

Bachelorthesis

Entwicklung eines methodischen
Datenblatt generierenden Vergleichs für
Free Open Source Energiesystem-
Simulationssoftware anhand eines
Fallbeispiels mit den Software-Tools
Oemof und PyPSA

Cedric Körber

Matr.-Nr.: 21639680

Erstprüferin: Prof. Dr.-Ing. Kristin Abel-Günther
Zweitprüfer: M. Sc. Mathias Ammon
Betreuer: M. Sc. Mathias Ammon

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit eigenständig und ohne fremde Hilfe angefertigt habe. Textpassagen, die wörtlich oder dem Sinn nach auf Publikationen oder Vorträgen anderer Autoren beruhen, sind als solche kenntlich gemacht. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Diese Arbeit ist nach bestem Wissen erstellt worden. Für den Inhalt kann jedoch keine Gewähr übernommen werden.

Hamburg, 22.10.2021

(Cedric Körber)

Zusammenfassung

Die vorliegende BA entwickelt eine Methode zum Vergleich von "Free Open Source Energiesystem-Simulationssoftware" (FOSESSyS). Dabei wird ein Datenblatt generiert, anhand dessen der Vergleich stattfindet. Mit Hilfe einer Fallstudie wird anschließend eine exemplarische Durchführung der Methode vorgestellt. Innerhalb dieser Fallstudie wird ein Modell des Strom- und Wärmenetzes der Stadt Hamburg erstellt und mittels der Software-Tools *oemof* und *PyPSA* untersucht. Dafür werden geeignete Szenarien anhand aktueller Emissionsziele entwickelt. Die für die Parametrierung des Modells benötigten Daten werden dabei ausschließlich durch offene Datenquellen bezogen. Eventuelle Aufarbeitung erfolgt dabei mittels "Free Open Source Software".

Die vorliegende Arbeit leistet so einen Beitrag zur "Open-Science" Bewegung.

Abstract

The present BA develops a method for comparing "Free Open Source Energy System Simulation Software" (FOSESSyS). A data sheet is generated on the basis of which the comparison takes place. With the help of a case study, an exemplary implementation of the method is then presented. Within this case study, a model of the electricity and heating network of the City of Hamburg created and examined using the software tools *oemof* and *PyPSA*. For this purpose, suitable scenarios are developed based on current emission targets. The data required to parameterize the model are obtained exclusively from open data sources. Any processing is done using "Free Open Source Software".

The present work thus contributes to the "Open Science" movement.

Inhaltsverzeichnis

Abkürzungsverzeichnis	a
Abbildungsverzeichnis	c
Tabellenverzeichnis	c
Codeverzeichnis	c
1 Einleitung	1
1.1 Ziel der Arbeit	2
2 Theorie	3
2.1 Definition Energiesystemmodellframework	3
2.2 Open Source	3
2.2.1 Open Data	4
2.3 Optimierung von ESM	4
2.3.1 Lineare Programmierung (LP)	4
2.3.2 Gemischt-ganzzahlige lineare Programmierung (MILP)	5
2.3.3 Der Unterschied zwischen Simulation und Optimierung	6
2.4 Klassifizierung von Energiesystemen in der Literatur	6
3 Datenblattentwicklung	12
3.1 Klassifizierungsmerkmale	12
3.1.1 Basisinformationen	12
3.1.2 Mathematische Beschreibung	13
3.1.3 Modellbildung	16
3.1.4 Durchdringung erneuerbarer Energien	18
3.1.5 Kollaborative Eigenschaften	18
3.2 Auswahl der Parameter	19
4 Verwendete Software	22
4.1 Open Energy Modelling Framework (<i>oemof</i>)	22
4.1.1 Projekt-Architektur	23
4.1.2 Optimierung in <i>oemof</i>	24
4.2 Python for Power System Analysis (PyPSA)	25
4.2.1 Komponenten	25
4.2.2 Simulation ohne Optimierung	26
4.2.3 Simulation mit Optimierung	27
4.3 Transforming Energy Simulation Systems Framework (<i>tessif</i>)	27
4.3.1 Struktur	28

4.4 Atlite	28
5 Fallstudie	30
5.1 Der Vergleich von <i>oemof</i> und <i>PyPSA</i> anhand eines Factsheets	30
5.2 Datenquellen	33
5.2.1 Lastgangzeitreihen für den Energiebedarf	33
5.2.2 Energieerzeugungszeitreihen von erneuerbaren Energiequellen	34
5.2.3 Kraftwerksdaten	34
5.2.4 Emissionen	35
5.2.5 Kosten	35
5.3 Modellkomponenten	35
5.3.1 Busse	36
5.3.2 Quellen	36
5.3.3 Senken	37
5.3.4 Transformer	38
5.3.5 Energiespeicher	40
5.3.6 Emissionsziel	41
5.3.7 Zusammenfassung der Parameter	41
5.4 Szenarien	45
5.5 Ergebnisse der Modellszenarien	45
5.5.1 Referenzszenario	46
5.5.2 Szenario „CO2 2030“	47
5.5.3 Szenario „CO2 2050“	49
5.6 Interpretation der Simulationsergebnisse	51
6 Zusammenfassung	53
6.1 Fazit	53
6.2 Synergiepotential	54
6.3 Ausblick	55
Quellen	I
Anlagen	i
Anlage 1	i
Anlage 2	i
Anlage 3	ii
Anlage 4	vi
Anlage 5	vii

Anlage 6	viii
Anlage 7	ix
Anlage 8	x
Anlage 9	xi
Anlage 10	xii
Anlage 11	xiii
Anlage 12	xiv
Anlage 13	xv
Anlage 14	xvi
Anlage 15	xvii

Abkürzungsverzeichnis

AU	Aufgeschlüsselt
BD	Benutzerdefiniert
BHKW	Blockheizkraftwerk
B	Bewertung
BU	Bottom-Up
CO₂	Kohlenstoffdioxid
CSV	comma seperated values
DLR	Deutsche Luft- und Raumfahrtbehörde
E	Erforschung
EA	Eigene Auswahl
EB	Energiebedarf
EEG	Erneuerbare-Energien-Gesetz
EI	Einflüsse
ELS	Elektrizitätssektor
ES	Energiesystem
ESM	Energiesystemmodell
ESMF	Energiesystemmodellframework
ESMG	Energiesystemmodellgenerator
EUJF	Europa Universität Flensburg
EV	Energieversorgung
EZ	Emissionszieleinhaltung
FIAS	Frankfurt Institute for Advanced Studies
FOS	Free Open Source
FOSF	Free Open Software Frameworks
FOSS	Free Open Source Software
FOESSyS	Free Open Source Energiesystem-Simulationssoftware
GuD	Gas- und Dampf-Kombikraftwerk
GZB	Ganzzahligkeitsbedingungen
HSF	Hochschule Flensburg
HYB	Hybrid
IP	Investitionsplanung
IPV	Inneres-Punkte-Verfahren
KM	Kostenminimierung
KWK	Kraft-Wärme-Kopplung
KZ	Kurzzeit
LOPF	Linear Optimal Power Flow
LP	Lineare Programmierung
LZ	Langzeit
MA	Modularer Aufbau
MaStR	Markstammdatenregister
MILP	Mixed-Integer linear Programming

MILP	Gemischt-Ganzzahlige Lineare Programmierung
MÖ	Makroökonomie
MON	Monetär
MS	Mobilitätssektor
O	Optimierung
Ö	Ökonomie
oemof	open energy system modelling framework
OEP	Open Energy Platform
ÖGG	Ökonomisches Gleichgewicht
OMI	Open Modelling Initiative
openmod	Open Energy Modelling Initiative
OPF	Optimal Power Flow
OPSD	Open Power System Data
OSEMF	Open Source Energy Modelling Framework
OSF	Open-Source-Frameworks
OSI	Open Source Initiative
OvGUM	Otto-von-Guericke-Universität von Magdeburg
PTG	Power-to-Gas
PTH	Power-to-Heat
PV	Photovoltaik
PyPSA	Python for Power System Analysis
QL	Qualitativ
QN	Quantitativ
RLI	Reiner Lemoine Institut
R	Rückübertragung
S	Simulation
TB	Tabellenblatt
TD	Top-Down
tessif	Transforming Energy Simulation Systems Framework
V	Vorhersage
WS	Wärmesektor
ZNES	Zentrum für Nachhaltige Energiesysteme

Abbildungsverzeichnis

Abbildung 1: Open data, open source und open access im Modellierungsprozess [19].	4
Abbildung 2: Struktur der oemof-Architektur [7].	22
Abbildung 3: Klassen im oemof-Quellcode [7].	23
Abbildung 4: Beispiel eines Energiesystems in oemof [7].	24
Abbildung 5: Beispiel eines Energiesystems in PyPSA [2] - CHP Combined Heat and Power Plant	26
Abbildung 6: Vergleich von atlite und opsd Datensätzen zur Erzeugung elektrischer Energie aus PV-Anlagen in Deutschland 2012 [64].	34
Abbildung 7: Energiesystem Hamburgs mit visualisierten Knotenpunkten mit Hilfe der Funktion "tessif.visualize.nxgrph".	36
Abbildung 8: Vergleich oemof und PyPSA Installierte Leistung "pv1" und "won1" im Referenzszenario	46
Abbildung 9: Vergleich von elektrischer Energieerzeugung von pp1 und pp2 im Szenario "CO2 2030" gegenüber dem Referenzszenario.	48
Abbildung 10: Vergleich von oemof und PyPSA der Installierten Leistung von pv1 und won1 in den drei Szenarien	50
Abbildung 11: Vergleich von oemof und PyPSA der übertragenen Leistung in die excess-Senke in Szenario "CO2 2030" und "CO2 2050"	50
Abbildung 12: Emissionen ausgewählter Treibhausgase in Deutschland nach Kategorien in Tsd. t Kohlendioxid-Äquivalenten (Quelle: https://www.umweltbundesamt.de/sites/default/files/medien/384/bilder/8_tab_thg-emi-kat_2020.png , Zugriff am: 12.04.2021).	i
Abbildung 13: Lastgang Stromsektor ES HH Referenzszenario aller Komponenten	vi
Abbildung 14: Summe erzeugter bzw. verbrauchter elektrischer Energie Referenzszenario aller Komponenten	vii
Abbildung 15: Lastgang Wärmesektor ES HH Referenzszenario aller Komponenten	viii
Abbildung 16: Summe erzeugter bzw. verbrauchter thermischer Energie Referenzszenario aller Komponenten	ix
Abbildung 17: Lastgang Stromsektor ES HH Szenario "CO2 2030" aller Komponenten	x
Abbildung 18: Summe erzeugter bzw. verbrauchter elektrischer Energie Szenario "CO2 2030" aller Komponenten	xi
Abbildung 19: Lastgang Wärmesektor ES HH Szenario "CO2 2030" aller Komponenten	xii
Abbildung 20: Summe erzeugter bzw. verbrauchter thermischer Energie Szenario "CO2 2030" aller Komponenten	xiii
Abbildung 21: Lastgang Stromsektor ES HH Szenario "CO2 2050" aller Komponenten	xiv
Abbildung 22: Summe erzeugter bzw. verbrauchter elektrischer Energie Szenario "CO2 2050" aller Komponenten	xv
Abbildung 23: Lastgang Wärmesektor ES HH Szenario "CO2 2050" aller Komponenten	xvi
Abbildung 24: Summe erzeugter bzw. verbrauchter thermischer Energie Szenario "CO2 2050" aller Komponente	xvii

Tabellenverzeichnis

<i>Tabelle 1: Vergleich der Klassifizierungsparameter für die Charakterisierung von ESM(F/G) in der Literatur. EA – Eigene Auswahl</i>	9
<i>Tabelle 2: generisches Datenblatt</i>	20
<i>Tabelle 3: Vergleich von oemof und PyPSA anhand eines Factsheets</i>	31
<i>Tabelle 4: Zusammenfassung der Komponentenparameter von oemof und PyPSA</i>	43
<i>Tabelle 5: Globale Ergebnisdaten Referenzszenario</i>	46
<i>Tabelle 6: Globale Ergebnisdaten Szenario „CO2 2030“</i>	47
<i>Tabelle 7: Globale Ergebnisdaten Szenario „CO2 2050“</i>	49

Codeverzeichnis

<i>Code 1: Funktion zur Erstellung des Cutout in atlite</i>	29
<i>Code 2: Beispielfunktion zur Erstellung der Zeitreihe zur Energieerzeugung von Wind- und PV-Anlagen</i> .	29
<i>Code 3: Deklaration der spezifischen CO₂-Emissionen in oemof anhand des Beispiels "gas_supply"</i>	37
<i>Code 4: Deklaration der spezifischen CO₂-Emissionen in PyPSA anhand des Beispiels "gas_supply"</i>	37
<i>Code 5: Deklaration der Senken in oemof anhand des Beispiels der elektrischen Last</i>	38
<i>Code 6: Deklaration der Senken in PyPSA anhand des Beispiels der elektrischen Last</i>	38
<i>Code 7: Erstellung eines KWK-Kraftwerkes in oemof anhand des Beispiels HKW ADM</i>	38
<i>Code 8: Implementierung der dictionary-Erweiterungen in PyPSA anhand des Beispiels "bus2"</i>	39
<i>Code 9: Erstellung eines KWK-Kraftwerkes in PyPSA anhand des Beispiels HKW ADM</i>	40
<i>Code 10: Deklaration von Energiespeichern in oemof anhand des Beispiels "Electrical Storage Unit"</i>	40
<i>Code 11: Deklaration von Energiespeichern in PyPSA anhand des Beispiels "Electrical Storage Unit"</i>	41

1 Einleitung

Auf der Weltklimakonferenz wurde beschlossen, dass, um das 1,5 Grad Ziel der Erderwärmung im Vergleich zur Referenztemperatur aus vorindustrieller Zeit einhalten zu können, Klimaneutralität in jedem Volkswirtschaftlichen Sektor der teilnehmenden Staaten herrschen solle. Dieses Ziel kann nur durch den Ausbau von erneuerbaren Energien erreicht werden. Diese stellen die Energiewirtschaft jedoch vor eine große Herausforderung. Durch die fluktuierende Energiebereitstellung erneuerbarer Energiequellen kann ohne Speichermöglichkeiten kein stabiles Energienetz gewährleistet werden. Die technischen Möglichkeiten der Zwischenspeicherung und Umwandlung dieser Energie sind vielfältig und reichen von Power-to-Gas (PTG) über Power-to-Heat (PTH) zu chemischen Energiespeichern. Eine weitere Veränderung im Energiesektor wird außerdem die enge Verflechtung der Sektoren eines Energiesystems sein. So wird beispielsweise durch den Trend hin zur E-Mobilität der Verkehrssektor direkt mit dem Elektrizitätssektor verknüpft. Dies ist bei der Energieversorgungsstruktur durch fossile Rohstoffe nicht der Fall, da nur eine indirekte Verbindung zum Elektrizitätssektor besteht.

Diese Umwandlungs- bzw. Speichermöglichkeiten sowie Verflechtungen der Sektoren müssen jedoch auch sinnvoll und nachhaltig in das Energiesystem eingebaut werden. Die Modellierung von Energiesystemen ist hier eine gute Möglichkeit zur Planung und Erforschung dieser Veränderungen der Energiesysteme.

Neben dem Energiesystemmodell, im Folgenden mit ESM abgekürzt, werden auch externe Daten in Form von Datenbankeinträgen zu Komponenten und Zeitreihen zur Simulation eines Energiesystems benötigt. Besonders im Zusammenhang mit erneuerbarer Energieerzeugung sind viele externe Daten erforderlich. So benötigt bspw. die Simulation von Windenergieerzeugungs- und Photovoltaikanlagen räumlich und zeitlich hoch aufgelöste Wetterdaten. Das Thema der Integration erneuerbarer Energien in das bestehende Energiesystem bzw. das Ersetzen von konventionellen mit fossilen Primärquellen betriebenen Kraftwerken durch ebendiese gewinnt immer mehr an Relevanz. Daraus folgt, dass auch vermehrt wissenschaftliche Forschung auf dem Gebiet der erneuerbaren Energien betrieben wird und somit auch die Zahl der veröffentlichten ESM ansteigt. Daraus folgt, dass auch inhaltliche Überschneidungen von Modellen nicht selten sind.

Dem Problem der Repetitivität von wissenschaftlicher Arbeiten nimmt sich insbesondere die Open Energy Modelling Initiative [3] an. So plädieren Pfenninger, DeCarolis et al. (2017) [4] dafür, dass Modelle und Daten frei verfügbar sein sollten, besonders aus folgenden vier Gründen:

- Der Verbesserung der wissenschaftlichen Qualität, dadurch, dass Fehler in offenen Codes leichter gefunden werden können,
- Effektivere Zusammenarbeit zwischen Wissenschaft und Gesetzgebern,
- Der Verbesserung der Produktivität durch Kollaborationen, um doppelte Arbeit bspw. beim Erstellen eines Modelles zu verhindern und
- Das Vorantreiben der sozialen Debatte, um Energiewirtschaftliche Themen für jeden überschaubar machen zu können.

Ein weiteres Projekt, welches sich der Ähnlichkeitsbetrachtung von Open Source Energy Modelling Framework (OSEMF) widmet, nennt sich open_MODEX. Dieses Modellexperiment, das von dem Reiner Lemoine Institut geleitet wird, soll das „Synergiepotential [...] von Open-Source-Frameworks (OSF) in

der Energiesystemanalyse“ [5] von den fünf OSF *oemof* [6, 7], *urbs* [8], GENESYS-2 [9], GENeSYS-MOD [10] und Balmorel [11] ermitteln.

Die beiden, in dieser Arbeit verglichenen, Frameworks zeichnen sich durch o.g. Gründen aus, indem sie ihren Code und wissenschaftliche Arbeit frei für jeden durch eine offene Lizenz verfügbar machen und im Falle von *oemof* darauf abzielen „community building“ zu betreiben, sodass sich jeder an der Entwicklung des Quellcodes beteiligen kann [7]. Sie umfassen dabei ein großes Spektrum der Modellbildung und sind so für die Entwicklung von ESM ein viel genutztes Werkzeug [12, 13].

1.1 Ziel der Arbeit

Ziel dieser Bachelorthesis ist es ein Datenblatt zu generieren, anhand dessen der Vergleich von ESMF-Software durchgeführt werden kann. Mit Hilfe einer Fallstudie wird anschließend eine exemplarische Durchführung der Methode vorgestellt. Innerhalb dieser Fallstudie wird zusätzlich ein Modell des Strom- und Wärmenetzes der Stadt Hamburg erstellt und mittels der Software-Tools *oemof* und *PyPSA* untersucht.

In Kapitel 2 werden hierfür zunächst die theoretischen Grundlagen der ESM und Open Source Bewegung sowie die Sichtung von bereits veröffentlichten Arbeiten bezüglich der Klassifizierung von ESM(F) aufgeführt. Zusätzlich wird die Struktur der zu vergleichenden Frameworks theoretisch erläutert.

Aus den zusammengestellten Klassifizierungsmerkmalen aus dem vorangegangenen Kapitel folgt in Kapitel 3 die Zusammenstellung und Erläuterung der Vergleichsparameter, die in Kapitel 2 aus der Literatur herausgearbeitet wurden, zur Erstellung des Datenblattes. Am Ende des Kapitels wird das generische Datenblatt als Vorlage zur Klassifizierung dargelegt und die Wahl der Parameter begründet.

Die verwendete OSS zur Entwicklung des Modellversuches in der Fallstudie wird in Kapitel 4 detailliert vorgestellt.

Die Fallstudie in Kapitel 5 zeigt den Vergleich der beiden Frameworks *oemof* und *PyPSA* einerseits in Form des in Kapitel 3 erstellten Datenblattes, andererseits in Form des modellierten ES Hamburgs des Strom- und Wärmesektors jeweils in den genannten Frameworks. Die Vergleichsszenarien wurden anhand der Reduktionsziele des CO₂-Ausstoßes für die Jahre 2030 und 2050 entwickelt, indem die ausgestoßene Menge aller Energieerzeugungsanlagen den jeweiligen jährlichen Grenzwert des Modells darstellt. Anschließend werden die Ergebnisse der Fallstudie zusammengefasst und analysiert.

Das Fazit bezüglich der Fallstudie in Kapitel 5 sowie der Anwendung des in Kapitel 3 entwickelten Datenblattes wird in Kapitel 6 vorgenommen. Zusätzlich wird ein Ausblick zur Entwicklung der FOSESyS gegeben und welchen Beitrag diese Arbeit hierzu leistet.

