined in Paragraph 4.5.4.4. A distinct ESSM is created using *Tessif*'s abstract component templates, deliberately designing each parameter, component identifier, interconnection, and the overall emission constraint. The primary aim is to compel the solver to utilise the storage component to the extent permitted by the emission constraint before switching to a costlier alternative, enabling the analysis of the storage component-emission relationship as outlined in storage-investigation-3. The system remains abstract and generic by supplying only the essential parameters, allowing for the isolation and comprehension of underlying mechanisms without delving into deeper optimisation problem formulation layers (refer to the ESSMOS-tool abstraction layers discussion in Subsection 2.1.8). The system uses abstract specifier names, as no actual energy system component is modelled within the context of this plausibility check. The underlying ESSM is available in *Tessif*'s example library[9] [40].

The generic system visualisation (GSV) of the ESSM created for this PLC is depicted in Figure 5.22, alongside the carefully chosen parameterisation in Table 5.17. Thereby, CU represents cost unit, EU denotes energy unit, and EMU signifies emission unit. To facilitate comprehension of how this PLC instigates the switch between discharging the `Energy Storage Component` and utilising the `Energy Source Component 2`, the following core aspects were considered during parameter and ESSM design:

1. The optimisation encompasses four time steps.
2. The optimisation's objective is the minimisation of total costs.
3. The overall emission constraint is set to 20 EMU.
4. The `Energy Demand Component` requires 10 EU per time step and no specific costs or emissions are allocated to its inflow (refer to the `flow_rates`, `flow_costs` and `flow_emissions` rows of the `Energy Demand Component` in Table 5.17).
5. The `Energy Storage Component` enables power flow between −10 and 10 EU per time step as determined by the optimisation process. It has an installed capacity of 100 EU, an initial SOC of 0 EU, and no specific flow costs are allocated (refer to the `flow_rates`, `installed capacity`, `initial_soc`, and `flow_costs` rows of the `Energy Storage Component` in Table 5.17). The crucial design decision is the chosen **outflow**-specific emission value of 1 EMU/EU (see also the `outflow_emissions` row of the `Energy Storage Component` in Table 5.17). Consequently, a 10 EU discharge generates 10 EMU from the `Energy Storage Component`.
6. The `Energy Souce Component 1` provides 110 EU at the first time step and 0 EU at the subsequent steps. No specific costs or emissions are allocated to its outflow (refer to the `flow_rates`, the `flow_costs` and `flow_emissions` rows of the `Energy Source Component 1` entry in Table 5.17). Its sole purpose is to fully charge the `Energy Storage`

[9] Code and access to the storage-emissions ESSM: https://tessif-examples.readthedocs.io/en/latest/source/examples/plausibility/storage_emissions.html

Component	Parameter		Value
Energy Demand Component	flow_rates	$t=1$:	10EU
		$t=2$:	10EU
		$t=3$:	10EU
		$t=4$:	10EU
	flow_emissions		0 EMU/EU
	flow_costs		0 CU/EU
Energy Storage Component	float_rates	$t=1$:	$[-10,10]$ EU
		$t=2$:	$[-10,10]$ EU
		$t=3$:	$[-10,10]$ EU
		$t=4$:	$[-10,10]$ EU
	installed_capacity		100 EU
	initial_soc		0 EU
	outflow_emissions		**1 EMU/EU**
	flow_costs		0 CU/EU
Energy Source Component 1	flow_rates	$t=1$:	110 EU
		$t=2$:	0 EU
		$t=3$:	0 EU
		$t=4$:	0 EU
	flow_emissions		0 EMU/EU
	flow_costs		0 CU/EU
Energy Source Component 2	flow_rates	$t=1$:	$[0,10]$EU
		$t=2$:	$[0,10]$EU
		$t=3$:	$[0,10]$EU
		$t=4$:	$[0,10]$EU
	flow_emissions		0 EMU/EU
	flow_costs		**1 CU/EU**

Table 5.17.: Parameterisation of the Storage-Emissions PLC. The **total emissions are limited to 20 MU**. The optimisation time span length is 4 steps. The parameterisation can also be accessed through *Tessif-Examples'* online documentation [95].

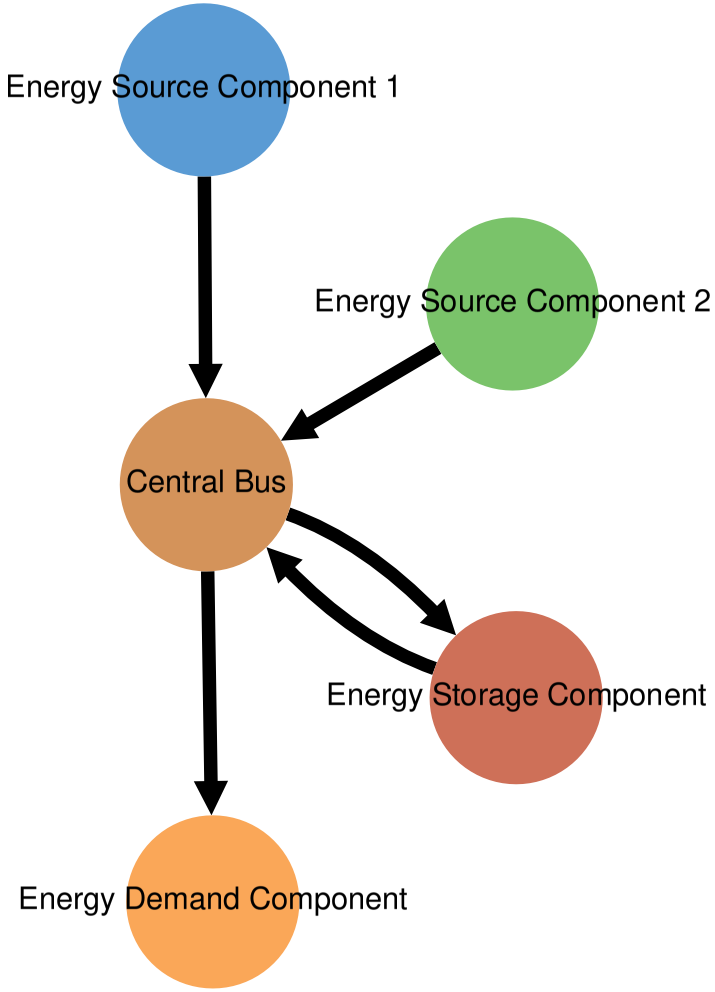

Figure 5.22.: GSV of the Storage-Emissions. Component parameters are listed in table 5.17 using the same name. All components and parameters are explicitly designed for this PLC and can be accessed through *Tessif-Examples* [40].

`Component` during the first time step while also providing the energy for the `Energy Demand Component`. This design choice derives from *FINE*'s initial SOC restriction, only allowing an initial SOC of zero. By "manually" charging the `Energy Storage Component`, a focused examination of the storage-emission relationship is enabled, isolated from the differences in initial SOC modelling between the ESSMOS.

7. The `Energy Source Component 2` allows power flow from 0 and 10 EU per time step, dependent on the optimisation results. While no specific emissions are allocated, the provided energy entails a cost of 1 CU/EU (refer to the `flow_rates`, `flow_costs`, and `inflow_emissions` rows of the `Energy Source Component 2` in Table 5.17). Thus, providing the required 10 EU for the `Energy Demand Component` in one time step results in a cost of 10 CU attributed to the `Energy Source Component 2`.

The following optimisation result expectations can be formulated based on the parameterisation above:

1. In the first time step, the `Energy Source Component 1` supplies the energy required by the `Energy Demand Component`.
2. In subsequent time steps, energy requirements are met by using 10 EU from the `Energy Source Component 2` and discharging 20 EU from the `Energy Storage Component`. Consequently, the emission limit is fully exploited, resorting to the costlier yet less emitting `Energy Source Component 2` only when indispensable.
3. Given the two above expectations, the overall total cost results will be 10 CU, and the emissions will amount to 20 EU

Table 5.18 displays the Sotrage-Emissions plausibility check results. The table contains five main columns. The first column lists the four time steps. Columns two to five represent the results for *Calliope*, *FINE*, *oemof*, and *PyPSA*, respectively. Each of these four main columns is split into two columns showing total emission and cost results in the first row and the `Energy Storage Component` and `Energy Source Component 2` flow results in subsequent rows. The two components are abbreviated as `Storage` and `Source`. Flow results are expressed in EU per time step, with the sign indicating flow direction: A negative sign denotes inflow, while an omitted positive sign signifies outflow.

Table 5.18.: Results of the Storage-Emissions Plausibility Check.

	Calliope		*FINE*		*oemof*		*PyPSA*	
	`Storage`	`Source`	`Storage`	`Source`	`Storage`	`Source`	`Storage`	`Source`
	20 EMU	10 CU	20 EMU	10 CU	20 EMU	10 CU	**30 EMU**	**0 CU**
0	-100	0	-100	0	-100	0	-100	0
1	**0**	**10**	10	0	10	0	10	0
2	10	0	10	0	10	0	10	0
3	10	0	**0**	**10**	**0**	**10**	10	0

All examined ESSMOS, except for *PyPSA*, produce the expected results. *Calliope*, *FINE* and *oemof* discharge 20 EU from the `Energy Storage Component` and utilize 10 EU from the `Energy Source Component 2` (refer to the flow results for *Calliope*, *FINE* and *oemof* in Table 5.18). Conversely, *PyPSA* discharges the `Energy Storage Component` during three time steps to meet the demand, without using `Energy Source Component 2`. As a result, *PyPSA* solves the optimisation problem at lower costs but with higher emissions (see the two *PyPSA* columns in Table 5.18).

5.5.5.5. Executing IRC-Step 5: Conducting a Final Assessment of the Modified CompE MSC IRC

The tabular parameter comparison in Paragraph 5.5.5.2 demonstrated that *Tessif* and *PyPSA* provide interfaces for allocating emissions to storage-related component templates.

The parameter check in Paragraph 5.5.5.3 confirmed that a *Tessif* `Storage` component could be transformed into a *PyPSA* `StorageUnit` component while preserving the emission allocation parameter. However, the parameter check also uncovered that *PyPSA* calculates storage-allocated emissions based on the final and initial SOC difference, effectively ignoring *Tessif*'s outflow allocation.

The plausibility check executed in Paragraph 5.5.5.4 showed that the modelling approach differences concerning storage emissions could lead to *PyPSA* employing storage-related components more extensively, in both full load hours and installed capacity, than the other tools, resulting in a more cost-efficient solution.

As *Tessif*'s current implementation cannot readily address this discrepancy, Reimer investigated [63] a further modified version of the Modified CompE MSC, called Modified-Plus CompE, in which no emissions were allocated to the storage components. This modification yielded highly similar results among all ESSMOS (relative deviation less than one per cent), including the emission results, substantiating the plausibility of this root cause analysis in the context of emission-constrained scenario formulations.

In the context of this analysis, the root cause of the different Modified CompE MSC results has been successfully identified in the different modelling approaches of *Tessif*'s and *PyPSA*'s emission allocation to storage-related components.

It is important to highlight that this does not imply that *PyPSA* is misfunctioning. Instead, the current *Tessif-PyPSA* interface cannot implement *Tessif*'s modelling approach for allocating emissions to storage component outflows. However, this problem could be addressed by expanding the *Tessif-PyPSA* interface to transform the *Tessif*'s `Storage` component into a set of *PyPSA*'s `Bus`, `Store` and `Link` components described in *PyPSA*'s online documentation[10] [93]. However, this is considered an advanced topic, requiring profound expertise in the Python programming language, as well as the code bases of *Tessif* and *PyPSA*.

[10] Online documentation of *PyPSA* relevant for expanding the *Tessif-PyPSA* interface: https://pypsa.readthedocs.io/en/v0.19.3/examples/replace-generator-storage-units-with-store.html

6. Conclusion

Chapter 6 summarises and assesses the most relevant findings and accomplishments of this thesis and its underlying research. Section 6.1 presents the key insights obtained, encompassing modelling insights, tool recommendations, advice for new ESSMOS development, and typical relationships between high-priority result differences and their root causes. Section 6.2 offers an evaluative summary of the developed software framework, comparative method, model scenario combinations, and the conducted case study. Lastly, in Section 6.3, the overall success of this work is evaluated by examining the degree to which the established objectives have been met.

6.1. Key Insights

This work yields several notable findings worth emphasising. The following subsections layout the tangible results considered valuable beyond the author's research purview. These subsections are thus split into concise conclusions to facilitate future reference.

6.1.1. Key Modelling Lessons Learned

A significant drawback of ESSMOS is the inability of the utilised solver software tools to identify contradictory parameterisations. These solver software tools can only detect infeasible problem formulations. Consequently, **the modelling process should commence with the simplest possible model scenario combination**, ensuring optimisability. **The working combination should be incrementally expanded** with additional components, constraints, or timeframes, maintaining closed-loop optimisation cycles. Otherwise, rectifying a non-functional model scenario combination becomes challenging, often even impossible, due to the limited feedback.

Given the absence of comprehensive, large-scale datasets for validating the results of modelling software like *Tessif* and its supported tools, **it is crucial to provide numerous small debugging ESSMs and corresponding plausibility checks**, as outlined in Lüdcke [71]. These small-scale ESSMs are well-suited for providing confidence in the functionality of the investigated ESSMOS. Accordingly, a supplementary example library, *Tessif-Examples* [40, 95] was created and published alongside this thesis, offering several small to medium-scale examples and plausibility checks to support the scientific community in developing reliable ESSMOS.

The conducted case study demonstrated that **generating identical results on relatively large and complex ESSMs for all investigated ESSMOS is possible but not directly implied when given the same input.** However, the case study also revealed that **harmonising data input can significantly impact optimisation results generated by different ESSMOS**, particularly regarding emission allocations or the general use of storage and combined heat and power (CHP) components. To streamline the process of shifting or re-parameterising model components, *Tessif* provides a hooking[1] functionality, enabling component and ESSM adjustments. This tool allows

[1] `tessif.hooks`: https://tessif-phd.readthedocs.io/en/latest/api/frused/hooks.html

for straightforward parameter or scenario changes to fully exploit the data input harmonisation benefits while also addressing its drawbacks by offering specific non-harmonised parameter adjustments. Internal representation differences in cost and emission allocation can thus be compensated without altering the actual interpretation of the input. All aspects considered, **data-input harmonisation frameworks like *Tessif* can facilitate the comparison process significantly** by enabling seamless ESSM creation, transformation, optimisation, post-processing, and subsequent result visualisation and comparison.

The case study also showed that CHP and storage components, particularly when combined with allocated emissions, are modelled very differently across ESSMOS. These deviations can result in substantial differences in correspondingly parameterised storage and CHP components, **necessitating a careful selection process when deciding which ESSMOS to use if storage or CHP component emission allocation is required.**

Harmonising data input and output through, for example, *Tessif*, **necessitates additional computational resources**. This requirement is significant for relatively small optimisation problems (i.e., few components or timesteps), but **negligible for larger problems**, as shown by Emmel [96] and Jessen [97].

During the attempt of modelling modern energy systems, Reimer [63], Hanke [64], Körber [66], and Schoeneich [98] identified large viable ranges of relevant component parameters in contemporary literature, particularly concerning technology-allocated emission values and expansion costs. Given the data input sensitivity described above, another layer of potential difference-inducing parameters is introduced, potentially causing even more significant deviations. Considering the importance of modern energy supply system optimisation models in investigating potential transformation pathways, **this thesis strongly recommends focusing on extensive parameter-sensitivity analyses in any new related study.**

6.1.2. Key Differences Between the ESSMOS Tools

Several key differences between ESSMOS were identified throughout the modelling work conducted for the case study in Chapter 5, the *Tessif*-related student projects, and the general development of *Tessif*. These differences can be categorised into the tool's general modelling approach, the designed field of application, and inherent functionalities for input data aggregation. The ESSMOS tool *Urbs* is also included in the following analysis of key differences since it was successfully integrated into *Tessif* by the time this conclusion was written.

6.1.2.1. Different Modelling Approaches

Tools such as *oemof* and *PyPSA* employ bus components to connect all other components, while *Calliope*, *FINE* and *Urbs* do not. The latter three group their component-like entities by region or location, assuming lossless energy transport within this clustering.

The investigated ESSMOS tools vary in their approach to flow-related parameterisation. *Oemof* distinctly separates flows into individual entities, whereas all other tools only provide component interfaces, often limiting adjustability as they do not permit general flow parameterisation.

Although all compared tools are FOSS, only *oemof* consistently and comprehensively embraces this aspect, being the sole community-driven project designed for high modularity and easy extension.

6.1.2.2. Different Applicability Designs

Among the compared ESSMOS, only FINE offers a comprehensive toolset for a structured reduction in computational effort, lowering the threshold for complex and large-scale optimisations. This toolset includes built-in time series and spatial aggregation techniques, as well as a myopic forecast approach. *Calliope* also provides a useful toolkit, albeit less sophisticated than *FINE*'s, incorporating a simple time series aggregation strategy and a pre-built-in option for utilising a rolling or receding horizon during optimisation.

The compared tools exhibit distinct approaches to managing existing installed capacities. Only *Oemof* and *PyPSA* can expand upon pre-existing capacities, while *Calliope* and *FINE* presume zero installed capacities in their expansion problem scenarios.

Only *Calliope* and *Urbs* permit adjustments to the actual optimisation variable. *Urbs* allows selection between minimising global costs or emissions, while *Calliope* computes near cost-optimal alternatives, maximising the use of different technologies within a given secondary constraint on additional global costs.

Oemof was found to be particularly adept at modelling detailed and flexible combined heat and power plants (CHP) and storage systems, particularly concerning emission allocation and expandability. *Oemof* was also identified as capable of modelling mixed-integer linear problems (MILPs) due to its strict separation of flow and component entities, allowing each flow to be characterised by MILP constraints. Consequently, every component can potentially be related to MILP parameters. Notably, *FINE* and *PyPSA* also offer some MILP parameterisation, albeit to a lesser degree.

Urbs exclusively provides advanced tools for modelling and analysing demand-side management topics. Conversely, comprehensive capabilities for modelling and optimising electrical power systems and analysing associated power flows are solely present within *PyPSA*. This distinction highlights the specialised strengths of each ESSMOS, underscoring the importance of selecting the appropriate tool based on the specific requirements of a given application domain.

6.1.2.3. Different Input Data Aggregation Capabilities

Energy system modelling and corresponding scenario development rely heavily on specific input data, such as cost and emission parameters or load profiles of renewable energy sources. Identifying viable parameter ranges often necessitates considerable time and effort. Consequently, developer teams associated with *Oemof* and *PyPSA* have initiated the development of supplementary tools and databases to provide such data sets. Although these auxiliary tools operate independently of the primary modelling software, their utilisation is likely to yield greater synergy when employed alongside corresponding ESSMOS. This enhanced compatibility can be attributed to the similarity in programmatic interfaces and documentation design.

6.1.3. Overview of the Recommended ESSMOS Tool Choices and Their Suggested Use Cases

Based on the compared ESSMOS analysis, specific tool selection and application recommendations can be provided, as displayed in Table 6.1. All mentioned tools can be used through Tessif, albeit not always to their maximum potential. In cases of uncertainty regarding tool selection, *Tessif* proves valuable, facilitating the creation of an initial ESSM compatible with

all supported ESSMOS. This ESSM can be converted into specific representations of one or multiple supported ESSMOS. This ESSMOS-specific ESSM can then be modified utilising the respective tool's capabilities, thus catering to individual modelling requirements and harnessing the ESSMOS' full range of features. An exemplary use case illustrating the synergistic application of *Tessif* and *Calliope* can be found in the works of Reimer [70]. Synergistic use cases of this type are designated as "syn" in Table 6.1, while "x" and "o" signify fully and partially recommended tool applications, respectively. These recommendations reflect the key insights stated in Subsections 6.1.1 and 6.1.2, as well as the author's subjective assessment based on using these tools during the case study conducted in Chapter 5 and his general research.

Table 6.1.: Overview of the Recommended ESSMOS Tool Choices and Their Suggested Use Cases.
x = Fully Recommended; o = Partially Recommended
syn = Synergetic Use Case of *Tessif* and Supported ESSMOS.

ESSMOS Use Case	*Tessif*	*Calliope*	*FINE*	*oemof*	*PyPSA*	*Urbs*
Extremely Large ESSMs	x	o	x			o
Near Optimal Result Variation	syn	x				
Choice of Optimisation Variable	syn	o				o
Emission Constrained Expansion Problem	x	o	o	x	o	o
Power System Modelling	syn			o	x	
Demand Side Management	syn			o		x
Detailed CHP Modelling	x			x		
Detailed Storage Modelling	x			x		
MILP	x		o	x	o	
Modularity and Customisability	x			x		
Load Profile Data Generation				o	x	
Demand Profile Data Generation				o		
Power Plant Data Aggregation					x	
Suitable for Student Courses	x	x	o	o	o	x

6.1.4. Recommended Use Cases for Tessif

Tessif's extensive documentation, straightforward engineering-oriented interface, and general design make it well-suited for an initial entry into energy supply system modelling. It provides a valuable introduction to modern software options for users who have not yet settled on a specific ESSMOS.

Tessif proves particularly useful when investigating multiple similar or related ESSMs. Its modular approach to formulating model scenario combinations, combined with the data manipulation capabilities of the `tessif.frused.hooks` package, greatly streamlines the permutation process.

One of *Tessif*'s most significant strengths lies in its post-processing, visualisation, transformation, and analysis capabilities for system-model-related data. Its user-friendly, well-documented [61] design allows for seamless integration with numerous projects, making it a good choice for handling data input and output tasks.

Tessif was developed, in part, to simplify and automate the comparison of its supported ESS-

MOS. However, its plausibility check feature can also be employed for general modelling tasks beyond the methodical comparison process detailed in this thesis.

Tessif is commonly used for modelling comparatively small ESSMs in localised expansion-problem scenarios, such as transforming a municipal district heating system with renewable energy sources and heat pumps. In these cases, creating the ESSM is often less labour-intensive than gathering the necessary input data. Typically, building the ESSM involves using a basic set of abstract `Source`, `Sink`, `Transformer`, `Bus` and `Storage` components, which can sometimes be achieved by simply copying and pasting from *Tessif*'s public example library [40, 95] and adjusting the parameter values accordingly. Corresponding examples using *Tessif* can be found in the works of Körber [66] and Zickert [99].

Tessif might not be the ideal first choice for highly specialised system modelling and optimisation tasks. For instance, using *Tessif* for the *PyPSA-EUR* project, which models the European power system at the transmission network level, would not be recommended due to its lack of abstract passive power transport components. However, when only a few speciality components are required, *Tessif* can be highly advantageous. It enables the creation of a general ESSM formulation compatible with each supported ESSMOS, along with a set of reference optimisation results using the harmonised data input. The specialised components and features can then be manually added using the respective ESSMOS programming interface, as *Tessif* allows extracting its transformed ESSM formulations. This approach enables the synergistic combination of *Tessif* and its supported ESSMOS.

In summary, *Tessif* serves as a versatile and accessible entry point for energy supply system modelling, offering valuable insights into modern ESSMOS. Its modular approach, data manipulation capabilities, and post-processing features make it a viable option for a wide range of applications.

6.1.5. Recommendations for Developing New ESSMOS Tools

This subsection outlines a set of core design aspects based on the work in this thesis and the critical insights discussed so far, recommending essential design features for a potential new ESSMOS. The focus lies on two overarching aspects: general modelling design goals and targeted applicability design goals. Both design goals represent a compilation of the best features offered by the compared ESSMOS, including *Tessif*.

6.1.5.1. Recommendations for General Design and Modelling Approaches

Concerning component-related modelling approaches, it is highly recommended to implement bus-like components connecting all other entities. The logical grouping of components, as seen in *Calliope*, *FINE*, and *Urbs*, was found cumbersome and lacking in flexibility and modularity. *Tessif*'s approach of combining *oemof*'s flow-entity abstraction with component-focused parameterisation from other tools would offer the best of both approaches. Therefore, providing a flow-like parameterisation interface, similar to *Tessif*, within the actual components is recommended.

Tessif's approach to modelling an ESSM using a discrete mathematics graph proved highly useful, especially when combined with the result data representation introduced in Subsection 2.2.3. Both are, thus, recommended for inclusion.

The modular, easily extensible, and community-driven approach of *oemof* is regarded as a valuable contribution to the scientific community and the interested public. Combining this with overarching goals and development efforts, as demonstrated by the teams behind *Calliope*, *FINE*, and *PyPSA*, is ideal for creating long-lasting, highly capable, free, and open-source modelling software.

6.1.5.2. Recommended Applicability Design

Implementing a comprehensive set of pre-built-in tools for reducing computational effort is strongly recommended to enable the modelling of extensive and complex energy supply systems, such as continental Europe. *FINE* offers excellent examples and should serve as a reference in this regard. Simpler but more straightforward implementation examples can also be found in *Calliope*.

Calliope's and *Urbs*' feature of not strictly enforcing the objective variable to be global costs is considered valuable. The *Calliope* `spores` feature of finding additional near-cost-optimal solutions maximising the use of different technologies was found to be particularly beneficial. Any new ESSMOS should be designed around such a feature, potentially offering an even broader range of possible objective variables.

Since modelling emission-constrained expansion problems of existing energy supply infrastructure is of significant public relevance, following *oemof*'s and *PyPSA*'s approach is highly recommended to allow already installed capabilities in expansion-scenario archetype problem formulations.

Regarding the modelling of specific supply system components, it is strongly recommended to adopt *oemof*'s approach for their storage and CHP-like component templates, as they were found to be an excellent compromise between accuracy, flexibility, and complexity.

Furthermore, it is strongly recommended to include modelling capabilities like *PyPSA*'s static power flow analysis to model power-system-related specifics on a given ESSM.

Concerning the investigation of demand-side management benefits, the respective modelling capabilities of *Urbs* are recommended for inclusion in any potential new ESSMOS.

With regard to modelling actual existing supply system components, mixed-integer-linear parameterisation is often considered the best compromise between modelling complexity and accuracy [54]. A newly designed ESSMOS should incorporate the MILP parameter design of *oemof* and *Tessif* while extending the number of parameters to match those found in *FINE* and *PyPSA*.

6.1.6. High-Priority Result Differences and Their Common Root Causes

Often, differences in high-priority results originate from the same category. Although challenging to prove scientifically, the author's experience in comparing energy system optimisation results may be helpful. Thus, this summary presents subjective, personal experience-based categories of relationships between observed high-priority result differences and their potential root causes. For these cases, it is assumed that obvious causalities, such as different components, parameterisation or scenario formulations, do not apply. This implies that the comparison is performed as equally as possible regarding predispositions, as it would be if using data input harminsation tools like *Tessif*.

6.1.6.1. Differing Costs and Equal Emissions

Differing costs with equal total emissions often occur in emission-constrained expansion problems. To satisfy the emission constraint, the solver must expand the installed capacities of low-emitting components (renewables, storage systems, transportation capacities). Observed differences in costs often arise from one of the following:

1. The assigned specific emissions are not respected by every ESSMOS for one or several components.
2. A non-cyclic state-of-charge (SOC) constraint was chosen for one or several storage systems, combined with ESSMOS-specific implementations that calculate storage emissions by taking the difference between the first and final time step SOC multiplied by the assigned specific emissions.

Both causes can be easily identified by inspecting the advanced system visualisation of these components and observing a difference in the black scale of their outputs. Identifying these components, however, can be less straightforward. If the identification techniques of Subsection 4.5.3 do not detect these, good places to start the manual investigation are storage-like components (Subsection 3.2.5) and connector-like components (Subsection 3.2.6).

6.1.6.2. Differing Emissions and Equal Costs

Differing emissions with equal costs often occur in non-emission-constrained dispatch problems, where the solver seeks the cost-optimal dispatch pattern of all components present. Most model implementations only distinguish between specific emissions if the total amount is constrained. However, the solver does not prefer any particular component in cases of equal operational costs but different specific emissions. This ambiguity can lead to potential differences in total emissions between ESSMOS.

Again, this cause can be easily detected using the advanced system visualisation (Subsection 4.5.2) of these components and observing that the output arrow for one tool is relatively broad and black, while the other tool's arrow is thinner and greyer.

These components are usually flagged as highly interesting by the statistical identification technique (Paragraph 4.5.3.1) since their dynamic behaviour and actual load profiles typically differ substantially.

6.1.6.3. Differing Costs and Emissions

Differing emissions with different costs can occur in both the most common scenario archetypes: emission-constrained expansion problems and non-emission-constrained dispatch problems. For the former, this combination usually implies one of the following aspects:

1. The tools differ substantially in at least one component regarding the allocation or internal calculation of emissions.
2. The emission constraint is so high that the scenario effectively becomes a non-emission-constrained dispatch problem formulation. The most common root cause for these scenario types is that efficiency or cost calculations differ for the same component between tools. An example would be the connector-like components for *PyPSA*, which implements only a one-directional efficiency, as opposed to the implementations of other tools.

Identifying causative components can be complex since the difference can be subtle or substantial, making it challenging to recommend a precise identification technique. However, a suitable starting point for manual investigation includes connector-like, storage-like, and CHP-like components.

6.2. Evaluation

The following subsections evaluate the developed Comparative Method, the developed software framework for helping to execute the method, and the conducted case study showcasing an execution of the Comparative Method using the developed software framework.

6.2.1. Evaluating the Developed Comparative Method

In the context of this thesis, a successful and functional method was developed that allows comparing ESSMOS based on their results, implying that no prior knowledge is needed on how problems are formulated mathematically or computationally. This low threshold approach makes this method a useful asset for helping decision-makers and engineers by aiding the decision process of which tool to use for what task and what order of magnitude of deviations to be expected once a selection of tools is chosen. Due to its complexity, however, the presented method has the potential not to be comprehended quickly, and its extent makes it necessary to balance gains and effort when applying it to smaller-scale problems.

The recommended preselection strategy provides excellent value regarding essential knowledge gained compared to the research effort required. It is worth noting, though, that due to the amount of work needed on the result-based comparison, a preselection is quasi-mandatory for choosing one or several tools out of a large pool of candidates.

The recommended combination of preliminary and comparative result analysis is very well suited for both relatively short and straightforward analyses, as well as long and difficult ones, due to their dynamic nature and their assistance in focusing on critical aspects guiding the chain of analysis.