2 Theorie

In diesem Kapitel werden allgemeine Begriffe zur ESM und die theoretischen Grundlagen erläutert, bereits veröffentlichte Berichte, die ESM vergleichen sowie die in dieser Thesis verglichenen Frameworks vorgestellt.

2.1 Definition Energiesystemmodellframework

Zunächst ist eine Unterscheidung der Begriffe Energiesystemmodell (ESM), -Generator (ESMG) und -Frameworks (ESMF) vorzunehmen. Ein ESM ist nach Wiese et al. (2018) [14] und OEP Community [15] eine konkrete Repräsentation eines Systems in der realen Welt für eine bestimmte Region und Zeit mit einer bestimmten Auswahl an Technologien und anderen vom Modellbenutzer berücksichtigten Aspekten. ESM können mit Hilfe von ESGM erschaffen werden, welche es dem Benutzer erlauben Modelle mit einem definierten analytischen und mathematischen Ansatz zu erstellen [14]. Ein ESMF ist ein strukturierter Programmwerkzeugkasten, welcher eine Vielzahl von Sub-Frameworks und ESGM sowie spezifische Modelle (z.B. zu erneuerbaren Energieerzeugern) enthält [14]. Diese Definitionen gelten für die folgende schriftliche Ausarbeitung.

2.2 Open Source

Die Softwareentwicklung basierend auf dem Teilen und der kollaborativer Verbesserung von Quellcodes besitzt eine nahezu genauso lange Geschichte wie die Softwareentwicklung an sich [16]. In den späten 1990er Jahren wurde das Interesse an Open Source größer, u.a. dadurch, dass Linux Einzug in den Mainstream fand und Netscape den Code ihres Browsers Netscape Navigator freigab, aus welchem die heutige Browseranwendung Firefox hervorging [17]. 1998, im selben Jahr, in welchem der Code des Netscape Navigator zur freien Verfügung gestellt wurde und somit die Open Source Bewegung, wie man sie heute kennt, anfang, gründete sich die Open Source Initiative (OSI) [16, 17]. Die Initiative vertritt die Interessen der Open Source Bewegung, stellt Bildungsangebote zu Open Source bereit und ist die zentrale Verwaltungsstelle für Open Source Projekte [16]. In der Definition von Open Source, welche als Manifest der OSI und der Open Source Bewegung im Allgemeinen gilt, sind 10 Kriterien festgelegt, an welche sich die Entwickler einer Anwendung halten müssen, um diese mit einer Open Source Lizenz ausstatten zu können. Diese Kriterien sind der Website der OSI [18] zu entnehmen.

Die Open Energy Modelling Initiative (openmod) [3] unterstützt die Ansicht der OSI im Hinblick auf die Open Source Verfügbarkeit von ESM. So führe eine offenere Struktur in der Energiesystemmodellierung zu erhöhter Transparenz und Qualität und würde „wasteful double-work“ verhindern [3]. Die erhöhte Transparenz durch Offenheit der ESM führe nach Pfenninger et al. [4] zu einer Verbesserung des öffentlichen Diskurses, indem es die Entscheidungsprozesse und die Gründe für Maßnahmen in der Energiepolitik durchleuchte.

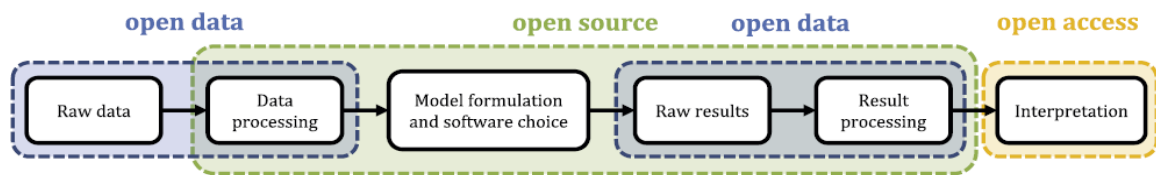


Abbildung 1: Open data, open source und open access im Modellierungsprozess [19].

2.2.1 Open Data

Daten in der ESM sind sowohl die Ein- als auch Ausgabe des Modellierungsprozesses. Openmod [3] weist auf die Notwendigkeit der Open Data hin, da so der Gesamtmodellierungsprozess geöffnet werden kann, welcher in Abbildung 1 dargestellt ist. Rohdaten im Bereich der Energieerzeugung und -nutzung sind weit verbreitet von der Qualität jedoch sehr schwankend, laut S. Pfenninger et al. [19]. Anders als bei Programmierskripten werden bei frei verfügbaren Eingabedaten oft keine Lizenzen vergeben, was jedoch nicht impliziert, dass eine legale Genehmigung zum Nutzen und Teilen dieser Daten besteht [19]. Die Datenbanken der Open Power System Data (OPSD) [20], des FfE Open Data Portal [21], sowie der Open Energy Platform (OEP) [22] sind hierbei gute Quellen für Input- und Modelldaten jeglicher Art. Die OEP stellt neben der Datenbank auch Factsheets bereit und unterstützt somit den gesamten Modellierungsprozess. Die OEP ist ein deutsches Instituts- und Universitätsübergreifendes Forschungsprojekt, an welchem u.a. das Reiner Lemoine Institut und die Europa-Universität Flensburg mit dem open_MODEX Projekt teilhaben [5, 15].

2.3 Optimierung von ESM

In der Mathematik baut die Methode der Optimierung auf der Minimierung bzw. Maximierung einer realen Funktion auf, bei welcher systematisch die Werte der Variablen innerhalb der zuvor definierten Grenzen (Nebenbedingungen) variiert werden. Sie wird zumeist bei einer großen Anzahl von Variablen, wie bei der Energiesystemanalyse, angewandt.

Oemof und *PyPSA* nutzen zur Optimierung ihrer Ergebnisse sowohl Verfahren der Linearen Programmierung (LP) als auch der Gemischt-Ganzzahlige Lineare Programmierung (*MILP*) und greifen dafür auf sog. Solver zurück, welche in Kapitel 3.1.2 erläutert werden.

2.3.1 Lineare Programmierung (LP)

Lineare Programme beschreiben das Problem der Optimierung mit linearen Zielfunktionen und Variablen aus dem Bereich der reellen Zahlen. Sie werden im Bereich der Modellierung auch oft in Verbindung mit dem Begriff BU-Modell gebracht [23]. Formal beschrieben werden sie folgendermaßen:

$$\min_{x \in \mathbb{R}} \{c^T x \mid Ax \leq b, x \geq 0\} \quad (1)$$

Dabei stellen x die Variablen, c und b die Vektoren und A eine Matrix dar.

Die LP-Probleme werden in der Praxis durch unterschiedliche Verfahren, meist dem *Simplex*- und dem Innere-Punkte-Verfahren (IPV), gelöst. Vereinfacht dargestellt löst das *Simplex*-Verfahren das Optimierungsproblem, indem es die Grenzen des zulässigen Lösungsbereiches bis zum Inneren der Grenzen durchläuft, bis keine Verbesserung mehr möglich ist. Das IPV nähert sich der optimalen Lösung in entgegengesetzter Richtung, durch das Innere des zulässigen Bereiches, an.

Bei sehr großen Optimierungsproblemen und dünn besetzten Matrizen sind die IPV den *Simplex*-Verfahren, laut Mainzer 2019 [24], überlegen und finden dort bevorzugt Anwendung. Das *Simplex*-Verfahren ist auch in anderen Verfahren, wie etwa dem *Branch-and-Cut* Verfahren, implementiert.

2.3.2 Gemischt-ganzzahlige lineare Programmierung (MILP)

Die Gemischt-ganzzahlige lineare Programmierung (engl.: mixed-integer linear programming (MILP)) bietet neben der linearen Optimierung der Variablen außerdem Ganzzahligkeitsbedingungen (GZB) an, bei denen bspw. binäre Entscheidungen getroffen werden müssen, ob in eine Anlage investiert wird oder nicht. Die formale Beschreibung des Optimierungsproblems wird mit folgender Formel beschrieben:

$$\min_{x_i \in \mathbb{R}, x_j \in \mathbb{Z}^n} \{c^T x \mid Ax \leq b, x \geq 0\} \quad (2)$$

Durch die GZB wird ist das Problem der *MILP* deutlich schwerer zu lösen als das der LP. Das bekannteste und wohl am häufigsten verwendete Lösungsverfahren ist das *Branch-and-Cut* Verfahren und wird auch von den, für die Optimierung in *oemof* und *PyPSA* verwendeten, *Solver* wie *GLPK* oder *cbc* angewandt. Dieses Verfahren ist eine Zusammenführung des Branch-and-Bound Verfahren mit dem Schnittebenenverfahren.

Das *Branch-and-Cut* Verfahren verfolgt den Ansatz zunächst die Ganzzahligkeitsbedingungen des Problems außer Acht zu lassen und mit Hilfe eines der oben genannten LP-Verfahren die sog. LP-Relaxierung des *MILP*-Problems zu lösen. Die LP-Relaxierung beschreibt genau diese Vorgehensweise zunächst ein *MILP*-Problem ohne Beachtung der GZB zu lösen. Erfüllt die LP-Relaxierung bereits die GZB, so handelt es sich um die global optimale Lösung.

Anderenfalls werden mit Hilfe des Schnittebenenverfahrens zunächst weitere Ungleichungen, welche von den Punkten der GZB erfüllt werden können, zu dem Lösungssystem hinzugefügt. Diese können jedoch noch nicht von der aktuellen Lösung der LP-Relaxierung erfüllt werden. Beim erneuten Lösen des Systems mit Hilfe des LP-Verfahrens soll die neue Lösung sich dem gesuchten, durch die LP-Relaxierung ermittelten, Optimum annähern. Ist die Lösung ganzzahlig, so handelt es sich um die optimale Lösung, ansonsten muss nach neuen Schnittebenen gesucht werden. Werden keine weiteren Schnittebenen mehr gefunden, ohne dass eine ganzzahlige Lösung erscheint, so wird die Branch-and-Bound Methode angewandt. Sie unterteilt das Problem sukzessive in kleinere Teilprobleme (engl.: *branching*), wobei der zulässige Bereich der Variablen im Rahmen der Ganzzahligkeitsbedingungen eingeschränkt wird (engl.: *bounding*). Gelingt es die optimalen Lösungen für die Teilprobleme zu finden, so ist die beste Lösung auch optimal. Solange Variablen vorliegen, die die GZB verletzen, kann mit den Teilproblemen auf gleiche Weise verfahren werden. Hierbei entsteht durch die Lösungspfade der Teilprobleme ein Lösungsbaum.

Sind an einer beliebigen Stelle alle GZB erfüllt, so stoppt das Verfahren an diesem Zweig, da eine zulässige Lösung gefunden wurde. Diese Lösung ist nun die Referenz für weitere Verzweigungen und stellt die obere Grenze für das Problem dar. Wird eine schlechtere Lösung in einem Teilproblem gefunden, so wird dieser Zweig nicht mehr verfolgt und kann von dem Baum abgeschnitten werden (*bound*). Die beste Lösung aller aktiven LP-Teilprobleme stellt die untere Grenze des Problems dar und kann nicht unterschritten werden, deswegen wird die Entfernung von der aktuell besten Lösung, also der oberen Grenze, zu eben dieser unteren Grenze als Indikator, der sog. *Gap*, für die Qualität der aktuell besten Lösung gesehen. Wenn der *Gap*-Wert 0 erreicht wird, so handelt es sich also um die optimale Lösung des *MILP*.

Die Dauer der Berechnung zur Lösungsfindung kann exponentiell mit der Anzahl der Variablen mit GZB ansteigen, wobei auch nach, im Gegensatz zu anderen Lösungsfindungsmethoden, kurzer Dauer eine gute zulässige Lösung des Problems gefunden werden kann. Der Nachweis der *Gap* und somit Optimalität dauert im Vergleich dazu jedoch sehr lange. Aus diesem Grund wird der *Gap* von den *Solvern* meist auch bei einem Wert größer als 0 zugelassen.

2.3.3 Der Unterschied zwischen Simulation und Optimierung

Häufig werden Begriffe der Energiesystemmodellierung in wissenschaftlichen Ausarbeitungen nicht präzise genutzt, was es für den unerfahrenen Nutzer schwer macht, diese korrekt verwenden zu können. Dieses Problem gilt auch für die Begriffe „Simulation“ und „Optimierung“, weswegen diese im Folgenden definiert werden.

Worrell, Ramesohl et al. 2004 [25] lieferten für die Unterscheidung beider Begriffe die passenden Definitionen. So werden Optimierungsmodelle dazu eingesetzt, die optimale Zusammenstellung technologischer Auswahl zu treffen, um ein zuvor definiertes Ziel mit den geringsten Kosten erreichen zu können. Simulationsmodelle hingegen liefern eine „[...] quantitative Darstellung exogen definierter Szenariostrategien [...]“ [25], d.h., dass der Modellierer des Systems alle Bedingungen für den Betrieb des ES zuvor vorgibt und das Modellierungsprogramm nur die Ergebnisse ohne jegliche endogenen Parameter liefert. Sie ermöglichen die Betrachtung und Bewertung der Zusammenhänge von bspw. verschiedenen politischen oder sozialen Maßnahmen und dem ES. Die Nutzung von Optimierungsmodellen ist beschränkt auf diskrete Energieumwandlungstechnologien und typisierte Energienutzungen (wie Autos oder Häuser) [26]. Der Ansatz der Simulation wird hingegen oft eingesetzt, um den zukünftigen Energiebedarf und die damit verbundenen Emissionen zu projizieren [26].

2.4 Klassifizierung von Energiesystemen in der Literatur

Zu ESM wurden bereits viele Klassifizierungsansätze und Vergleichsparameter entwickelt. So klassifizieren Greenberger und Richels (1979) [27] ESM nach zwei Kategorien: Einerseits mit Fokus auf ausschließlich ein Modell andererseits mit Hilfe einer sogenannte Forums-Analyse, bei welcher unterschiedliche Modelle miteinander verglichen werden.

Grubb et al. (1993) [28] und N.M.J.P. van Beek (1999) [29] ordnen ESM mit Hilfe von technischen Analysen nach den jeweiligen Spezifikationen der Modelle ein. Betrachtet werden in diesen der analytische und mathematische Ansatz, die verwendeten Daten, die Abdeckung der vier energiewirtschaftlichen Sektoren (Industrie, Verkehr, Wohnen, Dienstleistungen und Handel) sowie die geographische und zeitliche Abdeckung des Modells. Zusätzlich zu den genannten Merkmalen in [28] stellt [29] außerdem den Zweck des ESM, die Modellstruktur (interne/externe Annahmen, Grad der Endogenisierung) und die zugrunde gelegte Methodik als wichtige Klassifizierungspunkte heraus.

Pandey (2002) [30] entwickelt eine Methode, welche bei der Klassifizierung von ESM besonderen Fokus auf die Darstellbarkeit der Modelle zu der Anwendung in Entwicklungsländern legt. Dabei wird das ESM mit Kategorien klassifiziert, die bereits auch in [28] und [29] verwendet werden: den analytischen Ansatz, geographische Abdeckung, Sektoren, Zeithorizont, sowie die Problemstellung auf die das Modell angepasst wurde.

Durch den Fokus der heutigen Energiewirtschaft auf erneuerbare Energien müssen die ESM auch nach diesen klassifiziert werden, weshalb Connolly et al. (2010) [31] eine Einordnung von ESM nicht nur nach den o.g. Gesichtspunkten vornehmen, sondern auch die Durchdringung von erneuerbaren Energien in den jeweiligen Modellen untersuchen. So wird in der Klassifizierung geprüft, ob das Modell mit 100% erneuerbaren Energien im Stromsektor, als auch in allen Energiewirtschaftssektoren zu simulieren ist. Amerighi et al. (2010) [32] führen zur Einordnung weitere Themen wie Fähigkeiten zur Analyse von Gesetzen/Richtlinien, Verbindungsmöglichkeiten eines makroökonomischen Modells mit einem Teilgleichgewichtsmodell oder den ökonomischen Umfang des ESM ein.

Einen ausführlichen Überblick über Klassifizierungskriterien von ESM liefern Lopion et al. (2018) [33], indem die Klassifizierungsansätze in [29], [31] und [34] gegenübergestellt werden und ein eigene Auswahl von Klassifizierungsparametern von den Autoren gewählt wird. So wird in [33] zusätzlich vorgeschlagen, die Analyse der Programmierumgebung und die Untersuchung der Transformationspfade in der methodischen Klassifizierung von ESM vorzunehmen.

Bouw et al. (2021) [35] fokussieren sich in ihrer Arbeit auf eine lokale geographische Auflösung bei der Klassifizierung von ESM. Die Autoren interessieren sich neben Basisinformationen zu den Modellen, besonders für die Integration von sozialen Faktoren in die aufgeführten techno-ökonomischen ESM. Als grafische Auswertungsmethode dienen Netzdiagramme.

Insbesondere Frameworks in der Energiesystemmodellierung betrachten u.a. Wiese et al. (2018) [14], Ringkjøb et al. (2018) [36], Kriechbaum et al. (2018) [37] und Auer et al. (2019) [38]. Dabei werden Kriterien von ESM-Klassifizierungen an Frameworks angepasst und weitere wie die modellierten Komponenten zum Hinzufügen zum ESM oder die Durchführbarkeit einer Netzanalyse von elektrischen Netzwerken entwickelt [38].

Eine Herausforderungs-Eigenschaftsmatrix zur Evaluierung von ESMF wurde von F. Wiese et. al (2018) [14] ausgearbeitet und ein Fallbeispiel mit der Klassifizierung von *oemof* angebracht. Die fünf Herausforderungen, welchen ein ESM heutzutage ausgesetzt ist, sind folgende Punkte:

- Es muss eine ausreichende Komplexität bzgl. Sektorenkopplung, technischer/ zeitlicher/ geographischer Auflösung, Eingabedaten und Ergebnisdatenverarbeitung vorweisen,

- möglichst wenig Unsicherheiten bspw. bezüglich der Sprache oder Entscheidungen, die von dem Programm getroffen werden, aufweisen,
- interdisziplinäre Eigenschaften wie die Berücksichtigung von sozialen Faktoren oder Umweltaspekten in das Modell mit einbeziehen,
- die Benutzung sollte Entwicklern, Nutzern und Entscheidungsträgern einfach fallen und
- es sollte die wissenschaftlichen Standards ermöglichen bzw. erfüllen [14].

Dazu beitragen sollen eine open-source Philosophie, kollaborative Entwicklung und strukturelle Eigenschaften wie Modularität, objektorientiertes Programmieren, ein generisches Konzept der Repräsentation des ES sowie Datenmodelle [14]. Auer et al. [38] vergleichen ESMF anhand der Möglichkeit mit diesen ein autarkes Energiesystem (Off-Grid-Energiesystem) modellieren zu können. Ebenfalls auf die netzbasierte Modellierung von ES bzw. „multi-energy systems“ beziehen sich Kriechbaum et al. (2018) [37], indem sie die derzeitigen Herausforderungen dieser anhand einer Literaturrecherche zusammentragen und die drei OSMF *Calliope*, *oemof* und *urbs* hingehend dieser Kriterien analysieren.

Einen generellen Überblick über ESMF schafft indes die Arbeit von Ringkjøb et al. [36], in welcher 75 Modellframeworks sowie Modellgeneratoren miteinander verglichen werden. Es werden zusätzlich die Modellierungswerkzeuge herausgearbeitet, die die Herausforderung des Erstellens von ES mit Integration fluktuierender erneuerbarer Energiequellen bewältigen können und somit für heutige Energiesysteme mit ebendiesen eingesetzt werden können. Da jedes ESM(G/F) für ein eigenes Analyseziel von ES entwickelt wurde, soll mit [36] ein Überblick über die Modelle gegeben werden, sodass der Anwender ein, für sein Forschungsvorhaben, passendes Modellierungsframework bzw. einen passenden Modellgenerator auswählen kann. Für einen schnellen und prägnanten Überblick über ESM(F) sind Datenblätter bzw. Factsheets ein geeignetes Werkzeug. Sie werden u.a. von der Open Modelling Initiative [3], Open Energy Platform [15], Amerighi et al. (2010) [32] und Lechtenboehmer et al. (2018) [39] als Vergleichswerkzeug genutzt. Diese sollen einen standardisierten Überblick über die Eigenschaften eines ESM(F) schaffen. Besonders das Factsheet von [15] deckt die wesentlichen Eigenschaften zur Charakterisierung von ESMF, ESM und Szenarien ab. So werden die ESMF bezüglich der Basisinformationen, mathematischen Beschreibung, Offenheit und Modellbildung charakterisiert.