The developed combination of visual and machine-supported identification of significant differences covers a broad range of applications and model complexity, making it a valuable asset, even beyond the scope of this work. That being said, the developed method has yet to be tested on very large-scale ESSMs. The statistical identification part, however, is assumed to scale well as it offers high adaptability through its parameterisation, enabling it to be of use even on very different orders of magnitude and quantity regarding the observed differences. Its default parameterisation, however, was implemented based on this work's model scenario combinations and thus most likely requires tuning for other applications.

Finally, the developed root causes identification technique compels through being simple yet powerful and indeed able to locate and isolate root causes. A possible minor critique is that the processes may include consulting ESSMOS-specific documentation or code bases, making it potentially not immediately accessible to all of the intended audience. Therefore, comparing ESSMOS using the developed method does require some programming experience to be utilised at maximum potential.

In summary, the developed method is evaluated as highly successful and valuable, aiding the scientific community in navigating the landscape of (FOS) ESSMOS. The corresponding results

of this thesis are seen as tangible and reusable due to the developed method's high applicability, which holds for both its individual steps and the method as a whole.

6.2.2. Evaluating the Developed Software Framework

Within the scope of this thesis, a highly functional, FOSS called *Tessif* was developed [62]. High emphasis was given to its extensive online-accessible documentation [61], making it a highly accessible, very reusable and easily extendible contribution to the open science community. This documentation, however, is very technical and descriptive and currently lacks a rigorous quality control mechanism to ensure language and scientific proficiency.

Tessif was chosen to be written in Python, which is deemed one of the most accessible and relevant programming languages in modern (engineering) science. In its overall very clear and transparent design process, *Tessif* aims to be a community-driven project in the future.

Tessif provides a singular interface tailored towards engineers to model what is seen as the most common energy supply system optimisation problems. This harmonisation can impact certain specialised aspects of the supported ESSMOS, for example, shown in Reimer [70], Schoeneich [98] and Hanke [100], which does not outweigh its benefits. *Tessif* was certainly not designed to replace its supported ESSMOS but rather to synergise with them, as demonstrated by the *Calliope* integration [70]. Its primary purpose is to provide a low-threshold entry point for exploring and comparing existing ESSMOS.

Currently, *Tessif* offers full-fledged support for five different FOS ESSMOS, of which four are considered the most modern and relevant. Due to its extensibility, additional ESSMOS support may be added in the future. However, the process is not trivial and requires about two months of full-time development. *Tessif* also offers a comprehensive library of abstract components that can be used to model many common ESSMs regarding the commitment and expansion optimisation problem archetypes. However, it has yet to offer the capability to model passive transport components.

A wide range of input data parsing tools is offered by *Tessif* simplifying data input. However, *Tessif* does not provide in-house input data generation tools like *Calliope*'s *solar-and-wind-potentials* [101], *FINE*'s *LoadProfilegenerator* [102]. *oemof*'s *demandlib* [103] and *feedinlib* [104] or *PyPSA*'s *atlite* [105]. These input data providing software tools, however, usually take advantage of common output data formats like arrays and key-value-parirings, which can easily be used with *Tessif*.

One major reason *Tessif* was developed was to execute the comparative method designed in this thesis. It can, therefore, automate and facilitate cross-ESSMOS comparisons to a large degree and is generally very beginner-friendly. Being able to utilise *Tessif* to its fullest, though, requires going through a noticeable learning curve. Given the comprehensive online documentation [61] and its extensive example library [40], however, this is highly achievable, even for unversed researchers.

Finally, *Tessif* is considered to be highly modular, offering a wide range of unique tools. Assisting the modelling process, ranging from data input over post-processing to result-visualisation.

Considering everything, *Tessif* is evaluated as highly beneficial to the scientific community. It is able to help advance the process of open science in the field of energy supply system modelling and capable of assisting researchers, developers and decision-makers.

6.2.3. Evaluating the Developed Model Scenario Combinations and Conducted Case Study

The case study successfully illustrated the implementation of the devised Comparative Method, encompassing all steps except for the optional CRA-step 5. The comparison of ESSMOS tools *Calliope*, *FINE*, *oemof*, and *PyPSA* revealed key disparities in their results, identifying the respective root causes utilising the resources of the software framework and its online documentation. Thereby, the developed model scenario combinations played a vital role in the demonstration's success.

Reimer [63] developed the component-focused model scenario combinations (MSCs), addressing all five modelling objectives outlined in Paragraph 3.8.1.1. The component-focused MSCs successfully integrate volatile renewable energy sources constrained by hourly resolved load profiles. Moreover, they incorporate electrical and thermal energy storage systems, modelling storage capacity, load constraints, standing losses and (dis)charging efficiencies. The MSCs also accommodate heat and power sector coupling technologies such as CHP and power-to-heat plants using maximum capacity, load profile and efficiency constraints. Reimer's component-focused MSCs can, thereby, represent both a pure commitment problem archetype and an emission-constrained expansion problem archetype.

The grid-focused model scenario combinations developed by Hanke [64] also achieved all their corresponding objectives as formulated in Paragraph 3.8.2.1. These MSCs model energy transportation losses, maximum transferable energy in grid-like entities, grid congestion issues, and subsequent redispatch requirements. Hanke's MSCs apply to both the commitment problem archetype and the (emission-constrained) expansion problem archetype, the latter explicitly modelling the interplay between grid expansion and redispatch costs.

Furthermore, the case study showed that employing a data input harmonisation framework, such as *Tessif*, can generate similar results on large and complex MSCs for different ESSMOS. However, the study also showed that discrepancies in the modelling approach of the compared tools may lead to significant deviations.

In addition to the result-based comparisons, the case study efficiently employed the preselection-steps to ascertain essential facts about potential ESSMOS candidates, leveraging the developed key criteria and factsheet comparisons.

In conclusion, the case study successfully demonstrated the apt use of the Comparative Method in conjunction with the developed software framework and the created model scenario combinations. The case study shows that even root causes deeply embedded in the mechanisms of the utilised ESSMOS can be identified by adhering to the proposed procedure. Nonetheless, the study's scope also highlights the complexity and level of detail necessary to derive coherent and well-founded conclusions, emphasising the expertise required.

6.3. Goal Assessment

In Section 1.2, six major aims were outlined for this thesis, addressing the underlying research and software development. Thereby, the second objective comprises five intermediate goals, focusing on the result-based, Comparative Method. All of these goals are revisited to analyse the extent to which they have been achieved. Table 6.2 summarises the assessment results, with columns one and two representing each goal by its number and alias. The degree of success is

denoted by the symbols "o", "+", and "++" in column three. Here, "o" represents a partially fulfilled goal, "+" a fulfilled goal, and "++" an exceedingly fulfilled goal.

Table 6.2.: Goal Assessment Summary.
o=partially fulfilled; +=fulfilled; ++=fulfilled and exceeded

Goal	Alias	Eval.
1	Preselection	+
2a	Developed Software Framework	++
2b	Developed Model Scenario Combinations	+
2c	Result Data Assignment	+
2d	Identification Strategies	+
2e	Scalability Assessment	o
3	Case Study	+
4	Key Lessons Learned	+
5	ESSMOS Recommendations	+
6	New Development Recommendations	+

Goal 1 aimed to develop a preselective submethod for narrowing down candidates for a given task. The procedure, discussed in Section 4.2 and demonstrated in Section 5.2, was developed in collaboration with Körber [66]. It proved very effective in terms of knowledge gained versus research effort required. Consequently, Goal 1 is considered well achieved.

Goal 2a required the development of free and open-source software to support the Comparative Method established in this thesis. The *Tessif* software framework [40, 62, 106] was created to fulfil this need, serving as a topic for various student projects[2] and contributing to open science. With *Tessif* in an advanced state of development and providing several different release version, [62, 106] achieving Goal 2a has exceeded initial expectations.

Goal 2b involved designing modern model scenario combinations to compare different ESSMOS tools showcasing the Comparative Method. Developed mainly by Reimer [63] and Hanke [64], these combinations are introduced in Section 3.8 and extensively used throughout Chapter 5. The analysis in Sections 5.4 and 5.5, along with the evaluation in Subsection 6.2.3, demonstrate that Goal 2b has been well met. The developed model scenario combinations effectively mimic modern energy system modelling tasks and cover various modelling approaches, revealing non-obvious differences.

Goal 2c called for a post-processing strategy to represent result data comprehensively and universally. The approach, recommended in Section 2.2, forms the core design feature of *Tessif*. It also serves as the backbone of *Tessif*'s ESSM design, extending the capabilities of supported ESSMOS to post-process and assign result data as described in Subsection 2.2.3. The case study in Chapter 5 demonstrates that the discrete-graph backend, combined with a node-and-edge-oriented result data representation, enables easy comparison among diverse ESSMOS. Therefore, Goal 2c is deemed achieved.

Goal 2d centred on strategies to detect and analyse differing results from multiple ESSMOS on the same model scenario combination. Developed in Sections 4.4 and 4.5 and demonstrated in Sections 5.4 and 5.5, these methods proved reliable, adaptable, and widely applicable. Consequently, Goal 2d is clearly met.

[2]Published student theses: https://tore.tuhh.de/search?query=tessif

Goal 2e focused on assessing the scalability of ESSMOS tools. Corresponding tools were initially developed by Emmel [96] and later improved by Jessen [97]. However, these methods were not presented within this work due to brevity and developmental delays. As a result, Goal 2e is only partially met. Nevertheless, *Tessif* fully supports the respective functionalities in its `analyze` subpackage [3], offering these capabilities to the scientific community.

Goal 3 concerned testing and showcasing the entire developed method. The case study in Chapter 5 achieves this while also fulfilling Goal 2b. Thus, Goal 3 is also reached.

From Goal 4 onwards, the focus shifts from content design to usability and applicability beyond the scope of the technical comparison. Goal 4 demands the abstraction of key modelling lessons learned. Subsection 6.1.1 addresses this by listing critical lessons learned during the research process. Based on the subjective impact of these key aspects and the frequency of the author providing similar advice to fellow researchers, Goal 4 is considered met.

Goal 5 pertains evaluating the ESSMOS comparison process, highlighting differences and offering specific recommendations. Subsections 6.1.2 and 6.1.3 summarise the knowledge gained through executing the case study, accompanying student theses, and underlying research conducted over the years. Given the author's successful advice based on this argumentation and the summaries of essential aspects for four relevant, modern FOSS tools, Goal 5 is seen as well met.

Lastly, Goal 6 requires summarising expertise gained by recommending attributes for newly developed ESSMOS. Subsection 6.1.5 provides detailed information on the most valuable aspects for designing highly useful ESSMOS, applicable to both broad range applications and specialised, energy transport-focused use cases. Consequently, Goal 6 is reached.

[3] Online documentation of *Tessif* scalability assessment functionalities: https://tessif-phd.readthedocs.io/en/latest/api/analyze.html#tessif.analyze.assess_scalability

7. Outlook

Chapter 7 closes this thesis by presenting opportunities to further ESSMOS research based on the established foundations. Separate outlooks are provided for the comparative method in Section 7.1 and the developed software framework in Section 7.2.

7.1. Extending the Comparative Method

The primary limitation of this work is the absence of a scalability comparison, as outlined in the goal assessment in Section 6.3. The Comparative Method should be extended with a detailed comparison focusing on the scalability of the compared ESSMOS. This extension should build upon the works of Emmel [96] and Jessen [97], estimating the required computational resources and quantifying the number of formulated constraints per ESSMOS. The resulting analysis will provide essential insights into the performance and efficiency of different ESSMOS, enabling researchers to make informed decisions when selecting tools for their studies.

The scalability assessment should also include comparisons of techniques for reducing the computational effort. In particular, these should involve (semi)automated processes, such as the temporal aggregation techniques [107] of *Calliope*, the spatial aggregation [81] and intermittency approach [82] techniques of *FINE*, and the varying optimisation time span length options of *FINE* [83] and *PyPSA* [108]. Due to their degree of automation within the compared ESSMOS, these processes are easily deployed, even by inexperienced researchers. Consequently, an individual comparison subcategory should be introduced to the scalability assessment step. The works of Schnute [8] and Reimer [70] provide initial comparisons for *FINE* and *Calliope* and should serve as starting points for additional investigations. Including these techniques will also facilitate a more comprehensive understanding of trade-offs between computational effort and solution quality, ultimately leading to improved decision-making in energy supply system modelling.

The MSCs developed in Section 3.8 should serve as medium-scale examples for the recommended additional investigative points. Additional combinations should be developed for very-large-scale ESSMs to cover the entire scale. For developing, testing, and debugging the additional features, the existing small-scale MSCs in *Tessif*'s example library [40, 95] can be utilised. By expanding the range of ESSMs, researchers will be better equipped to tackle increasingly complex energy supply system challenges, fostering innovation and progress in the field.

7.2. Improving and Extending the Developed Software Framework

Three short- to medium-term options exist for enhancing *Tessif*'s value to the scientific community. These options focus on applied research topics, while *Tessif*'s online documentation offers guidance on more technically in-depth improvements.

7.2.1. Reducing the Framework's Complexity and Increasing Its Longevity

The following measures have already been implemented to reduce complexity and increase the developed software framework's longevity:

- Each supported ESSMOS has been outsourced as an individual, version-specific plugin [109, 110, 111, 112]. This approach allows for separate maintenance of each plugin, minimises the codebase, and facilitates new developers' work on *Tessif* and its supported ESSMOS tool plugins. It also enables support for multiple versions of each ESSMOS tool through distinct plugins for each version.
- *Tessif*'s core functionality as an input data harmonisation framework has been published in two versions: *Tessif* [106, 113] and *Tessif-phd* [61, 62].
    1. *Tessif-phd* [62] encompasses the entire framework discussed in Chapter 3, including comprehensive documentation [61], all examples, and auxiliary features such as automated comparisons, identifications, visualisations, and data input/output handling.
    2. *Tessif* [106] represents the minimal framework required for creating, transforming, optimising, and post-processing ESSMs. Its lightweight design renders it data-agnostic and portable, compatible with various Python programming language versions and operating systems. Concise documentation [113] is provided to facilitate first contact and long-term maintenance.

In the future, helpful auxiliary tools should be outsourced as additional *Tessif-Plugins*, such as *Tessif-Visualise*, *Tessif-Identify*, *Tessif-Compare*, and *Tessif-Scale*. These features are currently published in *Tessif-phd* as individual subpackages; *Tessif-Scale* and *Tessif-Compare* reside within the analyse subpackage of *Tessif-phd* and should be separated into distinct plugins as described. Splitting the software framework into individual entities significantly increases its portability since third-party dependencies need only be installed where strictly necessary. The smaller, less daunting individual plugin codebases facilitate long-term support from various developers.

Implementing these measures to reduce complexity and increase longevity makes the *Tessif* framework more accessible and sustainable for the scientific community. The modular design facilitates collaboration and encourages continued development and support for the framework, ultimately ensuring that *Tessif* remains a helpful resource for energy supply system modelling and optimisation research.