In Tabelle 1 werden die Ergebnisse der Literaturrecherche tabellarisch zusammengefasst. Zusätzlich wird angegeben, welche Parameter für die Entwicklung des Datenblatts in Kapitel 3 verwendet werden. Dort wird auch darauf eingegangen, weshalb einige Parameter keine Anwendung finden. Die Klassifizierungsparameter sind in die fünf Überpunkte Basisinformationen, Mathematische Beschreibung, Modellbildung, Erneuerbare Energien und kollaborative Eigenschaften unterteilt. Zur Klassifizierung und Beschreibung von Energiesystemmodellen und -modellierungssoftware existieren wesentlich mehr wissenschaftliche Veröffentlichungen, welche jedoch aus Gründen des Umfangs dieser Arbeit vernachlässigt wurden.

Tabelle 1: Vergleich der Klassifizierungsparameter für die Charakterisierung von ESM(F/G) in der Literatur. EA – Eigene Auswahl

1	Grubb, Edmonds et al. 1993 – The Costs of Limiting Fossil-Fuel [28]
2	van Beeck 1999 – Classification of Energy Models [29]
3	Pandey 2002 – Energy policy modelling [30]
4	Connolly, Lund et al. 2010 – A review of computer tools [31]
5	Amerighi, Ciorba et al. Oktober 2010 – Analysing Transition Planning and Systemic [32]
6	Després, Hadjsaid et al. 2015 – Modelling the impacts of variable [40]
7	Wiese, Hilpert et al. 2018 – A qualitative evaluation approach [14]
8	OEP Community 4/26/2021 – Open Energy Platform OEP [15]
9	Open Models 5/13/2021 [41]
10	Auer, Liße et al. 2019 – Power-flow-constrained asset optimization for off-grid [38]
11	Kriechbaum, Scheiber et al. 2018 – Grid-based multi-energy systems—modelling [37]
12	Lechtenboehmer, Pregger et al. 2018 – RegMex [39]
13	Lopion, Markewitz et al. 2018 – A review of current challenges [33]
14	Bouw, Noorman et al. 2021 – Local energy planning [35]
15	Ringkjøb, Haugan et al. 2018 – A review of modelling tools [36]

Klassifizierungsparameter	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	E A
Basisinformationen																
Entwickler				X	X			X	X		X	X			X	X
Veröffentlichungsdatum					X			X		X		X			X	X
Modellzweck		X	X	X	X			X				X	X	X	X	X
Geographische Abdeckung	X	X	X	X	X		X	X	X		X	X	X	X	X	X
Sektorenabdeckung	X	X	X		X		X	X	X	X	X		X	X	X	X
Zeitabdeckung	X	X		X	X	X	X	X		X	X	X	X	X	X	X
Version								X								X
Dokumentation							X				X					X
Letzter Commit (Aktualität)					X					X						
Anzahl der Herausgegebenen Versionen										X						
Anzahl der Mitwirkenden										X						
Anzahl von Nutzern				X												
Downloads/Verkäufe				X								X				
Mathematische Beschreibung																
Programmierungsumgebung					X		X	X	X	X	X	X	X		X	X
Solver							X	X								X
Eingabedaten	X			X		X	X	X			X	X		X		X

Klassifizierungsparameter	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	E A
<i>Ergebnisdaten(-verarbeitung)</i>	X				X		X	X	X		X	X				X
<i>Analytischer Ansatz</i>	X	X	X		X			X			X		X		X	X
<i>Mathematischer Ansatz</i>		X						X			X				X	X
<i>Methodik / Lösungsmethode</i>	X	X		X	X	X		X	X	X	X	X		X	X	X
<i>Optimierungsbeschränkungen</i>								X		X						X
<i>Dynamikentscheidung bei der Optimierung (Optimierungs-) Ziel</i>					X	X		X	X			X				X
<i>GUI</i>					X			X								X
Modellbildung																
<i>Repräsentierte Energieformen</i>						X						X				X
<i>Berücksichtigte Energieträger</i>					X							X				X
<i>Zeitschritte/ Zeitliche Auflösung</i>				X				X	X				X	X	X	X
<i>Räumliche Auflösung</i>									X		X	X	X			X
<i>Lastprofile</i>																X
<i>Ökonomische Abdeckung</i>					X							X			X	X
<i>Technologische Abdeckung</i>				X	X			X	X	X		X			X	X
<i>Netzmodelle</i>								X								X
<i>Modellstruktur</i>	X												X		X	X
<i>Generisches Konzept</i>							X									X
<i>Modularität</i>							X									X
<i>Erweiterbarkeit</i>								X								X
<i>Verbindungsmöglichkeit mit anderen Programmen zur Modellbildung</i>					X											X
<i>Endogene/Exogene Variablen/ Grad der Endogenisierung</i>					X	X			X			X				
<i>Systemansatz</i>						X										
<i>Unbestimmtheit</i>							X					X				
<i>Detailgrad</i>											X					

Klassifizierungsparameter	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	E A
<i>Fähigkeit zur Analyse von Gesetzen/Richtlinien</i>					X							X				
<i>Elektrische Netzanalyse</i>									X	X						
<i>Fähigkeit Planbedürfnisse und Prioritäten zu modellieren und evaluieren</i>					X											
<i>Interdisziplinarität</i>							X									
Erneuerbare Energien																
<i>Durchdringung erneuerbarer Energien</i>				X			X	X							X	X
<i>Einspeisedaten erneuerbarer Energien</i>										X						X
<i>Emissionen</i>					X										X	X
Kollaborative Eigenschaften																
<i>Verfügbarkeit</i>				X	X		X	X	X		X	X	X	X	X	X
<i>Offenheit der Daten</i>							X	X	X							X
<i>Lizenz</i>							X	X	X	X	X					X
<i>Entwicklerperspektivenspektrum</i>							X									X
<i>Einheitlichkeit der Terminologie</i>							X									X
<i>Wissenschaftliche Standards</i>							X									X
<i>Zur Verfügung gestellte Daten (Beispiele, Dokumentation etc.)</i>								X								X
<i>Zitationsanalyse</i>					X		X									X
<i>Support</i>							X									X
<i>Nutzerprofil</i>					X											
<i>Code review</i>							X									
<i>Transformationspfadanalyse</i>													X			
<i>Berechnungszeit</i>									X							
<i>Benötigte Fähigkeiten/Wissen</i>					X											

3 Datenblattentwicklung

Mögliche Merkmale zur Einordnung und dem Vergleich von Frameworks wurden in Tabelle 1 bereits vorgestellt. Diese wurden in Kapitel 2.4 durch eine umfangreiche Literaturrecherche ermittelt. Im Folgenden wird daraus ein Factsheet bzw. Datenblatt entwickelt, sowie die verwendeten Merkmale erläutert. Außerdem wird darauf eingegangen, warum etwaige Parameter keine Berücksichtigung finden.

3.1 Klassifizierungsmerkmale

Die Merkmale zur Einordnung und dem Vergleich beider Frameworks werden in Tabelle 1 bereits ausgewählt und im Folgenden erläutert.

Die Klassifizierungsmerkmale sind in fünf Kategorien aufgeteilt:

1. Basisinformationen, welche eine generelle Auskunft über das Framework geben. Hierzu zählen die Hauptintention des Entwicklers sowie die spezifischen Zielstellungen, da diese den Aufbau des Programmes bestimmt und die Möglichkeiten zur Simulation vorgibt. Die geographische, zeitliche und sektorale Abdeckung, die das Framework leisten kann, ist für den Anwender ebenso wichtig, da ein Energiesystemmodell stets einen zuvor festgelegten Rahmen spezifisch dieser Vorgaben haben muss.
2. Mathematische Beschreibung, die i.A. wichtig zur Einordnung der Ergebnisse aus der Simulation bzw. Optimierung ist.
3. Die Modellbildung umfasst die Eigenschaften der Frameworks bezüglich zur Erstellung der einzelnen Komponenten eines ES und welchen Umfang das ESM abdecken kann.
4. Die Integration erneuerbarer Energien ist in dieser Ausarbeitung als Einzelkategorie und nicht unter der Kategorie „Modellbildung“ aufgeführt, weil die Modellierung von Energiesystemen mit erneuerbaren Energien mittlerweile den Großteil der entwickelten Modelle ausmacht und diese sich stark von der konventionellen Energieerzeugung unterscheiden. So stellt u.a. die fluktuierende Energieerzeugung eine Herausforderung bei der Modellierung dar, um diese in das Modell integrieren zu können. Deshalb ist es wichtig möglichst detailliert erneuerbare Energien in das ESM integrieren zu können.
5. Die Kollaborativen Eigenschaften spielen für die open source Entwicklung eine große Rolle. Die Parameter, bspw. Offenheit des ESMF oder Einheitlichkeit der Terminologie, ordnen die Eignung für eine mehrparteiliche Entwicklung des ESMF ein.

3.1.1 Basisinformationen

Die Basisinformationen zu den Frameworks sollen einen kurzen Überblick über das Modellierungstool geben. Bei der Auswahl von Frameworks sollte zunächst die **Hauptintention des Entwicklers** bei der Entwicklung des Frameworks und mit der eigenen wissenschaftlichen Fragestellung überprüft werden. Dadurch wird seitens des Entwicklers der Fokus und somit auch die Möglichkeiten, die das ESMF bietet, festgelegt. So führt Van Beek (1999) [29] die Unterscheidung des Ziels eines ESM in Allgemeinen und

spezifischen Fragestellungen ein. Zu den Allgemeinen Zielen werden die Vorhersage zur Zukunft des dargestellten ES, die Erforschung mittels Szenarioanalyse sowie das sog. „*Backcasting*“ eingeordnet. Vorhersagen zur Zukunft des ES sind meist mit Hilfe von Extrapolation historischer Daten aufgestellt und lassen daher nur kurzfristige Beeinflussungen auf ein ESM vom Benutzer analysieren. Die Szenarioanalyse ist der Vergleich von Szenarien mit unterschiedlichem Entwicklungsfokus, bspw. bezüglich technischer oder sozialer Entwicklungen. Die einzelnen Szenarien sollen hierbei nicht die mögliche zukünftige Realität darstellen, sondern bewusst die Extrema bestimmter Entwicklungen aufzeigen, um den „*Worst-Case*“ und „*Best-Case*“ mit dem „*Business as usual*“-Szenario zu vergleichen. Das „*Backcasting*“ ist eine Methode, bei der eine Zukunftsvision geschaffen wird, welche auf die Gegenwart bezogen wird, um mögliche Pfade zu dieser zu erhalten.

Zu den **spezifischen Zielstellungen** eines ESM zählen nach [29] Energiebedarfs-, Versorgungs-, Einfluss- oder Bewertungsmodelle. Energiebedarfsmodelle betrachten den Energiebedarf als eine Funktion von Populations-, Einkommens- und Energiepreisveränderungen. Den Fokus hauptsächlich auf die technologischen Aspekte des ES legt das Versorgungsmodell. Dabei soll untersucht werden, ob die Versorgung für einen zuvor definierten Bedarf ausreicht und kann zusätzlich einen „*least-cost approach*“ [29] zur Entscheidungsfindung nutzen. Bei Einflussmodellen werden die Konsequenzen einer bestimmten Veränderung auf das ES betrachtet, wohingegen Bewertungsmodelle mehrere verschiedene Veränderungen getrennt voneinander am Modell vornehmen und diese Szenarien miteinander vergleichen.

Die **geographische, zeitliche und Sektorenabdeckung** geben den Umfang des Modells an. Die geographische Auflösung kann hierbei stark variieren zwischen der weltweiten bis hin zu einer stark regionalisierten Ebene. Die zeitliche Abdeckung wird in kurz-, mittel- und langfristige Periodendauer unterteilt, wobei jedoch keine Standarddefinition für die genaue Zeitabdeckung der Begriffe existiert. Daher wird durch Grubb, Edmonds et al. (1993) [28] eine grobe Einordnung der zeitlichen Spanne von den Begriffen gegeben: Eine Zeitspanne von weniger als 5 Jahren kann als kurzfristig, 3 bis 15 Jahre als mittelfristig und ab 10 Jahren als langfristiger Zeitrahmen angesehen werden. In einem ESM kann sowohl ein einzelner Sektor separat betrachtet werden als auch ein Multisektoraler Ansatz gewählt werden, welcher sich auf die Wechselwirkungen der wirtschaftlichen Sektoren fokussiert.

Der **Entwicklungsstand**, der **letzte Commit** und **Anzahl der veröffentlichten Versionen** geben Aufschluss darüber wie aktiv das Framework entwickelt wird und können dem Benutzer anzeigen, wie ausgereift das ESMF ist. Darüber hinaus ist so für den Anwender ersichtlich, ob die neuesten bzw. aktuellen Technologien schon implementiert sein könnten.

3.1.2 Mathematische Beschreibung

Die mathematische Beschreibung des Frameworks ist für eine wissenschaftliche Auswertung des ESM und bei der Nutzung des Frameworks unverzichtbar.

Die **Programmierungsumgebung** beeinflusst maßgeblich die Nutzungshürde eines Frameworks. Frei verfügbare und Plattformübergreifende Programmiersprachen wie Python können die Nutzbarkeit eines ESMF durch genannte Eigenschaften anheben. So können in Python sog. „*virtual environments*“ erstellt werden, in welcher die benötigte Python Version und „*Packages*“ installiert werden können,

um das Framework unabhängig von anderen Python Versionen und „*Packages*“ auf dem Rechner des Benutzers laufen lassen zu können [14]. Objekt-orientiertes Programmieren wirkt unterstützend auf den Modellierungsprozess eines ES, indem das Modell von verschiedenen Personen gleichzeitig entwickelt und am Ende mit geringem Aufwand zusammengefügt werden kann.

Ein **Solver** ist ein Programm, welches mathematisch formulierte Probleme lösen kann. „Je nach [...] den Eigenschaften der mathematischen Abhängigkeiten werden unterschiedliche Solver eingesetzt. Eigenschaften der mathematischen Abhängigkeiten sind beispielsweise lineare Optimierungsprobleme [...]“ [42], wie sie in der ES-Modellierung bei "*Bottom-Up*" (BU) - oder auch "*Top-Down*" (TD) -Modellen vorkommen. Die Solver unterscheiden sich in der vergebenen Lizenz und anhand der vorhandenen Features. Die populärsten free open source (FOS) sowie kommerziellen Solver stellen u.a. B. Meindl und M. Templ (2013) [43] sowie Gearhart et al. (2013) vor: GLPK (FOS), CLP (FOS), Ip_solve (FOS), MINOS (FOS), SCIP (FOS), SoPlex (FOS), CPLEX (kommerziell), Xpress (kommerziell) und Gurobi (kommerziell).

Keirstead (2012) [44] merkt an, dass die **Datenverfügbarkeit** eine Herausforderung in der Modellierung von Energiesystemen darstellt. Eingabedaten zu bekommen oder zu erstellen, erfordert umfassende Softwarekenntnisse zu geografischen Informationssystemen, Datenbanken etc. und kann auch mit anderen Forschungsbereichen, z.B. der Meteorologie zur Erstellung von Stromerzeugungsprofilen von Wind- oder PV-Anlagen, verbunden sein [14]. Da die Eingabedaten meist nicht nur von einer Quelle stammen, müssen diese einheitlich sein. ESM mit einer hohen zeitlichen und räumlichen Auflösung erzeugen i.d.R. große Mengen an Ausgabedaten. Auch wenn nur ein Parameter analysiert werden soll, muss dieser in Relation zu anderen Ergebnisparametern und variierenden Eingabeparametern gesetzt werden [14].

Der Unterschied zwischen TD und BU als **analytischer Ansatz** muss näher betrachtet werden, da nach Hourcade (1995) [45] und Grubb (1993) [28] sich große Differenzen in den Ergebnissen dieser beiden Ansätze ergeben.

Dabei ist der TD-Ansatz verbunden mit dem ökonomischen Blick, wohingegen der BU-Ansatz eher mit dem Ingenieursblick in Verbindung gebracht wird [28].

Ein reines ökonomisches Modell besitzt keine spezifische Repräsentation von vorhandenen Technologien, sondern stellt die technologischen Komponenten als Blackbox dar. BU-Modelle hingegen sind unabhängig von dem betrachteten Marktverhalten. Sie beschreiben die Technik, Leistung und direkte Kosten der technologischen Optionen, um mögliche technische Verbesserungen identifizieren zu können.

Eine weitere Charakteristik von TD-Modellen ist, dass sie gesammelte Daten zur Interaktion zwischen den Energiesektoren und anderer Sektoren des Marktes nutzen. Dies geschieht mit Hilfe endogener Variablen, die die Beziehungen so weit wie möglich darstellen sollen. Vergangene Handlungen können mit Hilfe von Extrapolation für die Zukunft abgeschätzt werden und dadurch kurzfristige Vorhersagen treffen.

Im Gegensatz dazu konzentrieren sich BU-Modelle exklusiv auf den Energiesektor und nutzen hoch aufgeschlüsselte Daten, um so den Endenergieverbrauch und technologische Optionen zu beschreiben. Nach Hourcade (1995) [45] können diese detaillierter in beschreibende und vorschreibende Modelle unterteilt werden. Beschreibende Modelle versuchen eine Abschätzung anhand aktueller Entwicklungen von Technologien abzugeben, vorschreibende Modelle hingegen

nutzen ausschließlich die effizientesten Technologien (oder versuchen die expliziten Kosten auf ein Minimum zu reduzieren). Beschreibende Modelle versuchen die Lücke von Ingenieurs- und ökonomischen Ansatz zu füllen, wohingegen die Vorschreibenden Modelle mehr in Richtung Erforschung von Potenzialen hinzielen. Der Hybride Ansatz kombiniert das makroökonomische Modell des TD-Ansatzes und den BU-Ansatz miteinander [30]. Es sollte nach Hourcade et al. (2006) [46] und Bataille (2005) [47] folgende drei Eigenschaften aufweisen: technologische Explizität, mikroökonomischen Realismus und makroökonomische Vollständigkeit. Wenn es um die Integration erneuerbarer Energien in das ESM geht, dann sind sowohl Langzeitänderungen wie in TD-Modellen berücksichtigt, als auch die technologischen Eigenschaften, die von BU-Modellen beschrieben werden, von großer Bedeutung. Aus diesem Grund ist der Hybride Ansatz auch für die Modellierung von ES mit hoher Integration von erneuerbaren Energien sinnvoll [48].

Ommen, Markussen et al. (2014) [49] untersuchen, inwiefern die **mathematischen Ansätze** Lineare, Gemischt-Ganzzahlige und Nicht-lineare Programmierung die Ergebnisse eines ESM signifikant beeinflussen kann. Die Forschungsergebnisse zeigen, dass Gemischt-Ganzzahlige Programmierung der geeignetste Programmieransatz aus Sicht der Genauigkeit der Ergebnisse und der Laufzeit des Programmes ist [49]. Demzufolge ist es von Vorteil, wenn das betrachtete Framework Gemischt-Ganzzahlige Programmierung nutzt. Im Rahmen der Fallstudie in Kapitel 2.3.2 wird detaillierter auf diesen Ansatz eingegangen.

Wichtig für die erwarteten Ergebnisse ist die **zugrundeliegende Methodik** des ESMF. Hierbei wird i.A. unterschieden zwischen der Simulation und der Optimierung. Simulationsmodelle sind nach Definition der World Energy Conference [50] beschreibende Modelle basierend auf einer logischen Repräsentation eines Systems, welche darauf abzielen eine vereinfachte Darstellung der Realität zu erzeugen. Optimierungsmodelle hingegen werden verwendet, um bspw. eine Investmententscheidung oder ein CO₂-Limit innerhalb des Systems von der Eingabe bestimmt, also endogen, zu optimieren.

Netzmodelle zielen auf die Analyse von elektrischen Netzwerken, die ein Teil eines ES darstellen, ab und sind in die vier Arten „*single-node*“- , „*transshipment*“- , „*DC power flow*“- und „*AC power flow*“-Modell zu unterteilen [51–53]. Das „*single-node*“-Modell, auch Kupferplattenmodell genannt, bezeichnet die am einfachsten mögliche Netzstruktur eines ESM. Es wird nach [54] häufig in Modellen mit ökonomischen Ansatz verwendet und stellt eine einfache Verbindung zwischen Knotenpunkten im ESM dar.

Das „*transshipment*“-Modell geht zurück auf ein Jahrhunderte altes Problem der Routenoptimierung von Kaufleuten. Es geht dabei um das Problem der Optimierung von Handelsrouten bezüglich der Kosten bzw. der zurückgelegten Strecke (s. hierzu auch „*Travelling Salesman Problem*“). In der ES Modellierung werden bei diesem Problem Knotenpunkte definiert, zwischen denen ein Austausch an Energieeinheiten stattfinden kann. Dieser Austausch wird durch die Transferkapazität beschränkt und schließlich hinsichtlich der Kosten des Energieaustausches zwischen den Knotenpunkten optimiert. Das Modell beschreibt jedoch nicht die Prinzipien des physikalischen Kraftflusses [54].