7.2.2. Adding Support for Additional ESSMOS

Shortly after the case study conducted in Chapter 5, Hanke [100] finalised the integration of the ESSMOS tool *Urbs* [114]. However, due to challenging third-party dependencies, integration was not fast enough for *Urbs* to be included in the study and *Tessif-phd*. Thus, *Urbs* currently remains a standalone project and should be incorporated into the *Tessif* plugin ecosystem in the future.

Reimer [70] systematically compared potential ESSMOS candidates, resulting in a shortlist of highly suitable options presented in Table 7.1. As *Calliope* and *Urbs* are already implemented into *Tessif*, the recommended ESSMOS to be added include *ficus* [115], *OSeMOSYS* [116], *SWITCH* [117] and *NEMO* [118].

Table 7.1.: Recommended Shortlist of Included ESSMOS Candidates.
Taken from [70]. The Higher the Placement, the Better Found Suited.

Placement	ESSMOS
1	*Calliope*
2	*Urbs*
3	*ficus*
4	*OSeMOSYS*
5	*SWITCH*
6	*NEMO*

7.2.3. Adding a Conveyor Component Template

Tessif's initial concept aimed to implement a passive energy transport component dependent on potential levels and resistance-like parameters for modelling components like electrical lines or pipeline fluid transport. However, an initial attempt by Schoeneich [98] proved more complicated than expected, so support for these conveyor components was not pursued further for Tessif's initial release. The original component description can be found in Appendix Section A.11. These conveyor-style components are believed to have great potential for expanding the currently limited capability of modern ESSMOS to model passive energy transport phenomena. Integrating these components into *Tessif* would also improve the *Tessif-PyPSA* interface making *PyPSA*'s passive electrical transport components available to *Tessif*.

Bibliography

[1] United Nations Framework Convention on Climate Change. "The Paris Agreement", Dec 2015. [Online]. Available from: https://unfccc.int/sites/default/files/resource/parisagreement_publication.pdf. (Accessed: 2023-04-23).

[2] Umweltbundesamt. "Energieziel 2050: 100 Prozent Strom aus erneuerbaren Quellen, Jan 2010. [Online]. Available from: https://www.umweltbundesamt.de/sites/default/files/medien/378/publikationen/energieziel_2050.pdf. (Accessed: 2023-04-23).

[3] Miguel Chang, Jakob Zink Thellufsen, Behnam Zakeri, Bryn Pickering, Stefan Pfenninger, Henrik Lund, and Poul Alberg Østergaard. "Trends in tools and approaches for modelling the energy transition". *Applied Energy*, 290:116731, 2021. doi: https://doi.org/10.1016/j.apenergy.2021.116731.

[4] Hans-Kristian Ringkjøb, Peter M. Haugan, and Ida Marie Solbrekke. "A review of modelling tools for energy and electricity systems with large shares of variable renewables". *Renewable and Sustainable Energy Reviews*, 96:440–459, 2018. doi: https://doi.org/10.1016/j.rser.2018.08.002.

[5] H.-C. Gils, Hedda Gardian, Martin Kittel, Wolf-Peter Schill, Alexander Zerrahn, Alexander Murmann, Jann Launer, Alexander Fehler, Felix Gaumnitz, Jonas van Ouwerkerk, Christian Bußar, Jennifer Mikurda, Laura Torralba-Díaz, Tomke Janßen, and Christine Krüger. "Modeling flexibility in energy systems – a scenario-based comparison of power sector models". *Renewable and Sustainable Energy Reviews*, 26.03.21, 2021. doi: https://doi.org/10.1016/j.rser.2021.111995.

[6] Uwe Kastens and Hans Kleine Büning. "Modellierung: Grundlagen und formale Methoden". *Hanser eLibrary; Carl Hanser Verlag, München*, 2018. 4. erweiterte Auflage. Available from: https://doi.org/10.3139/9783446455399. (Accessed: 2023-04-23).

[7] Peter Tabeling. "Softwaresysteme und ihre Modellierung: Grundlagen, Methoden und Techniken. *SpringerLink Bücher; Springer Berlin, Heidelberg*, 2006. 4. erweiterte Auflage. Available from: https://doi.org/10.1007/3-540-29276-4. (Accessed: 2023-04-23).

[8] Darwin Schnute. "Vergleichende Analyse von Software zur Modellierung von Energiesystemen mittels Integration in ein Framework zur Transformation". Master Thesis, Technische Universität Hamburg, 2022. doi: http://hdl.handle.net/11420/12087.

[9] Tobias Falke, Stefan Krengel, Ann-Kathrin Meinerzhagen, and Armin Schnettler. "Multi-objective optimization and simulation model for the design of distributed energy systems". *Applied Energy*, 184:1508–1516, 2016. doi: https://doi.org/10.1016/j.apenergy.2016.03.044.

[10] Jean C Hourcade, R Richels, JR Robinson, and L Schrattenholzer. "Estimating the Cost of Mitigating Greenhouse Gases. *Climate Change 1995: Economic and Social Dimensions of Climate Change*, 1996:263–296, 1995.

[11] Astrid Niesse. "Verteilte kontinuierliche Einsatzplanung in Dynamischen Virtuellen Kraftwerken". PhD Thesis, Universitäet Oldenburg, 2015. Available from: https://oops.uni-oldenburg.de/2416/1/niever15.pdf. (Accessed: 2023-04-23).

[12] Yoseba Koldobika Peña Landaburu. "Optimal Allocation and Scheduling of Demand in Deregulated Energy Markets". PhD Thesis, Vienna University of Technology. Available from: https://publik.tuwien.ac.at/files/pub-et_11012.pdf. (Accessed: 2023-04-23).

[13] Ernst Worrell, Stephan Ramesohl, and Gale Boyd. "Advances in Energy Forecasting Models Based on Engineering Economics". *Annual Review of Environment and Resources*, 29(1):345–381, 2004. doi: https://doi.org/10.1146/annurev.energy.29.062403.102042.

[14] Andrea Herbst, Felipe Toro, Felix Reitze, and Eberhard Jochem. "Introduction to Energy Systems Modelling". *Swiss Journal of Economics and Statistics*, 148(2):111–135, 2012. doi: https://doi.org/10.1007/BF03399363.

[15] Hooman Farzaneh. "Energy Systems Modeling". *Springer Singapore*, 2019. doi: https://doi.org/10.1007/978-981-13-6221-7.

[16] Kris Poncelet, Erik Delarue, Daan Six, and William D'haeseleer. "Myopic optimization models for simulation of investment decisions in the electric power sector". In *2016 13th International Conference on the European Energy Market (EEM)*, Porto, Portugal, 2016, Pages 1-9. doi: https://doi.org/10.1109/EEM.2016.7521261.

[17] Lukas Glomb, Frauke Liers, and Florian Rösel. "A rolling-horizon approach for multi-period optimization". *European Journal of Operational Research*, 300(1):189–206, 2022. https://doi.org/10.1016/j.ejor.2021.07.043.

[18] Anas Abuzayed and Niklas Hartmann. "MyPyPSA-Ger: Introducing CO2 taxes on a multi-regional myopic roadmap of the German electricity system towards achieving the 1.5 °C target by 2050". *Applied Energy*, 310:118576, 03 2022. doi: https://doi.org/10.1016/j.apenergy.2022.118576. (Accessed: 2023-04-23).

[19] Julien F. Marquant, Ralph Evins, and Jan Carmeliet. "Reducing Computation Time with a Rolling Horizon Approach Applied to a MILP Formulation of Multiple Urban Energy Hub System". In: *Procedia Computer Science. International Conference On Computational Science, ICCS 2015*, Reykjavik, Iceland, 2015, Pages 2137 - 2146. Available from: https://doi.org/10.1016/j.procs.2015.05.486. (Accessed: 2023-04-23).

[20] John Forrest, Ted Ralphs, Haroldo Gambini Santos, Stefan Vigerske, John Forrest, Lou Hafer, Bjarni Kristjansson, jpfasano, EdwinStraver, Miles Lubin, Jan-Willem, rlougee, jpgoncal1, Samuel Brito, h-i gassmann, Cristina, Matthew Saltzman, tosttost, Bruno Pitrus, Fumiaki MATSUSHIMA, and to st. "coin-or/Cbc: Release releases/2.10.9", April 2023. [Online] Available from: https://doi.org/10.5281/zenodo.7820266. (Accessed: 2023-04-23).

[21] Free Software Foundation. "GLPK (GNU Linear Programming Kit)", Nov 2021. [Online] Available from: https://github.com/xypron/glpk-cli. (Accessed: 2023-04-23).

[22] David Bauer, Stefan Kirschbaum, Gregor Wrobel, Julian Agudelo, and Philip Voll. "Modellbasierte Optimierung von Energiesystemen". In: *INFORMATIK 2015*, Bonn, Germany, 2015, Pages 137-149. doi: https://dl.gi.de/handle/20.500.12116/2164.

[23] Horst W. Hamacher and Kathrin Klamroth. "Lineare Optimierung und Netzwerkoptimierung: Zweisprachige Ausgabe Deutsch Englisch 2., verbesserte Auflage". *Springer eBook Collection. Vieweg, Wiesbaden*, 2006. doi: https://doi.org/10.1007/978-3-8348-9031-3.

[24] Roland Wunderling. "Paralleler und Objektorientierter Simplex". PhD Thesis, Universität Berlin, 1996. Available from: https://opus4.kobv.de/opus4-zib/files/538/TR-96-09.pdf. (Accessed: 2023-04-23).

[25] Thomas Unger and Stephan Dempe. "Lineare Optimierung: Modell, Lösung, Anwendung". *Springer eBook Collection. Vieweg+Teubner, Wiesbaden*, 2010. doi: https://doi.org/10.1007/978-3-8348-9659-9.

[26] Franz Liebl. "Simulation: Problemorientierte Einführung 2., überarbeitete Auflage. Reprint 2018". *Oldenbourg Wissenschaftsverlag, Berlin and Boston*, 1995. doi: https://doi.org/10.1515/9783486788747.

[27] Josef Kallrath. "Lineare Optimierung und Netzwerkoptimierung: Zweisprachige Ausgabe Deutsch Englisch 2., verbesserte Auflage". *Springer eBook Collection. Vieweg, Wiesbaden*, 2006. doi: https://doi.org/10.1007/978-3-8348-9031-3.

[28] Michael Jünger and Stefan Thienel. "The ABACUS system for branch-and-ct-and-price algorithms in integer programming and combinatorial optimization". *Software: Practice and Experience*, 30:1325–1352, September 2000. doi: https://doi.org/10.1002/1097-024X(200009)30:11<1325::AID-SPE342>3.0.CO;2-T.

[29] Matthias Elf, Carsten Gutwenger, Michael Jünger, and Giovanni Rinaldi. "Branch-and-Cut Algorithms for Combinatorial Optimization and Their Implementation in ABACUS". In: *Computational Combinatorial Optimization. Lecture Notes in Computer Science*, vol 2241. Springer, Berlin, Heidelberg, 2001. Pages 157–222. doi: https://doi.org/10.1007/3-540-45586-8_5.

[30] Ksenia Bestuzheva, Mathieu Besançon, Wei-Kun Chen, Antonia Chmiela, Tim Donkiewicz, Jasper van Doornmalen, Leon Eifler, Oliver Gaul, Gerald Gamrath, Ambros Gleixner, Leona Gottwald, Christoph Graczyk, Katrin Halbig, Alexander Hoen, Christopher Hojny, Rolf van der Hulst, Thorsten Koch, Marco Lübbecke, Stephen J. Maher, Frederic Matter, Erik Mühmer, Benjamin Müller, Marc E. Pfetsch, Daniel Rehfeldt, Steffan Schlein, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Boro Sofranac, Mark Turner, Stefan Vigerske, Fabian Wegscheider, Philipp Wellner, Dieter Weninger, and Jakob Witzig. "The SCIP Optimization Suite 8.0". Technical Report, Optimization Online, December 2021. [Online] Available from: https://optimization-online.org/?p=18429. (Accessed: 2023-04-23).

[31] FINE Developer Team. FINE Stable-Version Online Documentation - ConversionPartLoad class. Dec 2022. [Online] Available from: https://vsa-fine.readthedocs.io/en/latest/sourceCodeDocumentation/components/subclasses/conversionPartLoadClassDoc.html. (Accessed: 2023-04-23).

[32] oemof-developer group. oemof.solph v0.4.4 Online Documentation - OffsetTransformer (component). Dec 2021. [Online] Available from: https://oemof-solph.readthedocs.io/en/v0.4.4/usage.html#offsettransformer-component. (Accessed: 2023-04-23).

[33] D. Connolly, H. Lund, B. V. Mathiesen, and M. Leahy. "A review of computer tools for

analysing the integration of renewable energy into various energy systems". *Applied Energy*, 87(4):1059–1082, 2010. doi: https://doi.org/10.1016/j.apenergy.2009.09.026.

[34] openmod: Open Energy Modelling Initiative. "Manifesto", 2022. [Online] Available from: https://openmod-initiative.org/manifesto.html. (Accessed: 2023-04-23).

[35] Stefan Pfenninger, Joseph DeCarolis, Lion Hirth, Sylvain Quoilin, and Iain Staffell. "The importance of open data and software: Is energy research lagging behind?". *Energy Policy*, 101:211–215, 2017. doi: https://doi.org/10.1016/j.enpol.2016.11.046.

[36] Gunther Schmidt and Thomas Ströhlein. "Relations and Graphs: Discrete Mathematics for Computer Scientists". *Springer Science & Business Media*, 2012. doi: https://doi.org/10.1007/978-3-642-77968-8.

[37] Jonas Hörsch, Henrik Ronellenfitsch, Dirk Witthaut, and Tom Brown. "Linear optimal power flow using cycle flows". *Electric Power Systems Research*, 158:126–135, may 2018. doi: https://doi.org/10.1016/j.epsr.2017.12.034.

[38] Derek L. Hansen, Ben Shneiderman, Marc A. Smith, and Itai Himelboim. "Chapter 4 - Installation, orientation, and layout". In: *Analyzing Social Media Networks with NodeXL (Second Edition)*. Morgan Kaufmann, 2020. Pages 55–66. doi: https://doi.org/10.1016/B978-0-12-817756-3.00004-2.

[39] Christopher Mueller, Douglas P Gregor, and Andrew Lumsdaine. "Distributed Force-Directed Graph Layout and Visualization". In: *EGPGV06: Eurographics Symposium on Parallel Graphics and Visualization*. The Eurographics Association, 2006. Pages 83–90. doi: http://dx.doi.org/10.2312/EGPGV/EGPGV06/083-090.

[40] Mathias Ammon, Max Reimer, Andreas Jessen, Darwin Schnute, Tim Hanke, Cedric Körber, Arne Bo Wilckens, Frederik Emmel, and Finn Theel. "tessif-examples: Version 0.3.2", April 2023. Zenodo. doi: https://doi.org/10.5281/zenodo.7849603.

[41] Arne Bo Wilckens. "entwicklung einer statistischen Methodik zur vergleichenden Ergebnisanalyse von Open Source Energy System Simulations Software". Bachelor Thesis, Technische Universität Hamburg, 2021. doi: https://doi.org/10.15480/882.4269.

[42] Wolfgang Kohn and Riza Öztürk. "Mittelwert". In: *Statistik für Ökonomen*. Springer-Lehrbuch, Pages 35–41. Springer Verlag, Berlin, Heidelberg, 2013. doi: https://doi.org/10.1007/978-3-642-37352-7_7.

[43] Udo Kuckartz, Stefan Rädiker, Thomas Ebert, and Julia Schehl. "Statistik: Eine verständliche Einführung". VS Verlag für Sozialwissenschaften / GWV Fachverlage GmbH Wiesbaden, Wiesbaden 2010. doi: https://doi.org/10.1007/978-3-531-92033-7.

[44] Norbert Henze. "Kovarianz und Korrelation". In: *Stochastik für Einsteiger*. Pages 166–178. Springer Fachmedien Wiesbaden, Wiesbaden 2018. doi: https://doi.org/10.1007/978-3-662-63840-8_21.

[45] Wolfgang Kohn and Riza Öztürk. "Kovarianz und Korrelationskoeffizient". In: *Statistik für Ökonomen*. Springer-Lehrbuch, Pages 91–95. Springer-Verlag, Berlin, Heidelberg, 2011. doi: https://doi.org/10.1007/978-3-662-64754-7_15.

[46] Karl Siebertz, David van Bebber, and Thomas Hochkirchen. "Korrelationsanalyse". In: *Statistische Versuchsplanung*. VDI-Buch, Pages 381–394. Springer Vieweg, Berlin, Heidelberg, 2017. doi: https://doi.org/10.1007/978-3-662-55743-31.

[47] Udo Kuckartz, Stefan Rädiker, Thomas Ebert, and Julia Schehl. "Korrelation: Zusammenhänge identifizieren". In: *Statistik Eine verständliche Einführung*. Pages 189–213. VS Verlag für Sozialwissenschaften / GWV Fachverlage GmbH Wiesbaden, Wiesbaden 2010. doi: https://doi.org/10.1007/978-3-531-92033-7.

[48] Cort J. Willmott and Kenji Matsuura. "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance". *Climate Research*, 30(1):79–82, 2005.

[49] Janine Freeman, Jonathan Whitmore, Leah Kaffine, Nate Blair, and Aron P. Dobos. "System Advisor Model: Flat Plate Photovoltaic Performance Modeling Validation Report". Technical Report, 2013. doi: https://doi.org/10.2172/1115788.

[50] Christian Barrot. "Prognosegütemaße", Chapter 35. In: *Methodik der empirischen Forschung*. Pages 547-560. Gabler Verlag, Wiesbaden 2009. doi: https://doi.org/10.1007/978-3-322-96406-9_35.

[51] T. Chai and R. R. Draxler. "Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature". *Geoscientific Model Development*, 7(3):1247–1250, 2014. doi: https://doi.org/10.5194/gmd-7-1247-2014.

[52] Germán Ruiz and Carlos Bandera. "Validation of Calibrated Energy Models: Common Errors". *Energies*, 10(10):1587, 2017. doi: https://doi.org/10.3390/en10101587.

[53] Cort J. Willmott. "Some Comments on the Evaluation of Model Performance". *Bulletin of the American Meteorological Society*, 63(11):1309–1313, 1982. doi: https://doi.org/10.1175/1520-0477(1982)063<1309:SCOTEO>2.0.CO;2.

[54] Torben Ommen, Wiebke Brix Markussen, and Brian Elmegaard. "Comparison of linear, mixed integer and non-linear programming methods in energy system dispatch modelling". *Energy*, 74(2):109–118, 2014. doi: https://doi.org/10.1016/j.energy.2014.04.023.

[55] Jordan T. Wilkerson, Benjamin D. Leibowicz, Delavane D. Turner, and John P. Weyant. "Comparison of integrated assessment models: Carbon price impacts on U.S. energy". *Energy Policy*, 76(4):18–31, 2015. doi: https://doi.org/10.1016/j.enpol.2014.10.011.

[56] Jan Priesmann, Lars Nolting, and Aaron Praktiknjo. "Are complex energy system models more accurate? An intra-model comparison of power system optimization models". *Applied Energy*, 255(4):113783, 2019. doi: https://doi.org/10.1016/j.apenergy.2019.113783.

[57] S. Misconel, R. Leisen, J. Mikurda, F. Zimmermann, C. Fraunholz, W. Fichtner, D. Möst, and C. Weber. "Systematic comparison of high-resolution electricity system modeling approaches focusing on investment, dispatch and generation adequacy". *Renewable and Sustainable Energy Reviews*, 153:111785, 2022. doi: https://doi.org/10.1016/j.rser.2021.111785.

[58] C. Steinbrink, S. Lehnhoff, S. Rohjans, T. I. Strasser, E. Widl, C. Moyo, G. Lauss, F. Lehfuss, M. Faschang, P. Palensky, A. A. van der Meer, K. Heussen, O. Gehrke, E. Guillo-Sansano, M. H. Syed, A. Emhemed, R. Brandl, V. H. Nguyen, A. Khavari, Q. T. Tran, P. Kotsampopoulos, N. Hatziargyriou, N. Akroud, E. Rikos, and M. Z. Degefa. "Simulation-Based Validation of Smart Grids – Status Quo and Future Research Trends". In: *Industrial Applications of Holonic and Multi-Agent Systems*. Pages 171-185. Lecture

Notes in Computer Science, Volume 10444. Springer International Publishing, Cham 2017. doi: https://doi.org/10.1007/978-3-319-64635-0_13.

[59] openmod: Open Energy Modelling Initiative. "Open Models", 2023. [Online]. Available from: https://wiki.openmod-initiative.org/wiki/Open_Models. (Accessed: 2023-04-23).

[60] Stefan Pfenninger, Lion Hirth, Ingmar Schlecht, Eva Schmid, Frauke Wiese, Tom Brown, Chris Davis, Matthew Gidden, Heidi Heinrichs, Clara Heuberger, Simon Hilpert, Uwe Krien, Carsten Matke, Arjuna Nebel, Robbie Morrison, Berit Müller, Guido Pleßmann, Matthias Reeg, Jörn C. Richstein, Abhishek Shivakumar, Iain Staffell, Tim Tröndle, and Clemens Wingenbach. "Opening the black box of energy modelling: Strategies and lessons learned". *Energy Strategy Reviews*, 19(Part 2):63–71, 2018. doi: https://doi.org/10.1016/j.esr.2017.12.002.

[61] Mathias Ammon. "tessif-phd v0.0.1 Online Documentation", Mar 2023. [Online]. Available from: https://tessif-phd.readthedocs.io/en/latest/. (Accessed: 2023-04-23).

[62] Mathias Ammon, Max Reimer, Andreas Jessen, Darwin Schnute, Tim Hanke, Cedric Körber, Arne Bo Wilckens, Frederik Emmel, Marc Lüdcke, Leon Zickert, and Fabian Lauster. "tessif-phd: Version 0.0.1", March 2023. Zenodo. doi: https://doi.org/10.5281/zenodo.7771727. Files licensed under the MIT license: https://opensource.org/licenses/MIT.

[63] Max Reimer. "entwicklung eines Komponenten basierten Szenarios zum Vergleich von Free and Open Source Energiesystemmodellierungssoftware in Python". Project Thesis, Technische Universität Hamburg, 2022. doi: https://doi.org/10.15480/882.4160.

[64] Tim Jonas Hanke. "Entwickeln eines netzbasierten Energiesystemmodells zum Vergleich von Free und Open Source Energiesystemmodellierungssoftware in Python". Project Thesis, Technische Universität Hamburg, 2022. doi: https://doi.org/10.15480/882.4185.

[65] Frauke Wiese, Simon Hilpert, Cord Kaldemeyer, and Guido Pleßmann. "A qualitative evaluation approach for energy system modelling frameworks". *Energy, Sustainability and Society*, 8(1):1–16, 2018. doi: https://doi.org/10.1186/s13705-018-0154-3.

[66] Cedric Körber. "Entwicklung eines methodischen Datenblatt generierenden Vergleichs für Free Open Source Energiesystem-Simulationssoftware anhand eines Fallbeispiels mit den Software-Tools Oemof und PyPSA". Bachelor Thesis, Technische Universität Hamburg, 2021. doi: https://doi.org/10.15480/882.4249.

[67] Leander Kotzur, Peter Markewitz, Martin Robinius, and Detlef Stolten. "Impact of different time series aggregation methods on optimal energy system design". *Renewable Energy*, 117:474–487, 2018. doi: https://doi.org/10.1016/j.renene.2017.10.017.

[68] Timo Kannengießer, Maximilian Hoffmann, Leander Kotzur, Peter Stenzel, Fabian Schuetz, Klaus Peters, Stefan Nykamp, Detlef Stolten, and Martin Robinius. "Reducing Computational Load for Mixed Integer Linear Programming: An Example for a District and an Island Energy System". *Energies*, 12(14), 2019. doi: https://doi.org/10.3390/en12142825.

[69] Maximilian Hoffmann, Leander Kotzur, Detlef Stolten, and Martin Robinius. "A Review on Time Series Aggregation Methods for Energy System Models". *Energies*, 13(3), 2020. doi: https://doi.org/10.3390/en13030641.

[70] Max Reimer. "Vergleich von Modellierungsprogrammen zur Optimierung von Energiesystemen durch Integration in ein bestehendes Framework zur Transformation von Energiesystem-Modellen". Master Thesis, Technische Universität Hamburg, 2022. doi: https://doi.org/10.15480/882.4617.

[71] Marc Lüdcke. "Entwicklung einer Methodik zur Validierung der Plausibilität von Free and Open Source Energiesystemmodellierungs- und Optimierungssoftware anhand von Tessif in Python", 2022. doi: https://doi.org/10.15480/882.8003.

[72] Stefan Pfenninger and Bryn Pickering. "Calliope: a multi-scale energy systems modelling framework". *Journal of Open Source Software*, 3(29):825, 2018. doi: https://doi.org/10.21105/joss.00825.

[73] Stefan Pfenninger. "Calliope", Dec 2022. [Online] Available from: https://www.callio.pe/. (Accessed: 2023-04-23).

[74] Calliope Contributors. "Calliope – Stable Version – Online Documentation", Jan 2023. [Online]. Available from: https://calliope.readthedocs.io/en/stable/. (Accessed: 2023-04-23).

[75] Stefan Pfenninger, Bryn Pickering, Tim Tröndle, Francesco Lombardi, Suvayu Ali, Adriaan Hilbers, Graeme Hawker, Katrin Leinweber, Martial Garchery, brmanuel, and smorgenthaler. "calliope-project/calliope: Release v0.6.6-post1", October 2020. doi: https://doi.org/10.5281/zenodo.4080987.

[76] Lara Welder, D.Severin Ryberg, Leander Kotzur, Thomas Grube, Martin Robinius, and Detlef Stolten. "Spatio-temporal optimization of a future energy system for power-to-hydrogen applications in Germany". *Energy*, 158:1130–1149, 2018. doi: https://doi.org/10.1016/j.energy.2018.05.059.

[77] Helmholtz-Gemeinschaft Deutscher Forschungszentren. "Energie System 2050", Jan 2022. [Online] Available from: https://www.helmholtz.de/forschung/forschungsbereiche/energie/energie-system-2050/. (Accessed: 2023-04-23).

[78] Helmholtz-Gemeinschaft Deutscher Forschungszentren. "Helmholtz Energy Computing Initiative (HECI): Energie System 2050 (ES 2050)", Nov 2021. [Online] Available from: https://www.helmholtz.de/forschung/forschungsbereiche/energie/energie-system-2050/heci/. (Accessed: 2023-04-23).

[79] FINE Developer Team. "FINE Stable Version – Online Documentation", Dec 2022. [Online] Available from: https://vsa-fine.readthedocs.io/en/stable/. (Accessed: 2023-04-23).

[80] TSAM Developer Group. "tsam: Time Series Aggregation Module".

[81] FINE Developer Team. "FINE Stable-Version – Online Documentation – Aggregations", Dec 2022. [Online] Available from: https://vsa-fine.readthedocs.io/en/stable/sourceCodeDocumentation/aggregationsDoc.html. (Accessed: 2023-04-23).

[82] FINE Developer Team. "FINE Stable-Version – Online Documentation – aggregateTemporrally", Dec 2022. [Online] Available from: https://vsa-fine.readthedocs.io/en/stable/sourceCodeDocumentation/energySystemModelDoc.html#energySystemModel.EnergySystemModel.aggregateTemporally. (Accessed: 2023-04-23).

[83] FINE Developer Team. "FINE Stable-Version – Online Documentation – Myopic Transformation Pathways", Dec 2022. [Online] Available from: https://vsa-fine.readthedocs.io/en/stable/sourceCodeDocumentation/expansionModules/transformationPathDoc.html. (Accessed: 2023-04-23).

[84] FINE Developer Team. "FINE: Version 2.2.1", Jan 2023. [Online] Available from: https://github.com/FZJ-IEK3-VSA/FINE/commit/59e7c593a3b471f2a4eaffe6c4be758d3417f2da. (Accessed: 2023-04-23).

[85] S. Hilpert, C. Kaldemeyer, U. Krien, S. Günther, C. Wingenbach, and G. Plessmann. "The Open Energy Modelling Framework (oemof) - A new approach to facilitate open science in energy system modelling". *Energy Strategy Reviews*, 22:16–25, 2018. doi: https://doi.org/10.1016/j.esr.2018.07.001.

[86] Uwe Krien, Patrik Schönfeldt, Jann Launer, Simon Hilpert, Cord Kaldemeyer, and Guido Pleßmann. "oemof.solph—A model generator for linear and mixed-integer linear optimisation of energy systems". *Software Impacts*, 6:100028, 2020. doi: https://doi.org/10.1016/j.simpa.2020.100028.

[87] Uwe Krien, Birgit Schachler, and Berit Müller. "Strom, Wärme und Mobilität". [Online] Available from: https://www.strommarkttreffen.org/2017-05-04_6_Uwe_Krien-oemof.pdf. (Accessed: 2023-04-23).

[88] oemof community. "oemof - A modular open source framework to model energy supply systems", Dec 2022. [Online] Available from: https://oemof.org/. (Accessed: 2023-04-23).

[89] oemof-developer group. "oemof.solph – Stable Version – Online Documentation". Dec 2021. [Online] Available from: https://oemof-solph.readthedocs.io/en/stable/. (Accessed: 2023-04-23).

[90] Uwe Krien, Cord Kaldemeyer, Stephan Günther, smn, Patrik Schönfeldt, jnnr, Caroline Möller, Johannes Röder, gplssm, steffenGit, henhuy, Birgit Schachler, cwbach, FranziPl, Julian Endres, FabianTU, Johannes Kochems, pkassing, Bryan Lancien, c koehl, nesnoj, lmb, mloenneberga, Jens-Olaf Delfs, RD-OTH, Francesco Witte, escalacjo, stefansc1, elisapap, and Tjark Smalla. "oemof/oemof-solph: Capable Custom Components", June 2021. doi: https://doi.org/10.5281/zenodo.4896226.

[91] Tom Brown, Jonas Hörsch, and David Schlachtberger. "PyPSA: Python for power system analysis". *Journal of Open Research Software*, 6, 07 2017. doi: https://doi.org/10.5334/jors.188.

[92] Jonas Hörsch, Fabian Hofmann, David Schlachtberger, and Tom Brown. "PyPSA-Eur: An open optimisation model of the European transmission system". *Energy Strategy Reviews*, 22:207–215, 2018. doi: https://doi.org/10.1016/j.esr.2018.08.012.