Ein detaillierteres Modell zur Optimierung des Kraftflusses zwischen Knotenpunkten eines ESM stellt das Gleichstrommodell bzw. „*linear optimal flow*“-Modell dar. Unter Nutzung des Kirchhoff'schen Gesetzes kann ein aktiver Kraftfluss, welcher auf Widerstand und der maximalen Kapazität der Leitung besteht, optimiert werden [54]. Chatzivasileiadis (2018) gibt jedoch zu bedenken, dass die Bezeichnung

des Modells („*DC Optimal Power Flow*“) nicht mit Gleichstromnetzen oder -leitungen zusammenhängt, sondern es lediglich die Linearisierung der ursprünglichen nicht-linearen „*AC Power Flow*“-Gleichung beschreibt [55]. Auer (2019) spricht in diesem Kontext auch von einem „*Linear Optimal Power Flow*“ (LOPF) [38].

Das nicht-lineare „*AC power flow*“- [54] bzw. „*Optimal Power Flow*“ (OPF)-Modell [38] berücksichtigt neben dem aktiven auch den reaktiven Kraftfluss in elektrischen Netzen, wofür sowohl das kapazitive als auch das induktive Verhalten von Stromleitungen als Gleichung dargestellt wird [54]. Aus diesem Grund ist es als realistischer einzustufen, was jedoch auch mit einer steigenden Komplexität im Vergleich zum LOPF-Modell mit einhergeht. Dieses Modell hat jedoch einen begrenzten Einfluss auf die übliche ES-Modellierung, da es zumeist nur bei sehr kleinen ES, sog. Microgrids, zur Anwendung kommt, da diese die zuvor erklärte Verteilnetzcharakteristik aufzeigen [38].

Die mathematische Zielsetzung des Modelles wird vom Entwickler bestimmt und wird in Form der Zielvariable definiert. Diese Zielvariable ist zumeist die Optimierung der ökonomischen Kosten oder des CO₂- bzw. Treibhausgasausstoßes.

3.1.3 Modellbildung

Bei der Modellbildung werden die Funktionen eines Frameworks dargestellt, welche die Möglichkeiten des Benutzers beim Erstellen des Frameworks aufzeigen. Die **repräsentierten Energiesektoren** in ESM können dabei der Elektrizitäts-, Fernwärme-, individuelle Hauswärme-, Industrie- und Transportsektor sein.

Die **zeitliche Auflösung** ist ein wichtiger Faktor zur Beantwortung der Forschungsfrage durch das ESM. Je höher die zeitliche Auflösung ist, desto detaillierter lassen sich die Fluktuationen von erneuerbaren Energiequellen in dem Modell darstellen [33]. Zusätzlich beeinflusst die Zeitauflösung auch die Berechnungszeit, weswegen flexible bzw. hoch aufgelöste ESM (8760h pro Jahr) erst in den 2000er und 2010er Jahren entwickelt wurden [33]. Wie die zeitliche Auflösung beeinflusst auch die **räumliche Auflösung** die Berechnungszeit maßgeblich. So ist es gewissermaßen ein Tauschhandel zwischen zeitlicher und geographischer Auflösung, weswegen der Trend hin zur Flexibilisierung beider Parameter geht [33].

Das **modulare Design** erhöht die Benutzerfreundlichkeit insofern, als dass mit einem guten modularen Design eine bessere Anwendbarkeit des Frameworks ermöglicht wird. Neuen Nutzern ist es aufgrund dessen möglich Anwendungen mit Hilfe des ausgewählten Moduls zu erschaffen ohne Kenntnis von der gesamten Framework Struktur zu besitzen [14]. Darunter zählen Module, mit welchen Lastprofile erstellt werden können, Energieeinspeisepprofile von fluktuierenden erneuerbaren Energien oder Visualisationspakete zur visuellen Darstellung der Simulationsergebnisse.

Energiesystemmodelle können nach N.M.J.P. van Beek [29] nicht nur anhand des Zweckes unterschieden werden, sondern auch anhand ihrer Struktur, genauer gesagt anhand der Annahmen auf welcher die Struktur basiert. Er nennt hierzu vier voneinander unabhängige Dimensionen, durch derer sich die **Modellstruktur** charakterisieren lässt [29]:

1. Der **Grad der Endogenisierung**, welcher angibt, inwiefern Parameter innerhalb der Modellgleichungen determiniert werden, um die Zahl der vom Benutzer eingegebenen, exogenen, Parameter zu minimieren. Dieser lässt sich darstellen, indem die Zahl der endogenen Variablen sowie die Daten, welche als exogener Input bereitgestellt werden, erfasst werden und eine Sensitivitätsanalyse bzw. „*exogenous shocks evaluation*“ durchgeführt wird [32]. Letzteres kann mit Hilfe der Bewertung der Resilienz des ES gegenüber der „*exogenous shocks*“ (z.B. Finanzkrise, Energiequellen etc.) und der Erholung von diesen erforscht werden; oder durch die Einbeziehung von Entscheidungsparametern für Maßnahmen bei der Abweichung zuvor erwarteter Ziele analysiert werden [32]. Vorhersagemodelle weisen meist einen hohen Grad an Endogenisierung auf, wohingegen erforschende bzw. „backcasting“ Modelle einen niedrigen Grad der Endogenisierung zeigen [29].
2. Je detaillierter die **Beschreibung von Nicht-Energiesektoren** wie Investitionen, Handel etc., desto besser lassen sich durch das ESM die Effekte der Energiepolitik auf die gesamte Wirtschaft nachvollziehen.
3. Der **Detailgrad der Beschreibung der Energieverbraucher** innerhalb des Modells beeinflusst die Qualität der Ergebnisse und die Analyse des technischen Potentials für dessen Energieeffizienz.
4. Zumeist behandeln ESM den Transport der Energie mit Hilfe eines Black-Box-Modelles. Dies führt dazu, dass das Potenzial effizienterer (vgl. zum heutigen Stand) Transporttechnologien nicht erforscht werden kann, weswegen der Umfang der **Beschreibung von Energietransporttechnologien** ein wichtiges Kriterium der Modellstruktur ist.

Der **technologische Detailgrad** kann von Black-Box-Modellen aller technischen Komponenten, zumeist in Top-Down-Modellen zu finden, bis zu einem hohen Detailgrad in Bottom-Up-Modellen reichen. Dieser umfasst sowohl die standardmäßig darstellbaren technischen Komponenten mit Hilfe des Frameworks sowie deren Detailgrad in Form von exogenen Parametern. Die **technischen Modellkomponenten** können unterteilt werden in konventionelle und erneuerbare Erzeuger des Weiteren in Energiespeicher (siehe Kapitel 3.1.4), Übertragungsnetze (siehe Kapitel 3.1.2), Rohstoffe und Verbraucher.

Konventionelle Kraftwerke wie z.B. Thermische Kraftwerke, Kernkraftwerke und Biomassekraftwerke können nach Ringkjøb, Haugan et al. (2018) [36] in unterschiedlichster Weise modelliert werden. So kann bspw. jedes Kraftwerk einzeln oder Kraftwerke eines Typs für eine bestimmte Region als Verbund dieser betrachtet werden.

Während viele Modelle sich ausschließlich auf den Elektrizitätssektor beschränken, beinhalten andere weitere **Energiegüter**. Für ein zukunftsfähiges Energiesystemmodell mit einem hohen Anteil an erneuerbaren Energien sollten in jedem Falle elektrische und thermische Energie sowie Wasserstoff in diesem eingebunden werden können [36]. Endverbraucher in einem Energiesystem teilen sich in den Wärme-, Strom- und Transport-**Sektor** auf, wobei der Stromsektor der am häufigsten repräsentierte Zweig bei der Energiesystemmodellierung ist.

Das **Hinzufügen neuer Komponenten** sollte für Modellierer so einfach wie möglich gemacht werden. Der Grad der Unbestimmtheit in Form von Konsistenz und Eindeutigkeit der Bezeichnungen bestimmt

diesen Vorgang. Wenn die Vollständigkeit von Komponenten eines ES nicht gegeben ist, sollte das Integrieren eigener Modelle von Komponenten dem Nutzer des ESM so weit wie möglich erleichtert werden. Eine weitere Möglichkeit stellt die **Erweiterung eines ESMF mit Hilfe eines Zweiten** dar, um den Umfang der Analysemöglichkeiten des ESM signifikant erhöhen zu können. Darunter zählen Eigenschaften wie die Einheitlichkeit der Terminologie, die Offenheit der Daten und der Entwicklersupport, welche in Kapitel 3.1.5 besprochen werden.

Eine detaillierte **Analysemöglichkeit des Stromnetzes** ist sinnvoll, da ESM sehr häufig nur zur Analyse des Elektrizitätssektors eingesetzt werden. Durch verschiedene ESM werden, nach Analyse der OMI [56], Übertragungs- und Verteilungsnetze, Gleich- und Wechselstromlastfluss sowie die Nettoübertragungskapazitäten analysiert. Die Schwierigkeiten bei der Netzanalyse werden in Kapitel 3.1.2 unter dem Punkt Netzmodelle behandelt.

3.1.4 Durchdringung erneuerbarer Energien

Die Energieerzeugung durch erneuerbare Energien sollte in der ESM eine große Rolle spielen, da durch die Einhaltung der Klimaziele, diese in der Energieplanung unabdingbar sind. Für Modellierer ist es jedoch eine Herausforderung die un stetigen Energiequellen in die Modelle einzubringen, da durch die Abhängigkeit zu meteorologischen Gegebenheiten, eine Unsicherheit in der zur Verfügung stehenden Leistung besteht. Diese **Energieerzeugung fluktuierender erneuerbarer Energien** kann mit Hilfe standortbezogener meteorologischer Daten (z.B. Windgeschwindigkeitsdaten) oder stochastischer Methoden (z.B. stochastische Zufluss Modellierung zur Planung von Wasserkraftwerken) als Zeitreihe modelliert werden bzw. durch historische Messergebnisse ebenfalls als Zeitreihe in das Modell eingebracht werden [36]. Zu den erneuerbaren Energietechnologien zählen die fluktuierenden Energiequellen Wind-, Sonnen- (PV, Solarthermie, Sonnenwärmekraftwerke) und Wasserkraft (Stausee oder Fluss) sowie die nicht-fluktuierenden Energiequellen Geothermie, Wellen- und Gezeitenkraftwerke.

3.1.5 Kollaborative Eigenschaften

Wie in Kapitel 2.2 bereits ausgeführt ist die **Offenheit von ESMF** eine essenzielle Eigenschaft im Hinblick auf die Energiewende und wissenschaftliche Arbeit. Diese wird zunächst mit der vorhandenen **Lizenz** ausgedrückt. Im Allgemeinen lassen sich Lizenzen in permissive und nicht-permissive aufteilen, wobei es sich um den sog. „Copyleft“-Effekt handelt. *Copyleft* bedeutet, dass „[s]ämtlicher Quellcode, der einem GPL-lizenzierten Programm hinzugefügt wird, [...] Freie Software sein [muss], auch wenn er in einer anderen Datei steht“ [57]. Letztlich gibt die Auswahl einer Lizenz den Ausschlag darüber aus, wer was mit der Lizenz unter welcher Auflage machen darf und was in Zukunft mit dem Programm passiert [58].

Ein Open Source Programm dient in den meisten Fällen dazu, dass Entwickler den Code frei verwenden und ggf. gemeinsam weiterentwickeln können. Hierzu ist bspw. die **terminologische Einheitlichkeit** von großer Bedeutung, da so eine effektive Kommunikation zwischen den Entwicklern mit unterschiedlichen Hintergrundperspektiven und -wissen sichergestellt werden kann [14].

Für eine wissenschaftliche Nutzung des Frameworks ist die Einhaltung der **wissenschaftlichen Standards** unabdingbar. Die Einhaltung dieser beinhaltet die Aspekte Transparenz, Wiederholbarkeit,

Reproduzierbarkeit und Überprüfbarkeit [14], welche dazu beitragen die Wissenschaft voranzutreiben und damit Kurskorrekturen mit Hilfe von unabhängigen Überprüfungen durchgeführt werden können [59]. Zu der Überprüfbarkeit zählt u.a. eine detaillierte Dokumentation des Programmes, welche zusätzlich zur besseren Verständlichkeit beiträgt. Ebenfalls tragen Anwendungsbeispiele, die in der Dokumentation oder in dem *GIT-Repository* des jeweiligen Frameworks hinterlegt werden, zur besseren Verständlichkeit bei.

In dem Artikel der Universität Innsbruck über die Zitationsanalyse wird angemerkt, dass „[d]ie Zitierungshäufigkeit [...] als Maßzahl für die Rezeption der wissenschaftlichen Arbeiten eines Autors/einer Autorin [gilt]“ [60]. Die **Zitationsanalyse** umfasst neben der Zitierungshäufigkeit u.a. auch die Abfrage der zitierenden Arbeiten. Diese soll bezwecken, dass weitere Anwendungsbeispiele gefunden werden oder Kooperationen zu demselben Forschungsthema entstehen können. Der **Entwicklersupport** lässt sich mit Hilfe von Merkmalen wie der Möglichkeit den Entwicklern in einem Forum o.ä. Fragen zum Code stellen bzw. von Bugs berichten zu können. Aber auch der Umfang der Dokumentation, die den Benutzer im Modellierungsprozess unterstützen soll, ist ein Merkmal des Entwicklersupports.

3.2 Auswahl der Parameter

Die Auswahl der Klassifizierungsparameter erfolgte anhand der in Kapitel 2 aus der Literatur herausgearbeiteten Merkmale. Die ausgewählten Parameter aus Tabelle 1 lassen sich spezifisch auf Frameworks anwenden und charakterisieren das ESMF. Die Unterpunkte der „Basisinformationen“, die nicht in dem Factsheet integriert wurden, treffen keine Aussage zu der Qualität des analysierten ESMF und wurden aus diesem Grund nicht ausgewählt. Einzig die „Anzahl der Downloads/Verkäufe“ könnte die Rezeption der Software darstellen und somit einen Mehrwert für das Datenblatt sein. Da diese sich jedoch bei FOSS, welche in den meisten Fällen mit GIT zur Verfügung gestellt wird, nicht verlässlich bestimmen lässt bzw. diese Daten nicht zugänglich sind, wird auf diese Informationen in diesem Datenblatt verzichtet und stattdessen in dem Unterpunkt „Zitationsanalyse“ behandelt. Diese Information könnte jedoch, falls eine Möglichkeit zum Einsehen dieser Daten besteht, in Zukunft in das Datenblatt aufgenommen werden, um die Information bezüglich der Rezeption des Frameworks zu verbessern, da diese eine gute Aussage bezüglich der Anwendbarkeit bzw. Benutzerfreundlichkeit geben kann. Ob das ESMF eine Benutzeroberfläche (Abk. engl. „GUI“) besitzt, ist bei FOSS auszuschließen, da diese für gewöhnlich keine „GUI“ besitzen und dieser Punkt somit zur Klassifizierung eines ESMF überflüssig ist. Die nicht ausgewählten Parameter des Unterpunktes „Modellbildung“ beziehen sich in der Literatur auf ESM und keine ESMF, sodass diese Parameter ebenfalls unpassend zur Klassifizierung dessen sind (siehe zum Unterschied zw. ESM und ESMF Kapitel 2.1). Das „Nutzerprofil“ ergibt sich aus dem Unterpunkt „Modellzweck“, welcher in den „Basisinformationen“ hinterlegt ist. Der „Code Review“ sowie die „Transformationspfadanalyse“ sind zu spezifisch für ein Datenblatt, welches der Anwender zunächst nur benutzen sollte, um einen generellen Überblick über das ESMF zu bekommen. Die „Berechnungszeit“ hängt stark von der verwendeten Hardware des Anwenders und dem Detailgrad des ESM ab, sodass dieser Punkt nur geringe Aussagekraft für den individuellen Nutzer hat. Der Unterpunkt „benötigte

Fähigkeiten/Wissen“ geht aus den „Basisinformationen“ bzw. „Mathematischen Beschreibung“ hervor und stellt somit keinen Mehrwert für das Datenblatt dar.

In der folgenden Tabelle 2 wird das generische Datenblatt, welches zur Klassifizierung und dem Vergleich von ESMF dient, bereitgestellt.

Tabelle 2: generisches Datenblatt

Klassifizierungsparameter	[Name des Frameworks]
Basisinformationen	
Entwickler	
Veröffentlichungsdatum	
Modellzweck	
Genereller Zweck	
Spezifischer Zweck	
Geographische Abdeckung	
Sektorenabdeckung	
Zeithorizont	
Entwicklungsstand (Version)	
Letzter Commit (Aktualität)	
Anzahl der veröffentlichten Versionen	
Dokumentation vorhanden?	
Mathematische Beschreibung	
Programmierungsumgebung	
Frei verfügbar?	
Unterstützte Betriebssysteme	
Objekt-orientiert?	
Unterstützte externe Solver	
Daten	
Eingabedatenformate	
Eingabedatenart	
Ergebnisdatenformate	
Analytischer Ansatz	
Mathematischer Ansatz	
Methodik	
Optimierungsbeschränkung/ Netzmodelle	
Dynamikentscheidung bei der Optimierung	
Zielsetzung (-Funktion)	
Modellbildung	
Repräsentierte Energiesektoren	
Zeitliche Auflösung	
Räumliche Auflösung	
Zeitschritte	
Modularität	
(Lastprofile erstellen)	
Modellstruktur	
Grad der Endogenisierung	
Beschreibung von Nicht-Energiesektoren	
Beschreibung der Endverbraucher	

<i>Beschreibung der Transporttechnologien</i>	
Technologischer Detailgrad	
<i>Standardmäßige Darstellbare technische Komponenten</i>	
<i>Detailgrad der techn. Komponenten</i>	
<i>Anzahl exogener Parameter</i>	
Erweiterbar?	
<i>Elektrische Netzwerkabdeckung</i>	
Durchdringung erneuerbarer Energien	
<i>Erneuerbare Energien-Formate</i>	
<i>Erneuerbare Energieeinspeisungszeitreihen</i>	
<i>Darstellbare Anlagen zur erneuerbaren Energieerzeugung</i>	
<i>Emissionen</i>	
Kollaborative Eigenschaften	
<i>Verfügbarkeit</i>	
<i>Offenheit der Daten</i>	
<i>Lizenz</i>	
<i>Entwicklerperspektivenspektrum</i>	
<i>Einheitlichkeit der Terminologie</i>	
<i>Wissenschaftliche Standards</i>	
<i>Anzahl Beispiel Scripts in GitHub-Repository</i>	
Zitationsanalyse	
<i>Zitierhäufigkeit</i>	
<i>Zitierende Arbeiten</i>	
Entwicklersupport	

4 Verwendete Software

Dieses Kapitel stellt die in dieser Arbeit verwendete Software zur Erstellung der Fallstudie vor und geht auf die Struktur der jeweiligen Programme ein. Die Vorstellung der verwendeten Software trägt zu der Transparenz und Nachvollziehbarkeit der Fallstudie bei und ist somit wichtig, um den Kriterien der „Open Science“-Bewegung, der vollständigen Transparenz, gerecht zu werden.

4.1 Open Energy Modelling Framework (*oemof*)

Das "open energy system modelling framework" (*oemof*) basiert auf der Programmiersprache Python und ermöglicht dem Benutzer die Modellierung und Optimierung von Energiesystemen. Die erste Version wurde 2015 erstmals veröffentlicht und entstand aus einer Zusammenarbeit des Zentrums für Nachhaltige Energiesysteme (ZNES) Flensburg und dem Reiner Lemoine Institut (RLI), sowie der Otto-von-Guericke-Universität von Magdeburg (OvGUM). Der Ansatz der Entwickler war von "Anfang an [das Framework] kollaborativ" [61] zu gestalten, was es auch anderen Entwicklern ermöglicht an dem Projekt via GIT-Hub [62] mitzuarbeiten. Sie bilden in der GIT-Hub-Repository die sog. "developer group".

Pythons Eigenschaft der Objekt-orientierten Programmierung ermöglicht eine modulare Aufteilung des Codes. Durch diese modulare Aufteilung lässt sich die Software auch bei sektorübergreifenden Studien verwenden. Oemof teilt sich dabei in 15 sogenannte Repositories auf (dt.: Depots, Lager, Quellen), von denen die wichtigsten *oemof-solph*, *tespy*, *feedinlib*, *demandlib* und *oemof-thermal* sind.