[93] PyPSA Developers. "PyPSA – v0.19.3 – Online Documentation", Oct 2022. [Online] Available from: https://pypsa.readthedocs.io/en/v0.19.3/. (Accessed: 2023-04-23).

[94] Tom Brown, Jonas Hörsch, Fabian Hofmann, Fabian Neumann, Lisa Zeyen, Chloe Syranidis, Martha Frysztacki, David Schlachtberger, and Philipp Glaum. "PyPSA: Python for Power System Analysis", October 2022. doi: https://doi.org/10.5281/zenodo.7152823.

[95] Mathias Ammon. "Tessif-Examples – Stable Version – Online Documentation". Mar 2023.

[Online] Available from: https://tessif-examples.readthedocs.io/en/latest/. (Accessed: 2023-04-23).

[96] Frederik Emmel. "Developing a method for comparing free open source energy supply system modelling software in the context of computational complexity". Bachelor Thesis, 2022. doi: https://doi.org/10.15480/882.4726.

[97] Andreas Jessen. "Developing a Method for Estimating the Computational Resources Required for Optimising Energy Supply System Models When Using Different Software Tools Through a Framework for Input Data Harmonisation". Bachelor Thesis, 2023. doi: https://doi.org/10.15480/882.5078.

[98] Hannes Hugo Heinrich Schoeneich. "Vergleich der Möglichkeiten zur Modellierung von elektrischen Übertragungsnetz-Komponenten innerhalb eines bestehenden Frameworks zur Transformation von Energiesystem-Modellen". Bachelor Thesis, Technische Universität Hamburg, 2022. doi: https://doi.org/10.15480/882.4725.

[99] Leon Zickert. "Entwicklung eines klimaneutralen Fernwärmenetz-Modells unter Einsatz von Open-Source-Optimierungssoftware mittels eines Datenharmonisierungsframeworks". Project Thesis, 2023. doi: https://doi.org/10.15480/882.6116.

[100] Tim Jonas Hanke. "Integration von Free and Open Source Software zur Optimierung von Energiesystemen in ein bestehendes Framework von Energiesystem Modellen". Master Thesis, Technische Universität Hamburg, 2023. doi: https://doi.org/10.15480/882.4877.

[101] Tim Tröndle, Stefan Pfenninger, and Johan Lilliestam. Home-made or imported: On the possibility for renewable electricity autarky on all scales in europe. *Energy Strategy Reviews*, 26:100388, 2019. doi: https://doi.org/10.1016/j.esr.2019.100388.

[102] Noah Pflugradt, Peter Stenzel, Leander Kotzur, and Detlef Stolten. "LoadProfileGenerator: An Agent-Based Behavior Simulation for Generating Residential Load Profiles". *Journal of Open Source Software*, 7(71):3574, 2022. doi: https://doi.org/10.21105/joss.03574.

[103] Uwe Krien, Patrik Schönfeldt, gplssm, jnnr, Birgit Schachler, Caroline Möller, Pyosch, Stephen Bosch, and henhuy. "oemof/demandlib: Famous Future", January 2021. doi: https://doi.org/10.5281/zenodo.4473045.

[104] Uwe Krien, Guido Plessmann, Stephan Günther, Birgit Schachler, Stephen Bosch, and Cord Kaldemeyer. "feedinlib (oemof) - creating feed-in time series - v0.0.12". January 2019. doi: https://doi.org/10.5281/zenodo.2554102.

[105] Fabian Hofmann, Johannes Hampp, Fabian Neumann, Tom Brown, and Jonas Hörsch. "atlite: A Lightweight Python Package for Calculating Renewable Power Potentials and Time Series", June 2021. doi: https://doi.org/10.5281/zenodo.7525202.

[106] Mathias Ammon. "tessif: Version 0.0.28", April 2023. Zenodo. doi: https://doi.org/10.5281/zenodo.7856507.

[107] Calliope Contributors. "Time Resolution Adjustment", Calliope v0.6.6 Online Documentation. Jan 2023. [Online] Available from: https://calliope.readthedocs.io/en/v0.6.6/user/advanced_features.html#time-resolution-adjustment. (Accessed: 2023-04-23).

[108] PyPSA Developers. "Rolling Horizon Example", PyPSA v0.19.3 Online Documentation. Dec 2021. [Online] Available from: https://pypsa.readthedocs.io/en/v0.19.3/examples/unit-commitment.html#Rolling-horizon-example. (Accessed: 2023-04-23).

[109] Mathias Ammon and Max Reimer. "tessif-calliope-0-6-6post1: Version 0.1.2", April 2023. Zenodo. doi: https://doi.org/10.5281/zenodo.7846380.

[110] Mathias Ammon and Darwin Schnute. "tessif-fine-2-2-2: Version 0.1.3", April 2023. Zenodo. doi: https://doi.org/10.5281/zenodo.7846380.

[111] Mathias Ammon. "tessif-oemof-4-4: Version 0.1.10", April 2023. Zenodo. doi: https://doi.org/10.5281/zenodo.7846175.

[112] Mathias Ammon. "tessif-pypsa-0-19-3: Version 0.1.1", April 2023. Zenodo. doi: https://doi.org/10.5281/zenodo.7846497.

[113] Mathias Ammon. "Tessif Online Documentation", Mar 2023. [Online]. Available from: https://tessif.readthedocs.io/en/latest/. (Accessed: 2023-04-23).

[114] Johannes Dorfner, Konrad Schönleber, Magdalena Dorfner, sonercandas, froehlie, smuellr, dogauzrek, WYAUDI, Leonhard-B, lodersky, yunusozsahin, adeeljsid, Thomas Zipperle, Simon Herzog, kais siala, and Okan Akca. "tum-ens/urbs: urbs v1.0.1", July 2019. Zenodo. doi: https://doi.org/10.5281/zenodo.3265960.

[115] Dennis Atabay. "An open-source model for optimal design and operation of industrial energy systems". *Energy*, 121:803–821, 2017. doi: https://doi.org/10.1016/j.energy.2017.01.030.

[116] Mark Howells, Holger Rogner, Neil Strachan, Charles Heaps, Hillard Huntington, Socrates Kypreos, Alison Hughes, Semida Silveira, Joe DeCarolis, Morgan Bazillian, and Alexander Roehrl. "OSeMOSYS: The Open Source Energy Modeling System: An introduction to its ethos, structure and development". *Energy Policy*, 39(10):5850–5870, 2011. doi: https://doi.org/10.1016/j.enpol.2011.06.033.

[117] Josiah Johnston, Rodrigo Henriquez-Auba, Benjamín Maluenda, and Matthias Fripp. "Switch 2.0: A modern platform for planning high-renewable power systems". *SoftwareX*, 10(100251):100251, 2019.

[118] Ben Elliston, Iain MacGill, and Mark Diesendorf. "Least cost 100% renewable electricity scenarios in the Australian National Electricity Market". *Energy Policy*, 59:270–282, 2013. doi: https://doi.org/10.1016/j.enpol.2013.03.038.

A. Appendix

A.1. Underlying Parameter Assumptions for the Component Commitment and Component Expansion (CompCnE) MSC

Table A.1.: CompCnE Parameter Assumptions Adapted from [63].

#	Component	OPEX €/MWh	CAPEX €/kW	Installed Capacity	Specific Emissions $t_{CO2\text{-}eq}$/MWh	Efficiency
1	Solar Panel	80	1000	1100 MW	0.05	-
2	Onshore Wind Turbine	60	1750	1100 MW	0.02	-
3	Offshore Wind Turbine	105	3900	150 MW	0.02	-
4	Biogas CHP_{el}	150	3500	200 MW	0.25	0.4
5	Biogas CHP_{th}	11.25	262.5	250 MW	0.01875	0.5
6	Lignite Power Plant	65	1900	500 MW	1	0.4
7	Hard Coal CHP_{el}	80	1750	300 MW	0.8	0.4
8	Hard Coal CHP_{th}	6	131.25	300 MW	0.06	0.4
9	Hard Coal PP	80	1650	500 MW	0.8	0.43
10	Combined Cycle PP	90	950	600 MW	0.35	0.6
11	Heat Plant	35	390	450 MW	0.23	0.9
12	Power To Heat	20	100	100 MW	0.0007	0.99
13	Battery	400	1630	100 MWh	0.06	0.9
14	Heat Storage	20	4.5	50 MWh	0	0.85

A.2. Underlying Parameter Assumptions for the Transformer Commitment and Transformer Expansion (TransCnE) MSC

Table A.2.: TransCnE — Parameter Assumptions; Taken from [64].

Component	OPEX €/MWh	Installed Capacity MW or kW	Specific Emissions $t_{CO2\text{-}eq}$/MWh	Efficiency
PV	60.85	100000	-	-
Cogen. (Power)	124.4	8576	0.1576	0.33
Cogen. (Heat)	31.1	13513	0.0732	0.52
Onshore Wind	61.1	71000	-	-
Offshore Wind	106.4	20000	-	-
Hard Coal Power Plant	80.65	43913	0.5136	0.43
Hard Coal CHP (Power)	80.65	24509.6	0.5136	0.24
Hard Coal CHP (Heat)	20.16	61273.96	0.293	0.60
Combined Cycle Power Plant	88.7	26742	0.3366	0.59
Solar Thermal	73	14400	-	-

A.3. Factsheet-Based Comparison Tables of Preselection-Step 3

Institution Abbreviations	DLR – Deutsche Luft- und Raumfahrtbehörde ETHZ - Swiss Federal Institute of Technology in Zurich EUF – Europa Universität Flensburg, FIAS - Frankfurt Institute for Advanced Studies, HSF - Hochschule Flensburg, JRC - Jülich Research Center KIT - Karlsruhe Institute of Technology, OvGUM – Otto-von-Guericke-Universität Magdeburg, RLI - Reiner Lemoine Institut, TUB- Technical University of Berlin, TUM - Technical University of Munich
General Goal Abbreviations	P – Prediction R - Research RT – Retransfer
Specific Goal Abbreviations	EC – Energy Consumption ES – Energy Supply EV – Evaluation I - Impact
Temporal Coverage Abbreviations	LT – Long Term, ST – Short Term,
Sector Coverage Abbreviations	UD – User Defined,

Operating System Abbreviations	LIN - Linux Operating System MAC - Mac Operating System WIN - Windows Operating System
Input Data Abbreviations	nCDF - netCDF PaP - pandapower PyC - Python Code PyP - pypower
Input Parameterisation Abbreviations	BD – Broken Down, MON – Monetary QL – Qualitatively QN – Quantitatively
Analytic Approach Abbreviations	HYB – Hybrid, BU – Bottom-Up, TD – Top-Down,
Method of Analysis Abbreviations	Opt - Optimization Sim - Simulation
Problem Formulation Archetype Abbreviations	CP - Commitment Problem EP - Expansion Problem EC - Emission Constrained
Power Flow Analysis Abbreviations	LOPF – Linear Optimal Power Flow, OPF – Optimal Power Flow,
Sector Abbreviations	HS – Heat Sector, MS – Mobility Sector, PS – Power Sector,
Component Template Abbreviations	ELine - Electrical Line ExCHP - Extraction Turbine CHP Gen - Generator GenStor - Generic Storage GenCAES - Generic Compressed Air Energy Storage OffTran - Offset Transformer SinkDSM - Sink Demand Side Management SImp - Shunt Impedance Stor - Storage SU - Storage Unit Tran - Transformer

Table A.4.: Factsheet — Part One.

Category	Parameter	*Calliope*	*FINE*	*oemof*	*PyPSA*
Basic Information	Developer	ETHZ	JRC	RLI, HSF, DLR, EUF	TUB, KIT, FIAS
	Published	2013	2018	2015	2015
	General Goal	P, R, RT	P, R, RT	P, R, RT	P, R, RT
	Specific Goal	EC, ES, EV, I	EC, ES, EV, I	EC, ES, EV, I	EC, ES, EV, I
	Spatial Coverage	UD	UD	UD	UD
	Temporal Coverage	ST	ST, LT	ST	ST, (LT)
	Sector Coverage	UD	UD	UD	Focus on Power, UD
	Version	0.6.8	2.2.2	0.4.4	0.19.1
	Last Update	2021-09-10	2022-02-15	2021-12-21	2022-02-18
	Free Online Doc	Yes	Yes	Yes	Yes
Programmatic Formulation	FOSS	Yes	Yes	Yes	Yes
	Supported OS	LIN, MAC, WIN	LIN, MAC, WIN	LIN, MAC, WIN	LIN, MAC, WIN
	Progr. Language	Python	Python	Python	Python
	Object Oriented	Yes	Yes	Yes	Yes
	Solver Interfaces	Pyomo	Pyomo	Pyomo	Pyomo; Own
	Solvers	GLPK, CBC, Gurobi, CPLEX	GLPK, CBC Gurobi, CPLEX	GLPK, CBC Gurobi, CPLEX	GLPK, CBC Gurobi, CPLEX
	Input Data	YAML & Python Code	Python Code	Python Code	Python Code CSV, HDF5, nCDF PaP, PyP
	Output Data	Python Code	Python Code	Python Code	Python Code CSV, HDF5, nCDF PaP, PyP

Table A.5.: Factsheet — Part Two.

Category	Parameter	*Calliope*	*FINE*	*oemof*	*PyPSA*
Mathematical Description	Input-Parameterisation	BD, (MON), QL, QN	BD, MON, QL, QN	BD, MON, QL, QN	BD, (MON), QL, QN
	Analytic Approach	BU, HYB, TD	BU, HYB, TD	BU, HYB, TD	BU
	Method of Analysis	Opt,	Opt	Opt, Sim	Opt
	Optimisation Problem Formulation	LP,	LP, MILP, NLP	LP, MILP	LP, MILP, NLP
	Scenario Formulation	CP, EC. EP	CP, EC. EP	CP, EC. EP	CP, EC. EP
	Power Flow Model	-	-	LOPF	LOPF, OPF
	Additional Energy Flow Models	No	No	No	No
Modelling Approach	Supported Sectors	HS, MS, PS	HS, MS, PS	HS, MS, PS	HS, MS, PS
	Temporal Resolution	UD	UD	UD	UD
	Spatial Resolution	UD	UD	UD	UD
	Endogenous Factors	Low to Medium	Low to Medium	Low to Medium	Low to Medium
	Non-Energy-Sectors	No	No	No	No
	Consumer Description		No/Low	Yes/Medium	No/Low
	Energy Transporation	Yes/Medium	Yes/Medium	Yes/High	Yes/Detailed
	Electric Grid Model	No	No	Yes/rudimentary	Yes/Detailed
	Modular Design	Yes	Yes	Yes	Yes
	Native Components	Conversion/-plus Demand Storage Supply/-plus Transmission	Conversion, Sink Source Transmission Transmission	Bus, ELine, ExCHP, GenStor, GenCAES, OffTran, Sink, SiDSM Source, Storage, Tran	Bus, Gen, Line Link, Load, SImp, Storage SU, Tran

Table A.6.: Factsheet — Part Three.

Category	Parameter	*Calliope*	*FINE*	*oemof*	*PyPSA*
Modelling Approach	Component Detail		High	High	High
	Exogenous Parameters		High	High	High
	Extensible/Customizable	Yes	Yes	Yes	Yes
Renewable Energies	designed for renewable integration	Yes	Yes	Yes	Yes
	supply profile generation	Yes (*solar-and-wind-potentials*)	No	Yes (*feedinlib*)	Yes (*atlite*)
	demand profile generation	No	Yes (*LoadProfile Generator*)	Yes (*demandlib*)	No
	modelling renewable Components	Supply Profile + Supply Comp.	Supply Profile + Source Comp.	Supply Profile + Source Comp.	Supply Profile + Generator Comp.
	type of emissions	User Defined	User defined	User defined	User defined
Collaborative Properties	Availability	FOSS	FOSS	FOSS	FOSS
	Open Data	Yes	Yes	Yes	Yes
	Licence	Apache 2.0	MIT	MIT	MIT
	Terminological Uniformity	Medium	High	Very High	High
	Scientific Standards	Yes	Yes	Yes	Yes
	Quality Examples	Yes	Yes	Yes	Yes
	Developer Support	Yes/Medium	Yes/High	Yes/High	Yes/High

A.4. Component-Commitment (CompC) MSC Results Used In the Case Study

Table A.7.: CompC — High-Priority Content-Results Relative to *oemof*.

Result	*Calliope*	*FINE*	*oemof*	*PyPSA*
total costs	1.00	1.00	1.00	1.00
OPEX	1.00	1.00	1.00	1.00
CAPEX	1.00	1.00	1.00	1.00
emissions	1.00	0.96	1.00	1.00

Table A.8.: CompC — Summed-Up Loads Results Relative to *oemof*.

Flow Source	Flow Target	*Calliope*	*FINE*	*oemof*	*PyPSA*
Solar Panel	Powerline	1.13	1.00	1.0	0.92
Hard Coal PP	Powerline	0.98	1.00	1.0	1.01
Hard Coal Supply Line	Hard Coal PP	0.98	1.00	1.0	0.00
Hard Coal Supply	Hard Coal Supply Line	0.99	1.00	1.0	0.37
Heat Storage	Heatline	1.00	0.99	1.0	1.00

A.5. Component-Expansion (CompE) MSC Results Used In the Case Study

Table A.9.: CompE — High-Priority Content-Results Relative to *oemof*.

Result	*Calliope*	*FINE*	*oemof*	*PyPSA*
total costs	1.00	1.00	1.00	0.05
OPEX	1.00	1.00	1.00	0.03
CAPEX	1.00	1.00	1.00	1.19
emissions	1.00	0.96	1.00	20.1

Table A.10.: CompE — Summed-Up Loads Results in MWh.

Flow Source	Flow Target	*Calliope*	*FINE*	*oemof*	*PyPSA*
Battery	Powerline	59185	59143	59143	16594
Biogas CHP	Heatline	83466	83088	83088	1126470
Biogas CHP	Powerline	66772	66471	66471	901176
Biogas Line	Biogas CHP	166931	166177	166177	2252941
Biogas Supply	Biogas Line	166931	166177	166177	2252941
Combined Cycle PP	Powerline	11270	11513	11513	460929
Gas Line	Combined Cycle PP	18783	19188	19188	0
Gas Station	Gas Line	18783	19188	19188	0
Hard Coal CHP	Heatline	0	0	0	5252800
Hard Coal CHP	Powerline	0	0	0	5252800
Hard Coal Supply	Hard Coal Supply Line	0	0	0	13132000
Hard Coal Supply Line	Hard Coal CHP	0	0	0	13132000
Heat Storage	Heatline	110220	109928	109928	1
Heatline	Heat Demand	1116163	1116163	1116163	1116163
Heatline	Heat Storage	136673	136245	136245	5263109
Offshore Wind Turbine	Powerline	480373	479668	479668	412889
Onshore Wind Turbine	Powerline	10089237	10090425	10090425	1959678
Power To Heat	Heatline	1059151	1059391	1059391	0
Powerline	Battery	69736	69681	69681	19032
Powerline	El Demand	9809506	9809506	9809506	9809506
Powerline	Power To Heat	1069849	1070092	1070092	0
Solar Panel	Powerline	242254	242061	242061	824472

Table A.11.: CompE — Installed Capacity Results in MW or MWh.

Component	*Calliope*	*FINE*	*oemof*	*PyPSA*
Battery	2105	2105	2105	100
Biogas CHP Heatline	650	645	645	250
Biogas CHP Powerline	520	516	516	200
Biogas Supply	1299	1291	1291	500
Combined Cycle PP	600	600	600	600
El Demand	1526	1526	1526	1526
Gas Station	847	853	853	0
Hard Coal CHP Heatline	300	300	300	631
Hard Coal CHP Powerline	300	300	300	631
Hard Coal PP	500	500	500	500
Hard Coal Supply	0	0	0	1577
Heat Demand	400	400	400	400
Heat Plant	450	450	450	450
Heat Storage	11358	11273	11273	113392
Lignite Power Plant	500	500	500	500
Offshore Wind Turbine	2347	2346	2346	150
Onshore Wind Turbine	15776	15788	15788	1100
Power To Heat	576	576	576	100
Solar Panel	3812	3810	3810	1100

Table A.12.: CompE — Emissions Caused Results in Tons CO_2 Equivalent.

Flow Source	Flow Target	*Calliope*	*FINE*	*oemof*	*PyPSA*
Battery	Powerline	3551	3549	3549	996
Biogas CHP	Heatline	0	1558	1558	21121
	Powerline	18258	16618	16618	225294
Combined Cycle PP	Powerline	3945	4029	4029	161325
Hard Coal CHP	Heatline	0	0	0	315168
	Powerline	0	0	0	4202240
Offshore Wind Turbine	Powerline	9607	9593	9593	8258
Onshore Wind Turbine	Powerline	201785	201808	201808	39194
Power To Heat	Heatline	741	742	742	0
Solar Panel	Powerline	12113	12103	12103	41224

Table A.13.: CompE — Summed-Up Loads Interest Level Results.

Flow Source	Flow Target	*Calliope*	*FINE*	*oemof*	*PyPSA*
Battery	Powerline	low	low	low	high
Biogas CHP	Heatline	low	low	low	high
Biogas CHP	Powerline	low	low	low	high
Biogas Line	Biogas CHP	low	low	low	high
Biogas Supply	Biogas Line	low	low	low	high
Combined Cycle PP	Powerline	low	low	low	high
Hard Coal CHP	Heatline	None	None	None	high
Hard Coal CHP	Powerline	None	None	None	high
Hard Coal PP	Powerline	None	None	None	None
Hard Coal Supply	Hard Coal Supply Line	None	None	None	high
Hard Coal Supply Line	Hard Coal CHP	None	None	None	high
Heat Plant	Heatline	None	None	None	None
Heat Storage	Heatline	low	low	low	high
Heatline	Heat Demand	low	low	low	low
Heatline	Heat Storage	low	low	low	high
Lignite Power Plant	Powerline	None	None	None	None
Offshore Wind Turbine	Powerline	low	low	low	high
Onshore Wind Turbine	Powerline	low	low	low	high
Power To Heat	Heatline	low	low	low	high
Powerline	Battery	low	low	low	high
Powerline	El Demand	low	low	low	low
Powerline	Power To Heat	low	low	low	high
Solar Panel	Powerline	low	low	low	high

Table A.14.: CompE — Summed-Up Loads Correlation Coefficient Results.

Flow Source	Flow Target	*Calliope*	*FINE*	*oemof*	*PyPSA*
Battery	Powerline	1.00	1.00	1.00	0.03
Biogas CHP	Heatline	1.00	1.00	1.00	0.14
	Powerline	1.00	1.00	1.00	0.14
Biogas Line	Biogas CHP	1.00	1.00	1.00	0.14
Biogas Supply	Biogas Line	1.00	1.00	1.00	0.14
Combined Cycle PP	Powerline	1.00	1.00	1.00	0.23
Hard Coal CHP	Heatline	0.00	0.00	0.00	0.00
	Powerline	0.00	0.00	0.00	0.00
Hard Coal PP	Powerline	0.00	0.00	0.00	0.00
Hard Coal Supply	Hard Coal Supply Line	0.00	0.00	0.00	0.00
Hard Coal Supply Line	Hard Coal CHP	0.00	0.00	0.00	0.00
Heat Plant	Heatline	0.00	0.00	0.00	0.00
Heat Storage	Heatline	1.00	1.00	1.00	-0.00
Heatline	Heat Demand	1.00	1.00	1.00	1.00
	Heat Storage	1.00	1.00	1.00	0.10
Lignite Power Plant	Powerline	0.00	0.00	0.00	0.00
Offshore Wind Turbine	Powerline	1.00	1.00	1.00	-0.16
Onshore Wind Turbine	Powerline	1.00	1.00	1.00	0.52
Power To Heat	Heatline	1.00	1.00	1.00	0.00
Powerline	Battery	1.00	1.00	1.00	-0.01
	El Demand	1.00	1.00	1.00	1.00
	Power To Heat	1.00	1.00	1.00	0.00
Solar Panel	Powerline	1.00	1.00	1.00	0.35

Table A.15.: CompE — Summed-Up Loads Error Value Results.

Flow Source	Flow Target	*Calliope*	*FINE*	*oemof*	*PyPSA*
Battery	Powerline	0.01	0.00	0.00	1.26
Biogas CHP	Heatline	0.00	0.00	0.00	13.54
	Powerline	0.00	0.00	0.00	13.54
Biogas Line	Biogas CHP	0.00	0.00	0.00	13.54
Biogas Supply	Biogas Line	0.00	0.00	0.00	13.54
Combined Cycle PP	Powerline	0.02	0.00	0.00	39.15
Hard Coal CHP	Heatline	None	None	None	inf
	Powerline	None	None	None	inf
Hard Coal PP	Powerline	None	None	None	None
Hard Coal Supply	Hard Coal Supply Line	None	None	None	inf
Hard Coal Supply Line	Hard Coal CHP	None	None	None	inf
Heat Plant	Heatline	None	None	None	None
Heat Storage	Heatline	0.00	0.00	0.00	1.00
Heatline	Heat Demand	0.00	0.00	0.00	0.00
	Heat Storage	0.00	0.00	0.00	37.63
Lignite Power Plant	Powerline	None	None	None	None
Offshore Wind Turbine	Powerline	0.00	0.00	0.00	1.68
Onshore Wind Turbine	Powerline	0.00	0.00	0.00	0.81
Power To Heat	Heatline	0.00	0.00	0.00	1.00
Powerline	Battery	0.00	0.00	0.00	1.26
	El Demand	0.00	0.00	0.00	0.00
	Power To Heat	0.00	0.00	0.00	1.00
Solar Panel	Powerline	0.00	0.00	0.00	3.22

Table A.16.: CompE — Installed Capacities Interest Level Results.

Component	*Calliope*	*FINE*	*oemof*	*PyPSA*
Battery	low	low	low	high
Biogas CHP Heatline	low	low	low	high
Biogas CHP Powerline	low	low	low	high
Biogas Supply	low	low	low	high
Combined Cycle PP	low	low	low	low
El Demand	low	low	low	low
Gas Station	low	low	low	None
Hard Coal CHP Heatline	low	low	low	high
Hard Coal CHP Powerline	low	low	low	high
Hard Coal PP	low	low	low	low
Hard Coal Supply	None	None	None	high
Heat Demand	low	low	low	low
Heat Plant	low	low	low	low
Heat Storage	low	low	low	high
Lignite Power Plant	low	low	low	low
Lignite Supply	None	None	None	None
Offshore Wind Turbine	low	low	low	high
Onshore Wind Turbine	low	low	low	high
Power To Heat	low	low	low	high
Solar Panel	low	low	low	high

Table A.17.: CompE — Installed Capacities Relative Deviation Results.

Component	*Calliope*	*FINE*	*oemof*	*PyPSA*
Battery	0.00	0.00	0.00	0.95
Biogas CHP Heatline	0.01	0.00	0.00	0.61
Biogas CHP Powerline	0.01	0.00	0.00	0.61
Biogas Supply	0.01	0.00	0.00	0.61
Combined Cycle PP	0.00	0.00	0.00	0.00
El Demand	0.00	0.00	0.00	0.00
Gas Station	0.01	0.00	0.00	None
Hard Coal CHP Heatline	0.00	0.00	0.00	1.1
Hard Coal CHP Powerline	0.00	0.00	0.00	1.1
Hard Coal PP	0.00	0.00	0.00	0.00
Hard Coal Supply	None	None	None	inf
Heat Demand	0.00	0.00	0.00	0.00
Heat Plant	0.00	0.00	0.00	0.00
Heat Storage	0.01	0.00	0.00	9.06
Lignite Power Plant	0.00	0.00	0.00	0.00
Lignite Supply	None	None	None	None
Offshore Wind Turbine	0.00	0.00	0.00	0.94
Onshore Wind Turbine	0.00	0.00	0.00	0.93
Power To Heat	0.00	0.00	0.00	0.83
Solar Panel	0.00	0.00	0.00	0.71

A.6. Lossless-Commitment (LossLC) MSC Results Used in The Case Study

Table A.18.: LossLC — High-Priority Content-Results Relative to *oemof*.

Category	Result	*Calliope*	*FINE*	*oemof*	*PyPSA*
Content	total costs	1.00	1.00	1.00	1.00
	OPEX	1.00	1.00	1.00	1.00
	CAPEX	1.00	1.00	1.00	1.00
	emissions	1.00	0.99	1.00	1.00

Table A.19.: Selected LossLC — Summed-Up Loads Results Relative to *oemof*.

Flow Source	Flow Target	*Calliope*	*FINE*	*oemof*	*PyPSA*
Combined Cycle Power Plant	High Voltage Grid	0.96	1.00	1.00	1.00
Gas Supply Line	Combined Cycle Power Plant	0.96	1.00	1.00	0.0
Gas	Gas Supply Line	0.96	1.00	1.00	0.0
Biogas	Biogas Supply Line	1.41	1.00	1.00	1.00
Biogas Supply Line	Cogeneration	1.41	1.00	1.00	1.00
Cogeneration	Low Voltage Grid	1.41	1.00	1.00	1.00
Cogeneration	District Heating	1.41	1.00	1.00	1.00
CHP	High Voltage Grid	0.99	1.00	1.00	1.00
CHP	District Heating	0.99	1.00	1.00	1.00
Hard Coal Power Plant	High Voltage Grid	1.02	1.00	1.00	1.00

Table A.20.: Selected LossLC — Summed-Up Loads Result Differences Compared to *oemof*.

Flow Source	Flow Target	*Calliope*	*FINE*	*oemof*	*PyPSA*
Combined Cycle Power Plant	High Voltage Grid	-5457	0	0	0
Cogeneration	Low Voltage Grid	3641	0	0	0
Cogeneration	District Heating	5737	0	0	0
CHP	High Voltage Grid	-2295	0	0	0
CHP	District Heating	-5737	0	0	0
Hard Coal Power Plant	High Voltage Grid	4111	0	0	0

A.7. No-Congestion Transformer Commitment (No-Congestion TransC) Results Used in The Case Study

Table A.21.: No Congestion TransC — High-Priority Results Relative to *oemof*.