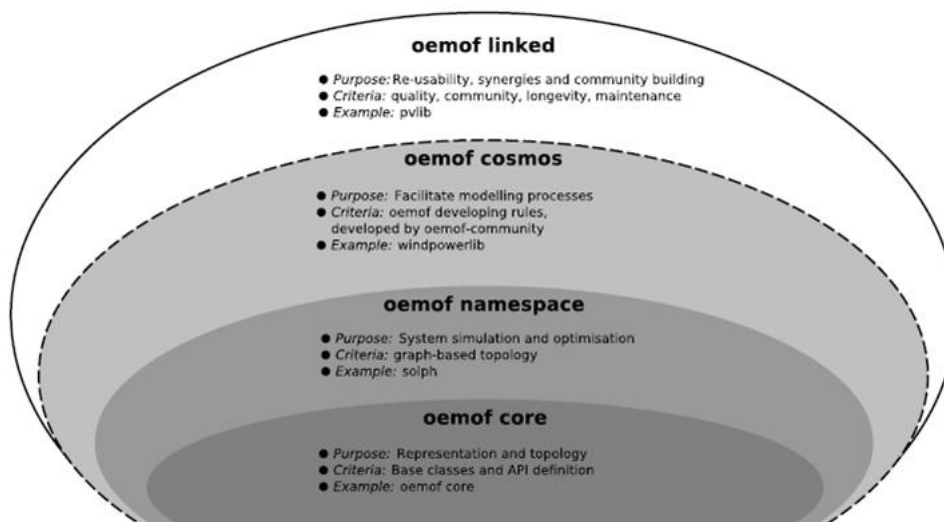


Abbildung 2: Struktur der oemof-Architektur [7].

4.1.1 Projekt-Architektur

Das Framework besteht aus drei Ebenen und, wie in Abbildung 2 (grau hinterlegt) zu sehen, und der mit oemof verbundenen Funktionen in *oemof linked*.

Die erste Ebene stellt die Kernebene (*oemof core*) dar. In dieser werden die physischen Komponenten eines ES in Klassen und mit individuellen spezifizierenden Parametern definiert. In der Graphentheorie sind alle physischen Komponenten eines ES als Knotenpunkte untereinander durch Kanten verbunden. Bei dem Graphen, der das ES modelliert, handelt es sich „[...] um einen bipartiten Graphen, da Kanten nur zwischen Knoten unterschiedlichen Typs existieren dürfen“ [63]. Ein Knoten weist zwei verschiedene Typen auf: Zum einen die Komponente (*component*), welche Erzeuger und Verbraucher modelliert, zum anderen die Sammelschiene, dem sog. *Bus*, an welcher die Komponenten angeschlossen sind. Der *Bus* benötigt immer sowohl mindestens einen Ein- als auch Ausgangsknoten, wohingegen die Komponente nur Ein-, nur Aus- oder Ein- **und** Ausgänge besitzen kann [63]. Wie in Abbildung 3 zu sehen, sind die Komponenten nochmals unterteilt in Senken, Quellen und Transformer:

- Senken besitzen ausschließlich eingehende, jedoch keine ausgehenden Ströme und repräsentieren Verbraucher im ES.
- Quellen haben gegenteilig zu Senken ausschließlich ausgehende Ströme. Als Beispiele sind Wind- und Solarkraftanlagen, sowie primäre Energieträger wie Kohle, Gas und Öl zu nennen.
- Transformer weisen sowohl ein-, als auch ausgehende Ströme auf, wie z.B. Kohlekraftwerke, welche von einem Lieferanten durch den *coal bus* Kohle beziehen und die chemisch gebundene Energie in elektrische Energie umwandeln und in den *electricity bus* einspeisen. Die Relation der Ströme kann in Form von Parametern, bspw. Dem Wirkungsgrad oder nominalen Leistung des Kraftwerkes, spezifiziert werden.

Die Kanten des Graphen repräsentieren einerseits die strukturelle, als auch die quantitative Verbindung von Knotenpunkten.

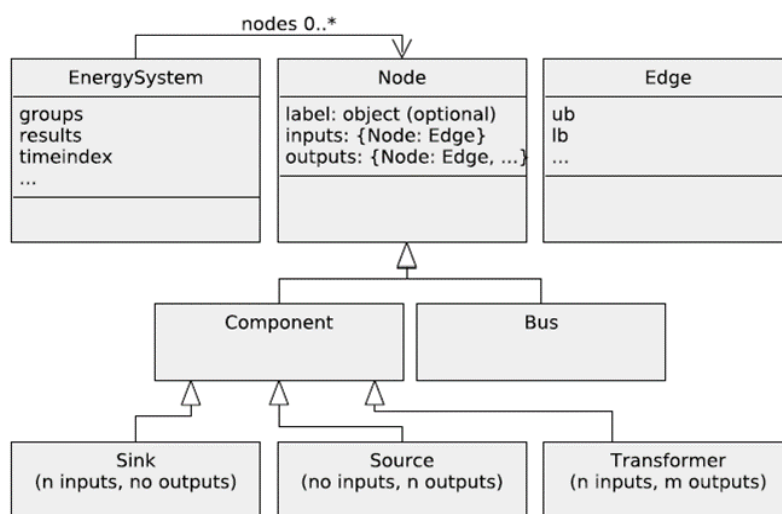


Abbildung 3: Klassen im oemof-Quellcode [7].

Alle diese Komponenten eines ES und ihre Verbindung durch *Busse* werden individuell zu der ES-Klasse hinzugefügt, wobei die zeitliche, räumliche Auflösung und zusätzliche Konfigurationen spezifiziert werden.

Ein schematisches Beispiel eines ES in *oemof* wird in Abbildung 4 repräsentiert:

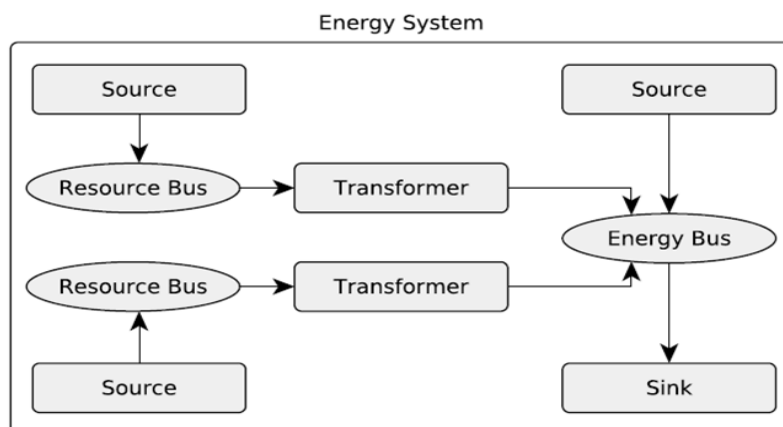


Abbildung 4: Beispiel eines Energiesystems in *oemof* [7].

Wenn der Graph des ESM durch die hinzugefügten Knoten und Ecken definiert ist, werden lineare Gleichungen, welche zu jeder Komponente erstellt werden, in das passende Format mit dem *solph*-Paket umgewandelt, um die Daten für den Solver lesbar zu machen und somit das ES zu optimieren.

Die zweite Ebene des *oemof*-Projektes ist die Namensraum-Ebene (*oemof namespace*). Der Namensraum ist dabei die Bezeichnung - besonders häufig bei objektorientierter Programmierung - für den Vorgang, bei dem Namen für die Objekte in einer Art Baumstruktur angeordnet und über entsprechende Pfadnamen eindeutig angesprochen werden. Jeder Name bezeichnet eindeutig das zugehörige Objekt. In einem anderen Namensraum kann der gleiche Name jedoch zur Bezeichnung eines anderen Objektes genutzt werden.

Die Namensraum-Ebene in *oemof* ermöglicht eine Erweiterung von Funktionalitäten basierend auf den Paketen der Kernebene. Diese Erweiterung ermöglicht verschiedene Modellierungsansätze durch *oemof*, basierend auf denselben Energiekomponenten der Kernebene.

Die dritte Ebene ist die Umgebungs-Ebene (*oemof cosmos*). Diese Ebene enthält nützliche Funktionen zur ESM und ist Teil der Standardentwicklungsrichtlinien des Frameworks. Beispiele hierfür sind die *Feedinlib*, welche Zeitreihen für die Einspeisung von volatilen Kraftwerken enthalten, und *demandlib*, durch welche Zeitreihen für die Last auf Grundlage des Gesamtjahresbedarfes von Strom oder Wärme erstellt werden können.

4.1.2 Optimierung in *oemof*

Der wohl gängigste Nutzungsfall von ESM ist die Optimierung des ES. Hierbei erlaubt die *solph*-Bibliothek einen gemischt-ganzzahligen linearen (engl.: *Mixed-Integer linear Programming (MILP)*) oder linearen Optimierungsansatz.

Bei der linearen Optimierung des ES werden i.A. die totalen Kosten für den simulierten Zeitrahmen minimiert. Die Berechnungsterme enthalten sowohl zeitabhängige als auch zeitunabhängige Variablen. Bei der gemischt ganzzahligen Optimierung fließen zusätzlich zu den o.g. Variablen, nicht-konvexe Parameter wie bspw. die minimale Stillstands Zeit eines Kraftwerkes oder die mit dem Herunterfahren des Kraftwerkes verbundenen Kosten ein.

Das *oemof* Optimierungsmodell wird an ein Modell des *pyomo*-Paketes (Solver-Bibliothek implementiert in Python) vererbt, um von diesem die Optimierung mittels Solver vornehmen zu lassen.

4.2 Python for Power System Analysis (PyPSA)

Python for Power System Analysis (PyPSA) wurde von dem *Frankfurt Institute for Advanced Studies (FIAS)* entwickelt. Die Software gehört zu einer von vielen Free Open Software Frameworks (FOSF) zur Energiesystemmodellentwicklung und stellt eine große Bandbreite an Werkzeugen für den Benutzer zur Verfügung. Es beinhaltet Modelle für Konventionelle Kraftwerke, erneuerbare Energieerzeugung, Energiespeichereinheiten, aber auch Kopplung der Energiesektoren und Gleich- und Wechselstrom-Übertragungsleitungen.

Die Entwickler erklären die Notwendigkeit des Frameworks, trotz zahlreicher vorhandener Energiesystemmodelle, dass diese zumeist vor der Ära der erneuerbaren Energieerzeugung und Elektrifizierung des Verkehrs- und Heizsektors entwickelt wurden und daher auf die heutige Zeit nur schwer anwendbar sind [2].

PyPSA berücksichtigt in der Modellierung fluktuierende Energieerzeugung durch erneuerbare, dafür notwendige Energiespeicher und Sektorenkopplung, indem es mit großen Netzwerkstrukturen umgehen kann und sich einfach erweitern sowie den Anwender lange Zeitreihen implementieren lässt. Es soll mit Hilfe der Möglichkeit zur Modellierung der Anwendungen und Erweiterungsoptimierung des ES die Lücke zwischen ES Analyse Software und ES Modellierungswerkzeugen schließen.

Zudem kann *PyPSA* mit Gesamtlastberechnungen sowie linearisierten optimalen Ladungsströmen unter zuvor definierten Beschränkungen, z.B. einem Kostenlimit, umgehen.

4.2.1 Komponenten

Busse sind, wie in *oemof*, die fundamentalen Knotenpunkte, mit welchen alle anderen Komponenten verbunden sind. Ihre mathematische Rolle ist es die Energieerhaltung, also die Summe aller Ein- und Ausgänge, bei jedem Zeitschritt gleich null aufrecht zu erhalten (s. Kirchoff'sches Gesetz). Lasten, Generatoren, Speichereinheiten (unterteilt in *Stores* und *Storage Units*), *Shuntimpedances* und *Links* können mit einem *Bus* verbunden werden und bestimmen die Energiebilanz an diesem:

- Lasten repräsentieren einen festgelegten Energiebedarf,
- die Einspeisung eines Generators kann innerhalb seiner nominellen Leistung optimiert werden,
- *Stores* können die Energie von einem Zeitschritt zum nächsten mit einem Standardverlustwirkungsgrad verlagern,
- *Storage Units* verhalten sich i.A. wie "*Stores*", können darüber hinaus jedoch auch Wirkungsgradverluste und Leistungslimitierungen beim Be- und Entladen darstellen.

- *Shuntimpedances* besitzen einen spannungsabhängigen Energiebedarf.
- Leitungen und Transformatoren verbinden zwei Busse mit einer festgelegten Impedanz. Energie fließt durch Leitungen und Transformatoren entsprechend dem Ungleichgewicht der Energie an den Bussen und der Impedanz im Netzwerk. Die Impedanzen sind als "passive Zweige" vermerkt, um sie von den kontrollierbaren Zweigen des Netzwerkes auseinander halten zu können.
- *Links* verbinden zwei Busse mit einem kontrollierbaren Leistungsausfluss und können vom Benutzer definiert oder von *PyPSA* optimiert werden. Sie stellen Hochspannungsleitungen, Import-Export-Kapazitäten in Transportmodellen sowie die Übertragungsnetzkapazität oder Energieumwandlungsprozesse (Wärmepumpen, Gasboiler, etc.) mit einem definierten Wirkungsgrad dar. Hierbei kann der Wirkungsgrad auch variieren, wie bei Wärmepumpen, bei denen die Leistungsziffer von der Raumtemperatur abhängt.

Allgemein lassen sich die Komponenten bei *PyPSA* in die Kernkomponenten (*Store*, *Bus* und *Link*) und erweiterte Komponenten (wie z.B. Generatoren, *Storage Units*, BHKWs, etc.) einteilen, welche sich aus den Kernkomponenten zusammensetzen lassen. Ein schematisches Beispiel für die Struktur eines ES in *PyPSA* liefert Abbildung 5.

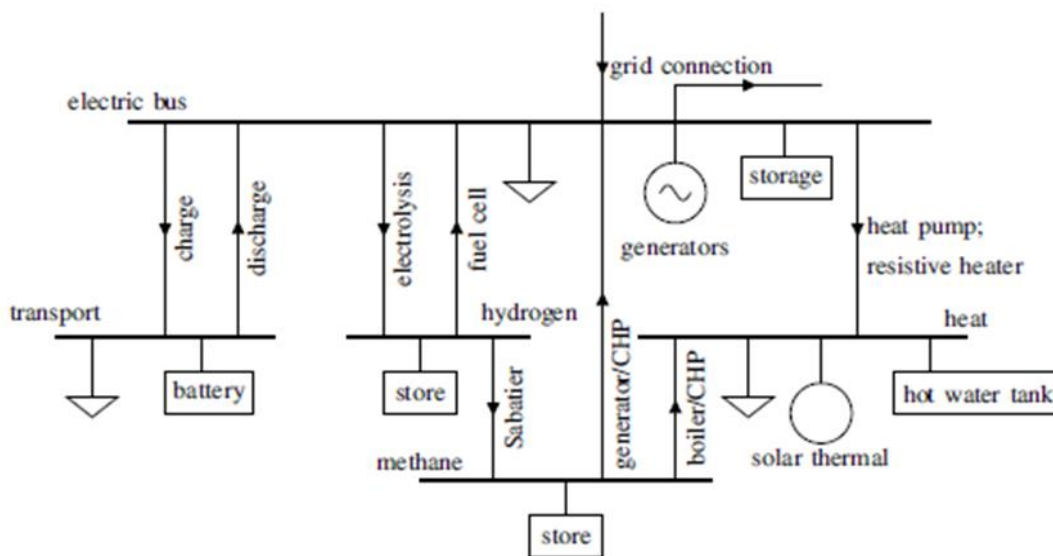


Abbildung 5: Beispiel eines Energiesystems in *PyPSA* [2] - CHP Combined Heat and Power Plant

PyPSA bietet ebenfalls die Möglichkeit Betrieb und Investment anderer Energiesektoren, wie Erdgas, Wärme oder Transport, zu optimieren. Diese Sektoren können mit Hilfe eines Netzwerkes mit Verbindungen, die mit Wirkungsgraden für Energieumwandlungsprozesse versehen sind, modelliert werden.

4.2.2 Simulation ohne Optimierung

Innerhalb der Energieflussberechnung spezifiziert der Benutzer die Energiebereitstellung aller Komponenten, die dem System Energie liefern. Daraufhin berechnet *PyPSA* die resultierende

Spannung im Netzwerk und dadurch die Energieflüsse in den passiven Zweigen, basierend auf deren Impedanz.

Dabei verfügt *PyPSA* über Energieflussgleichungen für (linearisierte) Gleich- und Wechselstromnetzwerke. Folgende Funktionen können hierbei verwendet werden:

- `Network.pf` = vollständig nicht-linearer Energiefluss im Netzwerk
- `Network.lpf` = Linearer Energiefluss im Netzwerk

4.2.3 Simulation mit Optimierung

PyPSA ist ein partielles Gleichgewichtsmodell, welches mit Hilfe von linearen Energieflussgleichungen sowohl den Kurzzeitbetrieb als auch Langzeitinvestments in einem ES als ein lineares Problem optimieren kann. Dabei minimiert *PyPSA* die Gesamtsystemkosten, welche variable und Fixkosten der Erzeugung, Speicherung und Übertragung, gegebener technischer und physischer Vorgaben sein können.

Die Optimierung basiert, wie bei *oemof*, auf der für Python entwickelten Optimierungsmodellierungssprache Pyomo. Dies stellt eine Kompatibilität mit anderen ESMF her, die ebenfalls Pyomo zur Optimierung nutzen, wie z.B. *oemof*, *calliope* oder *urbs* [2].

Zur Optimierung des Netzwerkmodelles bietet *PyPSA* die Funktion `Network.lopf` (= Linear optimierter Energiefluss für eine Zeitreihe).

4.3 Transforming Energy Simulation Systems Framework (*tessif*)

Das „*Transforming Energy Simulation Systems Framework*“ (*tessif*) ist ein OSF zur Energiesystemmodellierung, welches auf der Grundlage der Programmiersprache Python entwickelt wird und zurzeit die Frameworks *oemof* und *PyPSA* unterstützt. Das Ziel ist es eine einheitliche Benutzeroberfläche für sämtliche Frameworks zu erschaffen mit besonderem Fokus auf die Anwendung durch Ingenieure. Diese sollen durch die einfache Handhabung und Modellierung angesprochen werden, auch ohne umfassende Programmierkenntnisse zu besitzen.

Zur ESM gibt es viele OSF, die auf Python basieren. Sie werden zumeist unabhängig voneinander entwickelt und überschneiden sich daher teilweise in ihren Funktionen. Sie konzentrieren sich jedoch auch auf verschiedene Aspekte eines Energiesystems und könnten durch das vorhandene Synergiepotential durch die Einbindung in *tessif* ein umfangreicheres Modell erstellen und so fundiertere Ergebnisse bei der Analyse von Energiesystemen liefern, als es ein alleinstehendes Modell schaffen könnte.

Die Hauptprobleme können mit diesem Projekt nicht gelöst werden, wenn mehr als eines dieser Programme gleichzeitig zur Energiesystemanalyse genutzt wird: Es gibt keine einheitliche Dateneingabe oder -ausgabe der in Python implementierten OSF. Diese einheitliche Dateneingabe stellt *tessif* durch die Benutzeroberfläche zur Verfügung, indem es eine eigene Funktion zur Netzwerkerstellung besitzt, und dieses durch die *Transform*-Bibliothek in die gewünschte ESM-Struktur umwandelt.

4.3.1 Struktur

Das Framework *tessif* setzt sich aus 4 Hauptfunktionalitäten zusammen:

1. *Parse/Write*: Dieses Cluster dient zur Dateneingabe und erlaubt es dem Anwender sämtliche Datenformate zur Eingabe zu nutzen. So können neben „*hard-coded*“ Python Objects auch Dateien im .cfg-, .json-, .sdp-, .xlsx-, .xml- und .yaml-Format eingelesen werden.
2. *Simulate*: Das Modul *simulate* ermöglicht dem Benutzer eine vereinfachte Energiesystemsimulation zu erstellen. Dabei kann sowohl auf die *oemof*-, als auch auf die *PyPSA* und *tessif*-Bibliothek zurückgegriffen werden.
3. *Transform*: Die eingelesenen Energiesystemmodelle werden zu einem einheitlichen Datentyp *collections.Mapping* transformiert. Somit können Energiesysteme mit einem Framework erstellt werden und mit einem anderen simuliert bzw. optimiert oder gemappt werden, stark vereinfachte Energiesysteme erstellt oder aus diesen wieder in Energiesysteme umgewandelt werden.
4. *Visualize*: Um die erzeugten Daten analysieren zu können wird zumeist auf Graphen zurückgegriffen, welche mit diesem Modul geringen erzeugt werden können. So kann durch die Funktion *nxgraph*, ein Clusterdiagramm erzeugt werden, um die Verbindungen zwischen den Knotenpunkten wie Erzeugern, Quellen, Senken und *Bussen* zu zeigen. Eine weitere Visualisierungsmethode stellt das Modul *loads* dar, welches ein Balkendiagramm zu den Energieströmen, die zwischen zwei Komponenten erfolgen, erstellt sowie *ldc*, welches Liniendiagramme zu den erzeugten Daten plottet. Das Modul *compare* wird zum Vergleich mehrere Datenreihen von Energieströmen benutzt. Außerdem bietet *components* die Funktion, mehrere Sub Plots in einem Plot darzustellen, um den Verlauf der erzeugten, transportierten oder simulierten Energieflüsse von einzelnen Komponenten vergleichen zu können.

4.4 Atlite

Atlite [64] ist eine frei verfügbare auf Python basierende Software, welche auf Grundlage des *RE Atlas* der *Aarhus University* als eine im Umfang reduzierte Version von der *Renewable Energy Group* an der *FIAS* entwickelt wurde. Mit dieser Software können Wetterdaten wie Windgeschwindigkeit, Solare Strahlungsintensität, Temperatur etc. in Energiesystemdaten (Windenergie, Solarenergie, Wasserkraft) umgewandelt werden. Die Raum- und Zeitauflösung ist abhängig vom Wetterdatensatz, welches dafür herangezogen wird. Neben Wind-, PV- und Solarthermischer Energie können auch Zeitreihen zur Wasserkrafterzeugung und der Wärmelast erstellt werden.

Atlite bezieht zunächst Daten von Globalen Wetterdatenbanken wie *ERA5* [65], *NCEP* [66–68] oder *SARAH* [69, 70] und lässt Nutzer des Tools einen sog. *cutout* erstellen, mit welchem die räumlichen und zeitlichen Grenzen festgelegt werden können und die Wahl getroffen werden kann, welcher Wetterdatensatz bezogen werden soll (siehe Code 1). Die räumliche Auflösung der hier verwendeten Wetterdatenbank (*ERA5*) beträgt 9 x 9 km.

```

cutout = atlite.Cutout(path="../data/HH-2019_origshape.nc",
                      module="era5",
                      x=slice(x1- .2, x2+ .2),
                      y=slice(y1- .2, y2+ .2),
                      time="2019"
                      )

```

Code 1: Funktion zur Erstellung des Cutout in atlite

Diese wurden bei der Erzeugung der Daten für das Hamburger Energiesystem aus den Datenbanken *LAU_regions_2016* und *NUTS_3_regions_2016* für die räumliche Begrenzung gezogen. In diesen Datenbanken sind die Profile der einzelnen Regionen in Europa als Array von Koordinaten, die die Begrenzung der Region repräsentieren, hinterlegt. Mit diesen wird die Form des *cutout* bestimmt.

```

def renewable_df(renewable_type, columns, ):
    renewable_layout = capacity_layout(cutout, typ=renewable_type)

    if renewable_type == 'wind_onshore':
        ren = cutout.wind(turbine='Vestas_V112_3MW',
                          layout=renewable_layout)

    else:
        ren = cutout.pv(panel='CSi', orientation='latitude_optimal',
                        layout=renewable_layout)

    df = pd.DataFrame(ren)
    df['time'] = index_time
    df.set_index('time', inplace=True)
    df.columns = [columns]

    return df

```

```

pv = renewable_df('pv', "PV 2019 (MW)")
won = renewable_df('wind_onshore', "Wind 2019 (MW)")

```

Code 2: Beispielfunktion zur Erstellung der Zeitreihe zur Energieerzeugung von Wind- und PV-Anlagen

Nun kann der Benutzer die Standorte, Typen und dementsprechend technische Daten von Energieerzeugungsanlagen, also Windräder und PV-Zellen in dem *cutout* festlegen, woraus *atlite* die Energieerzeugung für jede Zelle berechnet. Die Standortdaten von Anlagen zur erneuerbaren Energieerzeugung in Deutschland sind bspw. in dem Markstammdatenregister (MaStR) der Bundesnetzagentur [71] zu finden.

5 Fallstudie

In diesem Kapitel werden zunächst die beiden Frameworks mit Hilfe des in Kapitel 3 entwickelten Factsheets bzw. Datenblattes verglichen und die Ergebnisse aus diesem Vergleich erläutert. Anschließend folgt eine Beschreibung des Aufbaus der ESM in *oemof* und *PyPSA*, sowie die Beschreibung der Datenquellen und Szenarien des praktischen Versuches. Am Ende des Kapitels erfolgt die Analyse der aus den Szenarien der ESM ermittelten Daten. Dabei werden die simulierten Ergebnisse aus den Optimierungen der Szenarien mittels *oemof* und *PyPSA* gegenübergestellt. Zusätzlich werden noch weitere Bewertungskriterien bezüglich Handhabbarkeit der Simulationstools besprochen.

5.1 Der Vergleich von *oemof* und *PyPSA* anhand eines Factsheets

Tabelle 3 vergleicht *PyPSA* und *oemof* anhand der in Kapitel ... erläuterten Parameter zur Klassifizierung von Modelling Frameworks. Hierbei wurden die Informationen von verschiedenen Quellen und eigenen Beobachtungen zusammengetragen. Aus den Parametern wurde ein sogenanntes Factsheet entwickelt, das einen schnellen und umfangreichen Überblick schaffen soll, damit der Nutzer das für seine Zwecke geeignetste Framework bzw. Modell auswählen kann. Die Parameter sind aufgeteilt in quantitative und qualitative Merkmale. Die Abkürzungen der quantitativen Merkmale sind dem Abkürzungsverzeichnis sowie der Tabellenüberschrift zu entnehmen. Die qualitativen Merkmale orientieren sich an der subjektiven Wahrnehmung und umfassen die drei Skalenwerte „Niedrig“, „Mittel“ und „Hoch“.

Tabelle 3: Vergleich von oemof und PyPSA anhand eines Factsheets

Abkürzungen: KIT - Karlsruhe Institute of Technology, FIAS – Frankfurt Institute for Advanced Studies, RLI – Reiner Lemoine Institut, HSF – Hochschule Flensburg, DLR – Deutsche Luft- und Raumfahrtbehörde, EUF – Europa Universität Flensburg, OvGUM – Otto-von-Guericke-Universität Magdeburg, V – Vorhersage, E – Erforschung, R – Rückübertragung, EB – Energiebedarf, EV – Energieversorgung, EI - Einflüsse, B – Bewertung, MA – Modularer Aufbau, BD – Benutzerdefiniert, KZ – Kurzzeit, LZ – Langzeit, QL – Qualitativ, QN – Quantitativ, MON – Monetär, AU – Aufgeschlüsselt, TD – Top-Down, BU – Bottom-Up, HYB – Hybrid, Ö – Ökonomie, MÖ – Makroökonomie, ÖGG – Ökonomisches Gleichgewicht, O – Optimierung, S – Simulation, TB – Tabellenblatt, LOPF – Linear Optimal Power Flow, OPF – Optimal Power Flow, IP – Investitionsplanung, KM – Kostenminimierung, EZ – Emissionszieleinhaltung, WS – Wärmesektor, ELS – Elektrizitätssektor, MS – Mobilitätssektor, FOSS – Free Open Source Software

Klassifizierungsparameter	PyPSA [2]	Oemof [7]
Basisinformationen		
Entwickler	KIT, FIAS	RLI, HSF, DLR, EUF, OvGUM
Veröffentlichungsdatum	2015	2015
Modellzweck		
Genereller Zweck	V, E, R	V, E, R
Spezifischer Zweck	EV, EF, B, MA	EV, EF, B, MA
Geographische Abdeckung	BD	BD
Sektorenabdeckung	BD	BD
Zeithorizont	KZ	KZ, LZ
Entwicklungsstand (Version)	0.17.1	0.4.2
Letzter Commit (Aktualität)	12.08.2021 (Stand: 28.09.2021)	03.06.2021 (Stand: 28.09.2021)
Anzahl der veröffentlichten Versionen	29 (Stand:28.09.2021)	30 (Stand: 28.09.2021)
Dokumentation vorhanden?	Ja	Ja
Mathematische Beschreibung		
Programmierungsumgebung	Python, Pyomo	Python, Pyomo
Frei verfügbar?	Ja	Ja
Unterstützte Betriebssysteme	Windows, Mac, Linux	Windows, Mac, Linux
Objekt-orientiert?	Ja	Ja
Unterstützte externe Solver	GLPK, CLP/CBC, Gurobi	CPLEX, Gurobi, GLPK, CLP/CBC, Pyomo supported solvers
Daten		
Eingabedatenformate	CSV, Python dicts, pandas Dataframes, netCDF, HDF5, pypower, pandapower	Python dicts, XLSX, CSV, db, OEP
Eingabedatenart	QL, QN, (MON), A	QL, QN, (MON), A
Ergebnisdatenformate	CSV, pandas Dataframes, Python dicts, netCDF, HDF5	Python dicts, CSV, txt, db, collecting, grouping
Analytischer Ansatz	BU	BU, TD, HYB
Mathematischer Ansatz	LP, GGP, NLP	LP, GGP
Methodik	Ö, ÖGG, O, S	ÖGG, O, S, TB
Optimierungsbeschränkung/ Netzmodelle	OPF, LOPF	LOPF
Dynamikentscheidung bei der Optimierung	IP	IP
Zielsetzung (-Funktion)	KM, EZ	KM, EZ

Modellbildung		
<i>Repräsentierte Energiesektoren</i>	WS, ELS	WS, ELS, MS
<i>Zeitliche Auflösung</i>	Flexibel	Flexibel
<i>Räumliche Auflösung</i>	Flexibel	Flexibel
<i>Zeitschritte</i>	BD	BD
<i>Modularität</i>	Ja	Ja
<i>(Lastprofile erstellen)</i>	Nein	Ja (demandlib)
<i>Modellstruktur</i>		
<i>Grad der Endogenisierung</i>	Niedrig	Niedrig
<i>Beschreibung von Nicht-Energiesektoren</i>	Niedrig	Niedrig
<i>Beschreibung der Endverbraucher</i>	Niedrig	Mittel
<i>Beschreibung der Transporttechnologien</i>	Hoch	Mittel
<i>Technologischer Detailgrad</i>		
<i>Standardmäßige Darstellbare technische Komponenten</i>	Line, Transformer, Link, Load, Generator, StorageUnit, Store, ShuntImpadance	Sink, Source, Transformer, ExtractionTurbineCHP, GenericStorage, OffsetTransformer, ElectricalLine, GenericCAES, Link, SinkDSM
<i>Detailgrad der techn. Komponenten</i>	Hoch	Hoch
<i>Anzahl exogener Parameter</i>	Hoch	Hoch
<i>Erweiterbar?</i>	Ja	Ja
<i>Elektrische Netzwerkabdeckung</i>	Ja	Ja
Durchdringung erneuerbarer Energien		
<i>Erneuerbare Energien-Formate</i>	Ja	Ja
<i>Erneuerbare Energieeinspeisungszeitreihen</i>	Ja (<i>atlite</i>)	Ja (<i>feedinlib</i>)
<i>Darstellbare Anlagen zur erneuerbaren Energieerzeugung</i>	Wind (on-/ offshore), PV, (Biomasse), Wasserkraftwerk	Wind (on-/ offshore), PV, (Biomasse), Wasserkraftwerk
<i>Emissionen</i>	CO ₂	BD
Kollaborative Eigenschaften		
<i>Verfügbarkeit</i>	FOSS	FOSS
<i>Offenheit der Daten</i>	Ja	Ja
<i>Lizenz</i>	GPLv3	MIT License
<i>Entwicklerperspektivenspektrum</i>	Hoch	Mittel
<i>Einheitlichkeit der Terminologie</i>	Mittel	Hoch
<i>Wissenschaftliche Standards</i>	Ja	Ja
<i>Anzahl Beispiel Scripts in GitHub-Repository</i>	47	48 (oemof-solph, neueste Version)
<i>Zitationsanalyse</i>		
<i>Zitierhäufigkeit</i>	229 (Quelle: Google Scholar, Stand: 28.09.2021)	132 (Quelle: Google Scholar, Stand: 28.09.2021)
<i>Zitierende Arbeiten</i>	[13]	[12]
<i>Entwicklersupport</i>	Mittel	Hoch

Allgemein lässt sich auf Grundlage von Tabelle 3 schließen, dass *oemof* und *PyPSA* zwei ähnliche ESMF sind, welche jedoch aufgrund ihres Programmierzweckes seitens der Entwickler für unterschiedliche Modellierungszwecke besser geeignet sind als das jeweils andere Framework. So ist *oemof* geeigneter für sektorübergreifende Studien wohingegen *PyPSA* geeigneter für die Modellierung von elektrischen Übertragungsnetzen ist. Der Fokus beider Frameworks liegt auf unterschiedlichen Aspekten der Energiesystemmodellierung. Dies wird u.a. aus den Forschungsprojekten deutlich, welche die beiden Frameworks als Modellgeneratoren und Analysewerkzeuge nutzen. *Oemof* weist eine höhere Bandbreite an Forschungsfeldern innerhalb der ESM auf als *PyPSA*. So wird *oemof* häufig in Verbindung mit sektorübergreifenden Studien in Verbindung mit erneuerbaren Energien genutzt, was u.a. aus den Veröffentlichungen von Maruf (2021) [72, 73] oder Hilpert et al. (2020) [74] hervorgeht. Des Weiteren ist *oemof* ein beliebtes Analysewerkzeug, um das Potential von Energiespeichern jeglicher Art zu bestimmen (siehe dazu u.a. Hilpert (2020) [75], Kaldemeyer et al. (2016) [76] und Boysen et al. (2017) [77]). *PyPSA* hingegen weist hingegen ein höheres Potential zur Analyse des Stromnetzes auf, was u.a. auf die ausführliche mathematische Beschreibung von Übertragungsnetzen hervorgeht. Dies spiegelt sich auch in den Studien wider, welche *PyPSA* als Modellwerkzeug nutzen, um diese simulieren und optimieren zu können. Beispiele hierfür sind etwa die Studien von Gorenstein Dedecca et al. (2018) [78], Hörsch et al. (2018) [79], Schlachtberger et al. (2018) [80] oder Schlott et al. (2018) [81].

Welches der beiden Frameworks für die Modellierung von Energiesystemen mit einem hohen Anteil an erneuerbaren Energien geeigneter ist, lässt sich aus der theoretischen Analyse nicht erschließen. Beide Frameworks weisen ähnliche Eigenschaften in Bezug auf die Integration dieser auf und sowohl mit *oemof* als auch *PyPSA* wurden Studien durchgeführt, in denen im Zusammenhang mit erneuerbaren Energien geforscht wurde (siehe bspw. [72–74, 80–84]). Aus diesem Grund wird im Folgenden eine Fallstudie auf Grundlage des Energiesystemmodells von Hamburg mit Fokus auf die Integration erneuerbarer Energien durchgeführt.

5.2 Datenquellen

Die Eingabedaten sind ein wesentlicher Bestandteil, um ein ES modellieren zu können. Um die Modellierung transparent und reproduzierbar zu gestalten, müssen, wie in Kap. 2.2 Open Source bereits erwähnt, alle Datenquellen offengelegt werden, was im folgenden Unterkapitel gemacht wird.

5.2.1 Lastgangzeitreihen für den Energiebedarf

Das Stromnetz Hamburg ist der städtische Verteilungsnetzbetreiber von elektrischer Energie in Hamburg. Es ist für Netzbetrieb, Erhalt und Ausbau des Netzes zuständig und bietet auf der Seite Energieportal Hamburg [85] Informationen zu dem Stromnetz in Hamburg an. Außerdem ist dort der stündlich aufgelöste Strombedarf, importierte Strom und erzeugte Strom als Zeitreihe in Form einer CSV (engl.: *comma seperated values*)-Datei herunterzuladen. Der Lastgang des Fernwärmebedarfes in Hamburg wird ebenfalls in Form einer CSV-Datei in das Modell implementiert. Die Daten stammen aus dem Jahr 2017, was der geringen Verfügbarkeit dieser Daten geschuldet ist.

5.2.2 Energieerzeugungszeitreihen von erneuerbaren Energiequellen

In diesem Falle wurde zur Erstellung der Zeitreihen einerseits die Vestas V112 3MW Turbine als Standardturbine gewählt und das Solarpanel mit Monokristallinen Silicium Zellen als Standardsolarpanel für die Simulation der Zeitreihe in *atlite* gewählt (siehe Code 2 in Kapitel 4.4).

Atlite besitzt keine graphische Benutzeroberfläche, kann keine Vorhersagen für die zukünftige Energieerzeugung treffen und durch die ausschließliche Nutzung von Wetterdaten, die zur Berechnung der erzeugten Energie durch die Anlagen hinzugezogen werden, keine Übereinstimmungen zur tatsächlich erzeugten Energie garantieren. Ein Vergleich von Daten der OPSD Plattform [86] zeigt jedoch, dass die durch *atlite* prognostizierte Energieerzeugung sich nah an den simulierten, der in OPSD hinterlegten, Daten befindet. Diese simulierten Zeitreihendaten der erneuerbaren Energieerzeugung in der OPSD Datenbank werden durch *Renewables.ninja* [87] zur Verfügung gestellt. Die Modelle zur Simulation der Daten, die in *Renewables.ninja* hinterlegt werden, sind hierbei der *Global Solar Energy Estimator* [88] und die *Virtual Wind Farm* [89] entnommen, welche die Wetterdatenbanken *MERRA-2* [90] und *SARAH* [69, 70] nutzen.

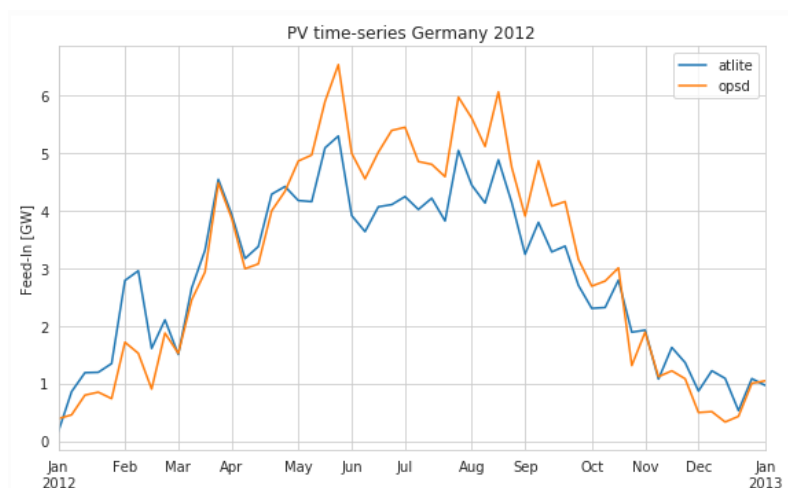


Abbildung 6: Vergleich von *atlite* und *opsd* Datensätzen zur Erzeugung elektrischer Energie aus PV-Anlagen in Deutschland 2012 [64]

Auf der Seite des Stromnetz Hamburg [91] kann zum zusätzlichen Vergleich die prognostizierte und tatsächliche Energieerzeugung bzw. -einspeisung von erneuerbaren Energien, ebenfalls nur auf Wetterdaten basierend, dargelegt werden, siehe dazu Kapitel 5.2.1.

5.2.3 Kraftwerksdaten

Zu den Konventionellen Kraftwerken wurden Daten aus Datenbanken, welche von Daten aus der Webrecherche eingespeist wurden, verwendet. So werden auf den Seiten der Energienetz- bzw. Kraftwerksbetreiber grundlegende Daten zu den Kraftwerken veröffentlicht (s. dazu [92–95]). Zusätzlich bieten die OPSD eine Liste [96] und das Umweltbundesamt die Datenbank „Kraftwerke in Deutschland“ [97] zu allen registrierten konventionellen Kraftwerken in Deutschland an.

5.2.4 Emissionen

Die ausgestoßenen Emissionen der einzelnen Energieträger sind für die zu beantwortende Forschungsfrage der Fallstudie unabdingbar. Die Emissionen der Primärenergieträger wurde Quaschnig 2019 [98] entnommen, wobei die die spezifischen Emissionen, die bei der Müllverbrennung entstehen, mit Hilfe von Schwarzböck 2015 [99] berechnet wurden (siehe dazu Anlage 1). Die Emissionen des „Dummy“-Importes, die für Strom und Wärme festgelegt wurden, sind der Grafik des Umweltbundesamtes [100] zu entnehmen.