Result	*Calliope*	*FINE*	*oemof*	*PyPSA*
total costs	1.00	1.00	1.00	1.00
OPEX	1.00	1.00	1.00	1.00
CAPEX	0	0	0	0
emissions	1.00	1.00	1.00	1.00

A.8. Congestion Transformer Commitment (Congestion TransC) MSC Results Used in the Case Study

Table A.22.: Congestion TransC — High-Priority Results Relative to *oemof*.

Result	*Calliope*	*FINE*	*oemof*	*PyPSA*
total costs	1.00	1.00	1.00	1.00
OPEX	1.00	1.00	1.00	1.00
CAPEX	0	0	0	0
emissions	1.00	0.94	1.00	0.94

Table A.23.: Congestion-TransC — Summed-Up Loads Results in MWh.

Flow Source	Flow Target	*Calliope*	*FINE*	*oemof*	*PyPSA*
Cogeneration	District Heating	218805	218805	218805	218805
Cogeneration	Low Voltage Grid	138857	138857	138857	138857
Biogas Supply Line	Cogeneration	420778	420778	420778	420778
Biogas	Biogas Supply Line	420778	420778	420778	420778
Coal Supply	Coal Supply Line	709038	709038	709038	709038
Hard Coal Supply Line	CHP	652109	652109	652109	652109
Hard Coal Supply Line	Hard Coal Power Plant	56929	56929	56929	56929
Deficit Source LV	Low Voltage Grid	301704	301704	301704	301704
Deficit Source MV	Medium Voltage Grid	31460	31460	31460	31460
District Heating	District Heating Demand	865460	865460	865460	865460
CHP	District Heating	391266	391266	391266	391266
CHP	High Voltage Grid	156506	156506	156506	156506
Hard Coal Power Plant	High Voltage Grid	24479	24479	24479	24479
High Medium Transfer	Medium Voltage Grid	450363	450363	450363	450363
High Voltage Grid	Excess Sink HV	6546	6546	6546	6546
High Voltage Grid	High Medium Transfer	484261	484261	484261	484261
Low Medium Transfer	Medium Voltage Grid	93716	93716	93716	93716
Low Voltage Grid	Commercial Demand	582460	582460	582460	582460
Low Voltage Grid	Excess Sink LV	20777	20777	20777	20777
Low Voltage Grid	Household Demand	552565	552565	552565	552565
Low Voltage Grid	Low Medium Transfer	100770	100770	100770	100770
Medium Low Transfer	Low Voltage Grid	222664	222664	222664	222664
Medium Voltage Grid	Car charging Station	37026	37026	37026	37026
Medium Voltage Grid	Industrial Demand	1229008	1229008	1229008	1229008
Medium Voltage Grid	Medium Low Transfer	239424	239424	239424	239424
Medium Voltage Grid	Power to Heat	169947	169947	169947	169947
Offshore Wind Power	High Voltage Grid	309822	309822	309822	309822
Onshore Wind Power	Medium Voltage Grid	1099866	1099866	1099866	1099866
Power to Heat	District Heating	169947	169947	169947	169947
Solar Panel	Low Voltage Grid	593347	593347	593347	593347
Solar Thermal	District Heating	85442	85442	85442	85442

A.9. Transformer Expansion (TransE) MSC Results Used in the Case Study

Table A.24.: TransE — High-Priority Results Relative to *oemof*.

Result	*Calliope*	*FINE*	*oemof*	*PyPSA*
total costs	1.00	1.00	1.00	1.00
OPEX	1.00	1.00	1.00	1.00
CAPEX	1.00	1.04	1.00	1.00
emissions	1.00	1.00	1.00	1.00

Table A.25.: TransE — Installed Capacity Results of the Expandable Components in MW.

Component	*Calliope*	*FINE*	*oemof*	*PyPSA*
Low Medium Transfer	30433	30433	30433	30433
Medium Low Transfer	53955	53955	53955	53955
High Medium Transfer	69567	69567	69567	69567
Medium High Transfer	1	1	1	1

Table A.26.: TransE — Emissions Caused Results in $Mt_{CO2\text{-}eq}$.

Flow Source	Flow Target	*Calliope*	*FINE*	*oemof*	*PyPSA*
Cogeneration	District Heating	0.005	0.002	0.002	0.002
Cogeneration	Low Voltage Grid	0.000	0.003	0.003	0.003
Deficit Source LV	Low Voltage Grid	0.004	0.004	0.004	0.004
Combined Cycle Power Plant	High Voltage Grid	0.055	0.055	0.055	0.055
CHP	District Heating	0.303	0.178	0.178	0.178
CHP	High Voltage Grid	0.000	0.125	0.125	0.125
Hard Coal Power Plant	High Voltage Grid	0.117	0.117	0.117	0.117

Table A.27.: TransE — Summed-Up Loads Results in MWh.

Flow Source	Flow Target	*Calliope*	*FINE*	*oemof*	*PyPSA*
BHKW	District Heating	30823	30823	30823	30823
	Low Voltage Grid	19561	19561	19561	19561
Biogas	BHKW	59275	59275	59275	59275
Biogas plant	Biogas	59275	59275	59275	59275
Coal Supply	Coal Supply Line	1541802	1541802	1541802	1541802
Coal Supply Line	HKW	1012523	1012523	1012523	1012523
	HKW2	529279	529279	529279	529279
Deficit Source LV	Low Voltage Grid	6910	6910	6910	6910
District Heating	District Heating Demand	865460	865460	865460	865460
Gas Station	Gaspipeline	277364	277364	277364	0
Gaspipeline	GuD	277364	277364	277364	0
GuD	High Voltage Grid	163645	163645	163645	163645
HKW	District Heating	607514	607514	607514	607514
	High Voltage Grid	243006	243006	243006	243006
HKW2	High Voltage Grid	227590	227590	227590	227590
High Medium Transfer	Medium Voltage Grid	877978	877978	877978	877978
High Voltage Grid	High Medium Transfer	944062	944062	944062	944062
Low Medium Transfer	Medium Voltage Grid	103346	103346	103346	103346
Low Voltage Grid	Commercial Demand	582460	582460	582460	582460
	Household Demand	552565	552565	552565	552565
	Low Medium Transfer	111125	111125	111125	111125
Medium Low Transfer	Low Voltage Grid	626332	626332	626332	626332
Medium Voltage Grid	Car charging Station	37026	37026	37026	37026
	Industrial Demand	1229008	1229008	1229008	1229008
	Medium Low Transfer	673475	673475	673475	673475
	Power to Heat	141681	141681	141681	141681
Offshore Wind Power	High Voltage Grid	309822	309822	309822	309822
Onshore Wind Power	Medium Voltage Grid	1099866	1099866	1099866	1099866
Power to Heat	District Heating	141681	141681	141681	141681
Solar Panel	Low Voltage Grid	593347	593347	593347	593347
Solar Thermal	District Heating	85442	85442	85442	85442

A.10. Modified Component-Expansion (Modifed CompE) MSC Results Used In the Case Study

Table A.28.: Modified-CompE — High-Priority Results Relative to *oemof*.

Result	*Calliope*	*FINE*	*oemof*	*PyPSA*
total costs	1.00	1.00	1.00	0.89
OPEX	1.00	1.00	1.00	0.89
CAPEX	1.00	1.00	1.00	1.12
emissions	1.00	1.00	1.00	1.06

Table A.29.: Modified-CompE — Load Profiles Interest Level Results.

Flow Source	Flow Target	*Calliope*	*FINE*	*oemof*	*PyPSA*
Battery	Powerline	low	low	low	high
Biogas CHP	Heatline	low	low	low	medium
Biogas CHP	Powerline	low	low	low	medium
Biogas Line	Biogas CHP	low	low	low	medium
Biogas Supply	Biogas Line	low	low	low	medium
Combined Cycle PP	Powerline	low	low	low	medium
Hard Coal CHP	Heatline	None	None	None	None
Hard Coal CHP	Powerline	None	None	None	None
Hard Coal PP	Powerline	None	None	None	None
Hard Coal Supply	Hard Coal Supply Line	None	None	None	None
Hard Coal Supply Line	Hard Coal CHP	None	None	None	None
Heat Plant	Heatline	None	None	None	None
Heat Storage	Heatline	low	low	low	medium
Heatline	Heat Demand	low	low	low	low
Heatline	Heat Storage	low	low	low	high
Lignite Power Plant	Powerline	None	None	None	None
Offshore Wind Turbine	Powerline	low	low	low	medium
Onshore Wind Turbine	Powerline	low	low	low	low
Power To Heat	Heatline	low	low	low	medium
Powerline	Battery	low	low	low	high
Powerline	El Demand	low	low	low	low
Powerline	Power To Heat	low	low	low	medium
Solar Panel	Powerline	low	low	low	medium

Table A.30.: Modified CompE — Load Profiles Correlation Coefficient Results.

Flow Source	Flow Target	*Calliope*	*FINE*	*oemof*	*PyPSA*
Battery	Powerline	1.0	1.0	1.0	0.59
Biogas CHP	Heatline	1.0	1.0	1.0	0.93
Biogas CHP	Powerline	1.0	1.0	1.0	0.93
Biogas Line	Biogas CHP	1.0	1.0	1.0	0.93
Biogas Supply	Biogas Line	1.0	1.0	1.0	0.93
Combined Cycle PP	Powerline	1.0	1.0	1.0	0.95
Hard Coal CHP	Heatline	0.0	0.0	0.0	0.00
Hard Coal CHP	Powerline	0.0	0.0	0.0	0.00
Hard Coal PP	Powerline	0.0	0.0	0.0	0.00
Hard Coal Supply	Hard Coal Supply Line	0.0	0.0	0.0	0.00
Hard Coal Supply Line	Hard Coal CHP	0.0	0.0	0.0	0.00
Heat Plant	Heatline	0.0	0.0	0.0	0.00
Heat Storage	Heatline	1.0	1.0	1.0	0.80
Heatline	Heat Demand	1.0	1.0	1.0	1.00
Heatline	Heat Storage	1.0	1.0	1.0	0.61
Lignite Power Plant	Powerline	0.0	0.0	0.0	0.00
Offshore Wind Turbine	Powerline	1.0	1.0	1.0	0.73
Onshore Wind Turbine	Powerline	1.0	1.0	1.0	0.93
Power To Heat	Heatline	1.0	1.0	1.0	0.81
Powerline	Battery	1.0	1.0	1.0	0.17
Powerline	El Demand	1.0	1.0	1.0	1.00
Powerline	Power To Heat	1.0	1.0	1.0	0.81
Solar Panel	Powerline	1.0	1.0	1.0	0.86

Table A.31.: Modified CompE — Load Profiles Error Value Results.

Flow Source	Flow Target	*Calliope*	*FINE*	*oemof*	*PyPSA*
Battery	Powerline	0.01	0.00	0.00	3.17
Biogas CHP	Heatline	0.00	0.00	0.00	0.38
Biogas CHP	Powerline	0.00	0.00	0.00	0.38
Biogas Line	Biogas CHP	0.00	0.00	0.00	0.38
Biogas Supply	Biogas Line	0.00	0.00	0.00	0.38
Combined Cycle PP	Powerline	0.02	0.00	0.00	0.6
Hard Coal CHP	Heatline	None	None	None	None
Hard Coal CHP	Powerline	None	None	None	None
Hard Coal PP	Powerline	None	None	None	None
Hard Coal Supply	Hard Coal Supply Line	None	None	None	None
Hard Coal Supply Line	Hard Coal CHP	None	None	None	None
Heat Plant	Heatline	None	None	None	None
Heat Storage	Heatline	0.00	0.00	0.00	0.53
Heatline	Heat Demand	0.00	0.00	0.00	0.00
Heatline	Heat Storage	0.00	0.00	0.00	1.17
Lignite Power Plant	Powerline	None	None	None	None
Offshore Wind Turbine	Powerline	0.00	0.00	0.00	0.77
Onshore Wind Turbine	Powerline	0.00	0.00	0.00	0.05
Power To Heat	Heatline	0.00	0.00	0.00	0.18
Powerline	Battery	0.00	0.00	0.00	4.40
Powerline	El Demand	0.00	0.00	0.00	0.00
Powerline	Power To Heat	0.00	0.00	0.00	0.18
Solar Panel	Powerline	0.00	0.00	0.00	0.51

Table A.32.: Modified-CompE — Installed Capacities Interest Level Results.

Component	*Calliope*	*FINE*	*oemof*	*PyPSA*
Battery	low	low	low	high
Biogas CHP Heatline	low	low	low	medium
Biogas CHP Powerline	low	low	low	medium
Biogas Supply	low	low	low	medium
Combined Cycle PP	low	low	low	low
El Demand	low	low	low	low
Gas Station	low	low	low	None
Hard Coal CHP Heatline	low	low	low	low
Hard Coal CHP Powerline	low	low	low	low
Hard Coal PP	low	low	low	low
Hard Coal Supply	None	None	None	None
Heat Demand	low	low	low	low
Heat Plant	low	low	low	low
Heat Storage	low	low	low	low
Lignite Power Plant	low	low	low	low
Lignite Supply	None	None	None	None
Offshore Wind Turbine	low	low	low	medium
Onshore Wind Turbine	low	low	low	medium
Power To Heat	low	low	low	medium
Solar Panel	low	low	low	low

Table A.33.: Modified-CompE — Installed Capacities Relative Deviation Results.

Component	*Calliope*	*FINE*	*oemof*	*PyPSA*
Battery	0.00	0.00	0.00	0.79
Biogas CHP Heatline	0.01	0.00	0.00	-0.15
Biogas CHP Powerline	0.01	0.00	0.00	-0.15
Biogas Supply	0.01	0.00	0.00	-0.15
Combined Cycle PP	0.00	0.00	0.00	0.00
El Demand	0.00	0.00	0.00	0.00
Gas Station	-0.01	0.00	0.00	None
Hard Coal CHP Heatline	0.00	0.00	0.00	0.00
Hard Coal CHP Powerline	0.00	0.00	0.00	0.00
Hard Coal PP	0.00	0.00	0.00	0.00
Hard Coal Supply	None	None	None	None
Heat Demand	0.00	0.00	0.00	0.00
Heat Plant	0.00	0.00	0.00	0.00
Heat Storage	0.01	0.00	0.00	-0.02
Lignite Power Plant	0.00	0.00	0.00	0.00
Lignite Supply	None	None	None	None
Offshore Wind Turbine	0.00	0.00	0.00	-0.17
Onshore Wind Turbine	0.00	0.00	0.00	-0.20
Power To Heat	0.00	0.00	0.00	0.10
Solar Panel	0.00	0.00	0.00	-0.03

A.11. Original Conveyer Component Description

Conveyer components are characterized by relating two bidirectional flows via a resistance to the potential of the neighbouring bus components. The direction of the two flows is fixed at any given timestep so that one flow is characterized as inflow and the other as outflow. Which flow is which, my vary between time steps. This effectively creates a bidirectional potentially lossy connection between two busses.

Examples range from power lines over gas pipelines to district heating pipeleines. Expecially in the context of passive transmitters. They however require additonal information form the bus components in form of potential levels (like pressure or voltage) and thus are often subject to special investigations of certain network types (power grid, gas grid, etc). Hence they are rarely utilizied in this thesis, since the developed method and therfor its underlying implementations aim to be applicable to a wide range of energy supply system optimization models. These do not necessarily provide a component of this sort.