Als Emissionslimit-Grundlage der Stadt Hamburg wurde der Ausstoß an CO₂ im Jahr 1990 in Hamburg dem Statistikportal Statistische Ämter des Bundes und der Länder entnommen. Dieser betrug im Jahr 1990 207 Mt_{CO2}/Jahr [101].

Der Anteil des Energiesektors am gesamten Treibhausgas-Ausstoß in Deutschland liegt gerundet bei 84% und wurde der Tabelle in Anlage 2 entnommen. Da der Anteil der anderen beiden Treibhausgase in dieser Tabelle sehr klein verglichen zum Gesamtausstoß von CO₂ ist, wird dieser Anteil von 84% als Anteil des CO₂-Ausstoßes des Energiesektors verwendet.

5.2.5 Kosten

Die Betriebs- und Investitionskosten der erneuerbaren Energieerzeugungsanlagen wurden auf Grundlage der Studie „Stromgestehungskosten erneuerbarer Energien“ des Fraunhofer-Instituts für Solare Energiesysteme in dem Modell bestimmt. Die Betriebskosten der konventionellen Kraftwerke wurden ebenfalls auf Grundlagen der o.g. Studie geschätzt [102].

5.3 Modellkomponenten

Die Komponenten eines realen Energiesystems sind komplex abgestimmte technische Anlagen, die in einem ganzheitlichen Energiesystemmodell nicht sonderlich technisch detailliert dargestellt werden können. Daher ist zunächst herauszufinden, wie man die technischen Anlagen auf ein Minimum an Komplexität herunterbrechen kann, ohne ihre Wirkung auf das Energiesystem zu verfälschen.

Bei der Erstellung der Komponenten weisen *oemof* und *PyPSA* grundsätzliche Unterschiede auf.

So lassen sich die Parameter der Energieströme bei *oemof* mit Hilfe einer Dictionary-Funktion dem jeweiligen Bus durch das *solph*-Modul und die Funktion *solph.Flow* zuweisen, wohingegen *PyPSA* mit reinen Schlüsselworten in der jeweiligen Funktion für die Komponente auskommt.

Die Indizierung jeder Komponente wurde mit Hilfe des *tessif*-tools *Namedtuples* vorgenommen, hierbei werden die Eckdaten zur eindeutigen Identifizierung einer Komponente durch Parameter wie Bezeichnung, Standort, Koordinaten, Energieträger etc. in einen zusammenhängenden String verwandelt, um diesen bei der Namensgebung in *oemof* und *PyPSA* verwenden zu können, da diese ausschließlich den Datentyp *Strings* zulässt.

Alle Komponenten des ES von Hamburg mit detaillierten Parameterwerten sind in Anlage 3 zu finden und in Abbildung 7 graphisch anhand der Knotenpunkte dargestellt, welche mit Hilfe des Visualisierungstools „*nxgraph*“ der „*visualize*“-Bibliothek von *tessif* erstellt wurde.

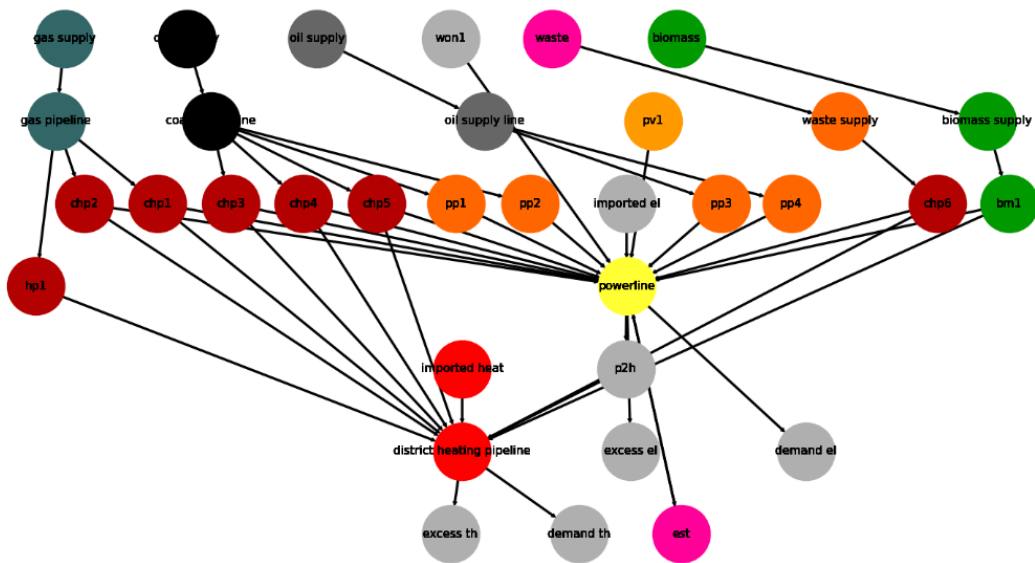


Abbildung 7: Energiesystem Hamburgs mit visualisierten Knotenpunkten mit Hilfe der Funktion "tessif.visualize.nxgrph"

5.3.1 Busse

Die Busse sowohl in *oemof* als auch in *PyPSA* erfüllen den Zweck eines Knotenpunktes, welcher dem Gesetz der Energieerhaltung unterliegt. Der *Bus* hat hierbei keine limitierende Wirkung auf das Energiesystem, da diese von den Quellen, Senken und Transformatoren ausgeht. Sie sind lediglich eine Verbindung sowie Schauglas zwischen den einzelnen Komponenten, die man sich vorstellen kann wie ein Rohr mit gegebenenfalls mehreren Abzweigungen.

D.h., dass in dem Fall des Energiesystems von Hamburg jeweils ein Bus für gesamte Stromnetz (*powerline*) und einer für das Fernwärmenetz (*district heating pipeline*) sowie für die Primärenergieträger wie Kohle, Heizöl, Gas und Biomasse implementiert wurde. Es wird vereinfacht angenommen, dass jeder Verbraucher im betrachteten Gebiet an demselben Verteilungssystem des jeweiligen Energieträgers angeschlossen ist.

Die *Busse* nehmen eine zentrale Rolle in der Analyse des optimierten Systems ein, da jede zeitlich abhängige Energieübertragung in Form von Rohstoffen und Energieströmen jeweils durch diese erfolgt.

5.3.2 Quellen

Als Quelle wird eine Komponente benannt, die die Primärenergie für das System bereitstellt oder im Falle der erstellten Energiesystemmodelle, welches die Energie aus erneuerbaren, also anthropogener Sicht unendlich reproduzierbaren, Energiequellen in nutzbare elektrische oder thermische Energie umwandelt.

Durch die Quellen werden die spezifischen CO₂-Emissionen der Energieträger in das Modell implementiert. In den spezifischen Emissionen der erneuerbaren Energien werden die durchschnittlichen emittierten CO₂-Mengen, die bei der Herstellung der Energieerzeuger pro installierter Leistungseinheit anfallen, dargestellt, da bei dem Stromgewinnungsprozess keine Treibhausgase emittiert werden.

Die spezifischen CO₂-Emissionen werden in *oemof* und *PyPSA* jedoch unterschiedlich implementiert. In *oemof* kann dies über die *solph*-Bibliothek direkt in der Dictionary-Funktion für die angeschlossenen Verbraucher für die jeweilige Energiequelle mit Hilfe des *emission_factor* deklariert werden (s. Code 3).

```
gas_supply = solph.Source(
    label=gas_supply_uid,
    outputs={es.groups[str(gas_pipeline_uid)]:
             solph.Flow(emission_factor=0.2)}
)
```

Code 3: Deklaration der spezifischen CO₂-Emissionen in *oemof* anhand des Beispiels "gas_supply"

Durch das *groups*-Modul in *oemof* lassen sich, wie z.B. in dem Fall von Code 3 die deklarierten Verbindungen mit der *Bus*-Komponente *gas_pipeline* zusammenfassen, da *oemof* sonst bei der Modellierung Probleme hat, die mit der Quelle verbundenen Komponenten erfassen zu können.

In dem *PyPSA* ESM wird die Komponente „Generator“ als korrespondierende Komponente zur *oemof*-Quelle verwendet werden. Üblicherweise verwendet *PyPSA* die Klasse namens *Carrier*, um die spezifischen CO₂-Emissionen eines Energieträgers zu definieren. So muss z.B. bei der Festlegung der spez. Emission des Energieträgers Erdgas zunächst die *Carrier*-Komponente angelegt und anschließend bei der Erstellung der Quellenkomponente (*Generator*) der deklarierte Name als String in das Schlüsselargument *carrier* hinzugefügt werden (siehe dazu Code 4).

```
network.add(class_name="Carrier",
            name="gas",
            co2_emissions=0.2)

network.add(class_name="Generator",
            name=gas_supply_uid,
            bus=str(gas_pipeline_uid),
            p_nom_extendable=True,
            )
```

Code 4: Deklaration der spezifischen CO₂-Emissionen in *PyPSA* anhand des Beispiels "gas_supply"

5.3.3 Senken

Die Implementierung der Senken unterscheidet sich von *oemof* zu *PyPSA* kaum. So wird in *oemof* die Komponente *Sink* (Code 5) und in *PyPSA* (Code 6) die Komponente *Load*, welche ausschließlich einen Eingang besitzen, verwendet. Durch die Attribute *fix* in *oemof.solph.Flow* und *p_set* in der *PyPSA network.add-Funktion* kann die vorhandene Zeitreihe der Last eingelesen werden und definiert so den Energiebedarf des Modells über den simulierten Zeitraum.

```
demand_el = solph.Sink(label=demand_el_uid,
                      inputs={powerline: solph.Flow(fix=de_HH_norm,
                                                    nominal_value=max_de,
                                                    ),
                              },
                      )
```

Code 5: Deklaration der Senken in oemof anhand des Beispiels der elektrischen Last

```
network.add(class_name="Load",
            name=demand_el_uid,
            bus=str(powerline_uid),
            p_set=de_HH)
```

Code 6: Deklaration der Senken in PyPSA anhand des Beispiels der elektrischen Last

5.3.4 Transformer

Die Energietransformer in einem Energiesystem lassen sich in Kraftwerke, die die Primärenergie in nur eine Form von Sekundärenergie umwandeln und gekoppelte Kraftwerke, die sog. Kraft-Wärme-Kopplungsanlagen wie z.B. BHKW, GuD etc., einteilen.

Oemof verfügt hierbei in seiner *solph*-Bibliothek über eine Komponente namens *Transformer*, welchen auszeichnet, dass dieser sowohl einen Input, welcher an den Bus des Primärenergieträgers gebunden wird, als auch mehrere Outputs besitzt, welche mit der *powerline* oder *district heating pipeline* und, im Falle der gekoppelten Kraftwerke, mit beiden *Busses* verbunden ist. Zwar gibt es darüber hinaus noch die Komponente *ExtractionTurbineCHP* [103] diese lässt sich jedoch aufgrund der unzureichenden Datenlage aus Anlage 3 nicht realisieren.

Mit Hilfe der *dictionary*-Schreibweise in den geschweiften Klammern können ohne großen Aufwand auch mehrere Ausgangs-Busse definiert werden (s. Code 7).

```
### HKW ADM ###

chp1_nc = solph.NonConvex(initial_status=in_stat,
                          minimum_uptime=1,
                          minimum_downtime=1,
                          startup_costs=24,
                          shutdown_costs=0,
                          activity_costs=cfba)

chp1 = solph.Transformer(
    label=chp1_uid,

    inputs={gas_pipeline: solph.Flow()},

    outputs={powerline: solph.Flow(nominal_value=22.5,
                                    variable_costs=90,
                                    nonconvex=chp1_nc),
            fernwaerme: solph.Flow(nominal_value=35,
                                    variable_costs=21.6,
                                    nonconvex=chp1_nc)},

    conversion_factors={gas_pipeline: 1, powerline: 0.3773,
                       fernwaerme: 0.3},
)
```

Code 7: Erstellung eines KWK-Kraftwerkes in oemof anhand des Beispiels HKW ADM

Die *NonConvex*-Funktion in *oemof* bietet zusätzlich weitere Modellierungsmöglichkeiten, indem in dieser etwa der anfängliche Kraftwerksstatus (*initial_status*), die Anfahrkosten des Kraftwerks (*startup_costs*) usw. definiert werden können.

Die Modellierung von Kraftwerken in *PyPSA* gestaltet sich gegenüber *oemof* etwas schwieriger, da die Komponente *Link* zur Modellierung von Kraftwerksblöcken verwendet wird. Dadurch, dass sich die *Link*-Komponente von *PyPSA* mit weniger Aufwand im Vergleich zur *Generator*-Komponente modifizieren lässt, wird der *Link* in dieser Arbeit so umgestaltet, dass nicht nur konventionelle Kraftwerke mit ausschließlicher Stromerzeugung oder Wärmeerzeugung, sondern auch KWKK modelliert werden können. Hierzu gibt es ein Beispieldokument zur Deklaration eines KWKK [104], vergleichbar zu *oemof*-Komponente *ExtractionTurbineCHP*, welches aus denselben Gründen, der unzureichenden Datenlage, nicht realisiert werden kann.

```
override_component_attrs = pypsa.descriptors.Dict(
    {k: v.copy() for k, v in pypsa.components.component_attrs.items()})

override_component_attrs["Link"].loc["bus2"] = [
    "string", np.nan, np.nan, "2nd bus", "Input (optional)"]
```

Code 8: Implementierung der dictionary-Erweiterungen in *PyPSA* anhand des Beispiels "bus2"

Der *Link* verfügt standardmäßig über einen Input-Bus (*bus0*) und einen Output-Bus (*bus1*), durch die Erweiterung des *dictionary*s (s. Code 8) dessen können weitere Busse hinzugefügt werden, in diesem Fall ein weiterer Output-Bus, um die gekoppelte Strom- und Wärmeerzeugung zu modellieren (*bus1* = *powerline*, *bus2* = *district heating pipeline*). Des Weiteren wird auch der Wirkungsgrad (*efficiency* = el. Wirkungsgrad; *efficiency2* = th. Wirkungsgrad) sowie die zugehörige nominelle Leistung (*p_nom* = nominelle el. Leistung, *p_nom2* = nominelle th. Leistung) für Strom- und Wärmeauskopplung als Erweiterung hinzugefügt. Die Optimierungsergebnisse der Leistungsflüsse in den einzelnen Zeitschritten der Komponenten werden in der Leistung *p0* (Quelle) und *p1* (elektrische Leistung) sowie der *dictionary*-Erweiterung *p2* (thermische Leistung) gespeichert.

Das Problem der Verwendung von *Link*-Komponenten als Kraftwerksmodell liegt in der begrenzten Möglichkeit von Attributverwendungen. So besitzt die *Generator*-Komponente von *PyPSA* MILP Attribute wie z.B. Anlaufkosten oder minimale Stillstandszeit, wohingegen die *Link*-Komponente diese Attribute standardmäßig nicht aufweist. Die *Generator*-Komponente kann jedoch nicht ohne größeren Aufwand, durch die Erweiterungsfunktion, in *PyPSA* erweitert werden, um beispielsweise einen weiteren Bus zur gekoppelten Energieerzeugung von elektrischer und thermischer Energie, wie sie in KWK-Anlagen stattfindet, zu modellieren. Dies liegt daran, dass die Hintergrundfunktion zur Optimierung der Komponente ebenfalls neu definiert werden müsste und die Attribute für den zweiten *Bus*-Output zu dieser hinzugefügt werden müssten, was ohne umfassende Kenntnisse von *PyPSA* nicht möglich ist.

```

### HKW ADM ###

network.add(class_name="Link",
            name=chp1_uid,
            multiple_inputs=False,
            multiple_outputs=True,
            bus0=str(gas_pipeline_uid),
            bus1=str(powerline_uid),
            bus2=str(fernwaerme_uid),
            p_nom=22.5,
            marginal_cost=90,
            efficiency=0.3773,
            efficiency2=0.3)

```

Code 9: Erstellung eines KWK-Kraftwerkes in PyPSA anhand des Beispiels HKW ADM

Die *Power-to-X*-Technologien lassen sich in *PyPSA* und *oemof* ebenfalls mit Hilfe der Link- bzw. Transformer-Komponente modellieren. So wird in dem Energiesystem von Hamburg der PTH-Heizkessel Karoline in dieser Weise dargestellt.

5.3.5 Energiespeicher

Energiespeicher spielen eine entscheidende Rolle in der Energiewende. Durch den fluktuierenden Leistungsfluss erneuerbarer Energien wie Wind- und Solarkraft wird eine kontinuierliche Lastabdeckung benötigt. Diese kontinuierliche Lastabdeckung kann auch mit Hilfe von Energiespeichern bewerkstelligt werden, weshalb diese zur CO₂-Emissionsreduzierung eine große Rolle spielen, um auch während geringem Ertrag aus erneuerbaren Energien die Netzlast abdecken zu können. Aufgrund dessen wurden die Energiespeicher von elektrischer Energie sowohl in *oemof* als auch *PyPSA* erweiterbar modelliert. D.h., dass bei der Optimierung die Kapazität und Leistung vom Solver bei entsprechend angegeben Investitionskosten erweitert werden können.

```

### Electrical Storage Unit ###

inv_est = solph.Investment(ep_costs=economics.annuity(capex=1000000,
                                                    n=10,
                                                    wacc=0.05)
                        ) # capex [€/MW], n [y], wacc [%]

est = solph.GenericStorage(label=energy_storage_uid,
                          inputs={powerline: solph.Flow(emission_factor=0,
                                                         variable_costs=20,
                                                         )},
                          outputs={powerline:solph.Flow(emission_factor=0,
                                                         variable_costs=20,
                                                         )},
                          initial_storage_level=None,
                          balanced=True,
                          investment=inv_est,
                          )

```

Code 10: Deklaration von Energiespeichern in *oemof* anhand des Beispiels "Electrical Storage Unit"

In *oemof* wird der Energiespeicher mit Hilfe der Komponente *GenericStorage* modelliert, wobei sowohl ein In- und ein Output-Bus deklariert werden muss, welcher im Falle des Energiespeichers jeweils der *powerline*-Bus ist (s. Code 10). Die Komponente, die in *PyPSA* in dieser Arbeit zur Modellierung eines Energiespeichers verwendet wird, ist die *StorageUnit*. Hierbei muss nur der Bus deklariert werden, an welchen der Speicher angeschlossen ist, um In- und Output zu definieren (s. Code 11).

```
### Electrical Storage Unit ###  
  
network.add(class_name="StorageUnit",  
            name=energy_storage_uid,  
            p_nom=0,  
            p_initial=0,  
            p_nom_extendable=True,  
            capital_cost=10,  
            marginal_cost=20,  
            bus=str(powerline_uid))
```

Code 11: Deklaration von Energiespeichern in *PyPSA* anhand des Beispiels "Electrical Storage Unit"

Des Weiteren wird bei beiden eine Erweiterung, wie o.g. erlaubt, sowie der Wirkungsgrad bei Eintritt sowie Austritt festgelegt.

5.3.6 Emissionsziel

Die Emissionsziele können durch die *Global Constraints* Komponente in *PyPSA* und in *oemof* in der Funktion *emission_limit* festgelegt werden.

Der Solver verfolgt anschließend bei der Optimierung das Emissionsziel als globale Systembeschränkung, welches dem primären Ziel, der Kostenreduktion, unterworfen ist. Hierbei kann es vorkommen, dass durch die vorrangige Kostenoptimierung die Emissionsziele nicht eingehalten werden können und der Solver deswegen die Gleichung nicht lösen kann.

5.3.7 Zusammenfassung der Parameter

In Tabelle 4 wird ein Überblick über die verwendeten übereinstimmenden Parameter der Komponenten beider Frameworks gegeben, welche in der Fallstudie mit denselben Werten aus Anlage 3 belegt wurden.

Tabelle 4: Zusammenfassung der Komponentenparameter von oemof und PyPSA

Parameter	oemof Parameter	Datenformat oemof	PyPSA Parameter	Datenformat PyPSA	Einheit	Beschreibung
Komponentenname	label	string	name	string	-	Name der Komponente
Bus-Verbindung	input / output	dict	bus (bus1, bus2, ...)	string	-	Verbindung zu Bus
Nominelle Leistung	nominal_value	float	p_nom	float	MW	Nominelle Leistung der Komponente
Normiertes Minimum/Maximum	min / max	float / series	p_min_pu / p_max_pu	float / series	-	Auf die nominelle Leistung der Komponente normiert minimale/maximale Leistung
Fixierter Energiefluss	fix	float / series	p_set	float / series	MW	Festgesetzter Wert oder Zeitreihe für Energiefluss
Energieerzeugungskosten	variable_costs	float / series	marginal_cost	float / series	€/MWh	Energieerzeugungskosten pro Einheit und Zeitschritt
Erweiterbarkeit	investment	solph. Investment	p_nom_extendable	boolean	-	Legt fest, ob die Leistung der Komponente erweitert werden kann
Erweiterungsgrenzen	solph.Investment(minimum, maximum)	float	p_nom_min / p_nom_max	float	MW	Legt die Grenzen der Leistungserweiterung im Zuge der ES-Optimierung fest
Erweiterungskosten	solph.Investment(ep_costs)	float	capital_cost	float	€/MW	Kapitalkosten pro zugebauter Leistungseinheit
Wirkungsgrad	conversion_factors	float	efficiency	float	%	Wirkungsgrad der technischen Komponente

Nichtkonvexe Optimierung	nonconvex	solph.NonConvex	-	-	-	Das mathematische Modell ändert sich zu einer MILP
Kosten für Hoch-/Herunterfahren der technischen Komponente	NonConvex(startup_costs , shutdown_costs)	float / series	start_up_cost / shut_down_cost	float	€	Hochfahr- / Stillstandskosten der technischen Komponente
Minimale Stillstands-/Erzeugungszeit	NonConvex(minimum_downtime, minimum_uptime)	integer	min_down_time / min_up_time	integer	h	minimale Stillstands-/ Erzeugungszeit
Energieflussgradienten	NonConvex(positive_gradient ,negative_gradient)	dict	ramp_limit_up / ramp_limit_down	float	MW	Minimale/maximale Steigerung der Leistung über einen Zeitschritt
	"ub"	float / series	-	-	-	-
	"costs"	float	-	-	-	-
(Initialer) Status	NonConvex(initial_status)	boolean (0/1)	status	boolean / boolean series (0/1)	-	(Anfangs-)Status der Komponente (0=Aus, 1=An)
Akkumulierte Energieflüsse	summed_min / summed_max	float	-	-	MWh	Spezifisch minimaler/maximaler Wert summiert über den gesamten Simulationszeitraum
Spezifische CO₂-Emissionen	emission_factor	float	Carrier	Klasse	t _{CO2} /MWh	spezifische CO ₂ -Emissionen pro Energieeinheit
	-	-	co2_emissions	float	-	-

5.4 Szenarien

Um einen Vergleichswert zu haben, wird zunächst das Referenzszenario erstellt. Hierbei werden alle Werte aus den untersuchten Szenarien übernommen mit dem Unterschied, dass das Emissionslimit nicht in die Betrachtung aufgenommen wird.

Die Szenarien „CO₂ 2030“ und „CO₂ 2050“ portraituren die nötige strukturelle Veränderung der technischen Komponenten des Energiesystems Hamburgs. Die festgelegten CO₂-Grenzwerte werden für die Jahre 2030 (55 % Reduktion gegenüber 1990) und 2050 (95% Reduktion gegenüber 1990) in die Optimierung mit einbezogen, ebenso wie die Ausbaufähigkeit für erneuerbare Energien, Energiespeicher und der Energieumwandlungstechnologie PTH.

Die Szenarien werden mit gleichbleibendem Strom- und Wärmebedarf berechnet und dienen ausschließlich zum Vergleich der beiden Frameworks, jedoch nicht zur wissenschaftlichen Analyse des Hamburger Energiesystems an sich. Diese Analyse würde simulierte Strom- und Wärmebedarf, sowie Klimadaten voraussetzen und die technische Weiterentwicklung der Komponenten ebenso wie veränderte ökonomische Parameter nach sich ziehen und den Umfang dieser Ausarbeitung übersteigen.

5.5 Ergebnisse der Modellszenarien

Als Output, der direkt verglichen werden kann erhält man, neben den globalen Parametern, wie den Gesamtsystemkosten, Zeitserien der Energieflüsse über die simulierte Dauer. Die einheitliche Ergebnisdatenverarbeitung erfolgte mit Hilfe der *tessif*-Funktion *es2mapping*. Diese kann sämtliche Daten der simulierten ES erfassen und in Variablen zur Analyse ablegen. Der *tessif LoadResultier* sammelt die Ergebnisdaten der *oemof*- und *PyPSA*-Optimierungen und legt die Energieflüsse in dem Tabellentool *pandas DataFrames* in dem jeweiligen *dict* für jeden Knotenpunkt als *key* ab. Dabei wird zwischen Knotenflüssen, die hinaus- (*dict: node_outflows*) oder hereinführen (*dict: node_inflows*) sowie kumulierte Flüsse (*dict: node_load*) in den jeweiligen *Bus*-Komponenten unterschieden. Zusätzlich summiert die Funktion alle Energieflüsse über dem betrachteten Zeitraum zwischen jedem Knotenpunkt auf (*dict: node_summed_loads*).

Der PC, der die Berechnungen aller Optimierungen durchführt, ist mit einem AMD Ryzen 5 4600H @ 3.00 GHz und 16 GB RAM sowie dem Betriebssystem Windows 10 ausgestattet.

5.5.1 Referenzszenario

Das Szenario operiert wie die anderen beiden Szenarios mit 8760 Zeitschritten und einem Investmentmodus für erneuerbare Energien, jedoch im Gegensatz zu den anderen beiden Szenarien ohne Emissionsgrenzwert. Die abweichenden globalen Ergebnisdaten wie Kosten, Simulationsdauer, Anzahl von Beschränkungen innerhalb der Simulation etc., wie in Tabelle 5 zu sehen, legen die Unterschiede zwischen den ESMF in den simulierten spezifischen Energieflussdaten dar.

Tabelle 5: Globale Ergebnisdaten Referenzszenario

	oemof	PyPSA	Verhältnis $\frac{oemof}{PyPSA}$
Anzahl Iterationen	41.081	183.904	0,22
Gesamtsystemkosten [€]	974.589.164,20	2.697.005.495,00	0,36
Anzahl Beschränkungen	306.603	473.041	0,65
Anzahl Nonzeros	96.365	106.919	0,90
Anzahl von Variablen	411.729	210.253	1,96
Wallclock time [s]	8,52	110,03	0,08

Das Ergebnis der Gesamtsystemkosten kommt u.a. dadurch zustande, dass *PyPSA* nur in geringerem Maße auf die bereits installierte Leistung zurückgreift und durch den Ausbau der erneuerbaren Energien, besonders im Bereich der Windenergie, die elektrischen Speicher ausbauen muss (s. Abbildung 8, Anlage 4 und Anlage 5).

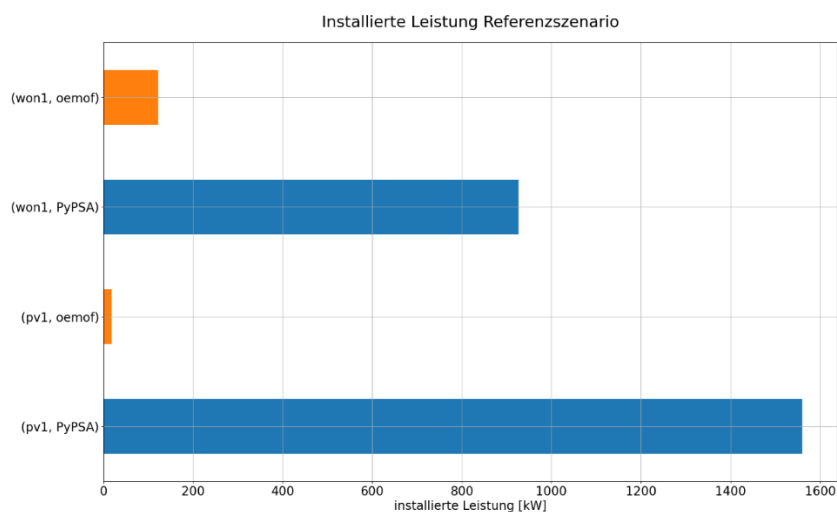


Abbildung 8: Vergleich oemof und PyPSA Installierte Leistung "pv1" und "won1" im Referenzszenario

Dabei ist die Leistung der Erzeuger im *PyPSA*-ES ausreichend, um die elektrische Last zu decken, jedoch wird ein Großteil der Energie in dem elektrischen Speicher zwischengespeichert, anstatt diese in das Netz einzuspeisen. Dies bedeutet, dass *oemof* im Elektrizitätssektor deutlich geringere Kosten als *PyPSA* wählt, indem es die bestehenden Kraftwerke nutzt und die Leistung der erneuerbaren Energien und Speicher nicht ausbaut. Außerdem fällt auf, dass *PyPSA* die Grundlast auf deutlich mehr Kraftwerke aufteilt. *Oemof* sichert die Grundlast hauptsächlich durch die Kohlekraftwerke *pp1* und

pp2, wohingegen *PyPSA* diese auf alle Kraftwerke, *pp1* bis *pp4*, sowie die KWKK *chp1* bis *chp5* aufgliedert (s. Abbildung 13, Abbildung 14). Ähnlich sieht es im Wärmesektor aus, wo die Optimierung seitens *PyPSA* einen Ausbau von Heizkesselkapazitäten (*hp1*) und der PTH-Technologie (*p2h*) wählt (s. Anlage 6 und Anlage 7). *Oemof* hingegen optimiert die Wärmeerzeugung indem die KWK-Kraftwerke *bm1*, *chp3*, *chp4* und *chp5* den Bedarf decken.

5.5.2 Szenario „CO2 2030“

In dieser Szenario-Simulation nähern sich *oemof* und *PyPSA* mit den globalen Ergebnissen (s. Tabelle 6) an. So ist die Anzahl der Iterationen, Variablen, Nicht-Null-Einträgen in den Zeitreihen beider Simulationen und die Simulationszeit deutlich ähnlicher. Die Gesamtsystemkosten divergieren jedoch weiter auseinander.

Tabelle 6: Globale Ergebnisdaten Szenario „CO2 2030“

	oemof	PyPSA	Verhältnis $\frac{oemof}{PyPSA}$
Anzahl Iterationen	64.821	89.868	0,72
Gesamtsystemkosten [€]	1.638.390.773,00	5.973.987.647,00	0,27
Anzahl Beschränkungen	306.604	473.042	0,65
Anzahl Nonzeros	105.126	106.945	0,98
Anzahl von Variablen	411.729	473.042	0,87
Wallclock time [s]	68,66	135,29	0,51

Veränderte Ergebnisse lassen sich auch bei der Analyse der Lastgänge beobachten: Das KWKK *bm1* wird von *oemof* zu keinem Zeitpunkt mehr aktiviert, wohingegen *chp1* und *chp2* in Teillast über nahezu das gesamte Jahr elektrische Energie in das Stromnetz einspeisen. *PyPSA* hingegen betreibt alle KWK-Kraftwerke genauso wie im Referenzszenario, in welchem keinerlei CO₂-Ausstoßbeschränkungen gelten (s. Anlage 8). Die Nutzung elektrischer Energie aus dem Strommix nimmt hingegen im *PyPSA*-ES um etwa das Doppelte zu, während die Energieerzeugung der Dampfkraftwerke *pp1* und *pp2* um mehr als die Hälfte abnimmt (s. Abbildung 9). Bei diesen verzeichnet das *oemof*-ES sogar eine Reduktion auf ein Drittel der ursprünglichen Energieumwandlung in das Übertragungsnetz eingespeiste elektrische Energie. Die Kraftwerke *pp3* und *pp4* werden in diesem Szenario ganz abgeschaltet (s. Anlage 8 und Anlage). In dem Ausbau der erneuerbaren Energien gibt es deutliche Unterschiede zu verzeichnen. Während *PyPSA* die installierte Leistung der Photovoltaikanlagen verdoppelt und etwa ein Drittel Mehrerträge gegenüber dem Referenzszenario dadurch verzeichnet, baut *oemof* die PV-Kapazitäten um das 125-fache von 18 MW auf 2230 MW aus. Die installierte Leistung der Windenergieanlagen wird, wie in Abbildung 10 zu sehen, von *oemof* ebenfalls deutlich von 123 MW auf 3135 MW und in der *PyPSA*-ES Optimierung von 930 MW im Referenzszenario um etwa 75% erhöht.

Vergleich von elektrischer Energieerzeugung von pp1 und pp2 im Szenario "CO2 2030" gegenüber dem Referenzszenario

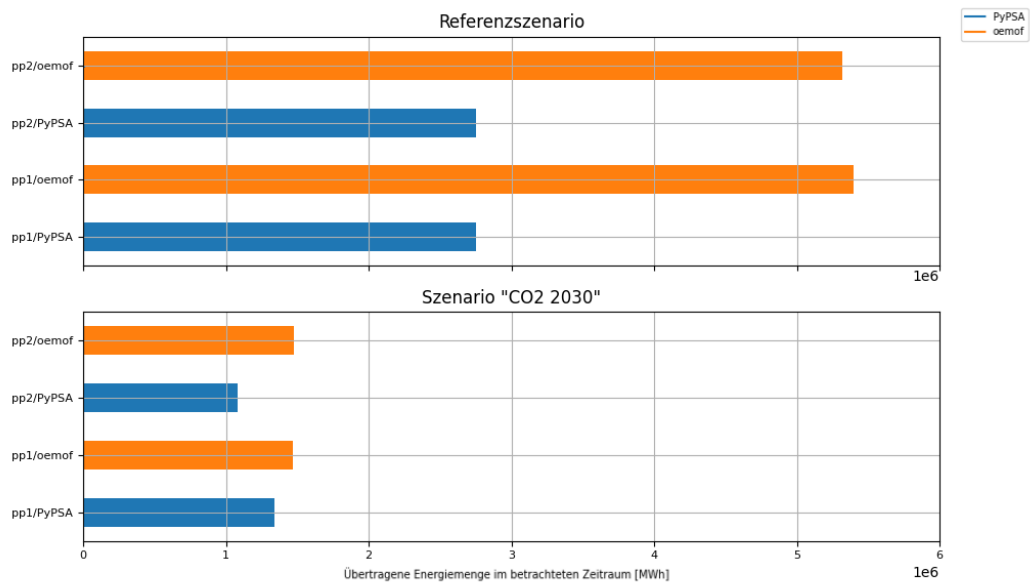


Abbildung 9: Vergleich von elektrischer Energieerzeugung von pp1 und pp2 im Szenario "CO2 2030" gegenüber dem Referenzszenario

Der Überschuss der elektrischen Energie wird von *oemof* überwiegend in die *excess el*-Senke übertragen und zu einem kleineren Teil in Nutzwärme mit Hilfe des Elektrodenheizkessels *p2h* in das Fernwärmenetzwerk eingespeist. *PyPSA* hingegen baut die Speicherkapazität der *est* aus und verdoppelt die gespeicherte Menge bzw. Kapazität der Speichervorrichtungen.

Durch die deutlich erhöhte Laufzeit und installierte Leistung der KWKK und *p2h* erzeugt das *oemof*-ES so viel Heizwärme, dass der Bedarf des Fernwärmenetzes von Hamburg gedeckt wird und somit 140 GWh thermische Energie in die *excess th*-Senke geleitet werden müssen (s. Anlage 11).

5.5.3 Szenario „CO2 2050“

Die globalen Ergebnisdaten „Anzahl Iterationen“ und „Anzahl Variablen“ sowie „Gesamtsystemkosten“ und „Berechnungsdauer“ (engl.: „Wallclock time“) der Optimierung aus den Simulationen von *oemof* und *PyPSA* in Tabelle 7 divergieren in diesem Szenario gegenüber „CO2 2030“ auseinander. So erhöht sich die Anzahl der Iterationen bei der *PyPSA*-Simulation um 15%, wohingegen diese sich bei der *oemof*-Simulation um ein Drittel verringern. Die Gesamtsystemkosten erhöhen sich in der *oemof*-Simulation um ca. das Vierfache bei *PyPSA* ist sogar ein Anstieg von knapp dem Fünffachen gegenüber dem Szenario „CO2 2030“ zu bemerken. Ebenfalls stark verändert hat sich die Anzahl der Variablen in der *PyPSA*-Simulation. Diese verringert sich im „CO2 2050“-Szenario um etwa die Hälfte. Die Simulationszeit von *oemof* verringert sich durch die geringere Anzahl an Iterationen um ein Drittel, die von *PyPSA* jedoch erhöht sich aufgrund dieser.

Tabelle 7: Globale Ergebnisdaten Szenario „CO2 2050“

	oemof	PyPSA	Verhältnis $\frac{oemof}{PyPSA}$
Anzahl Iterationen	43.795	104.636	0,42
Gesamtsystemkosten [€]	4.616.240.279,43	47.865.019.980,00	0,10
Anzahl Beschränkungen	306.604	473.042	0,65
Anzahl Nonzeros	105.126	106.945	0,98
Anzahl von Variablen	411.729	210.253	1,96
Wallclock time [s]	21,72	244,78	0,09

Die Analyse der von den Simulationen erzeugten Lastgänge weist ebenfalls starke Veränderungen gegenüber den vorherigen Szenarien auf. So wird die installierte Leistung der Windkraft- und PV-Anlagen in beiden Simulationen deutlich ausgebaut, sodass die PV-Anlagen in der *PyPSA*-Simulation eine maximale Leistung von 3470 MW und in der *oemof*-Simulation von 5580 MW aufweisen. Die installierte Leistung der Windkraftanlagen erhöht sich bei *oemof* auf 6000 MW, in *PyPSA* auf 2870 MW (s. Abbildung 10).

Hierbei kommt es zu einer deutlichen Überproduktion gegenüber des Strombedarfes, was dazu führt, dass die elektrischen Speichermöglichkeiten *est* und Heizkessel *p2h* und *hp* für die Deckung des Wärmebedarfes ausgebaut werden. Außerdem fällt auf, dass *oemof* fast die Hälfte der erzeugten elektrischen Energie in die *excess*-Senke fließen, sodass diese aus dem Bilanzkreis des Modells herausfließt (s. Anlage 12 und Anlage 13).

Vergleich von oemof und PyPSA der installierten Leistung von pv1 und won1 in den drei Szenarien

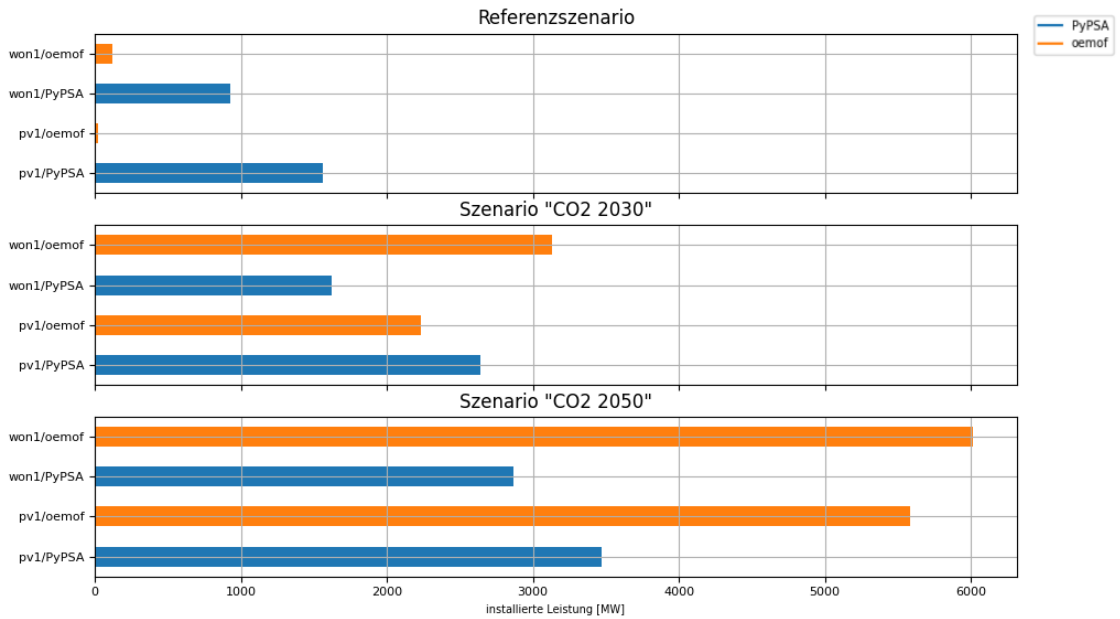


Abbildung 10: Vergleich von oemof und PyPSA der Installierten Leistung von pv1 und won1 in den drei Szenarien

Im Wärmesektor wird der Bedarf unterschiedlich von den beiden Simulationstools gedeckt. In der *oemof*-Simulation wird der Wärmebedarf hauptsächlich durch die PTH-Methode und zu 1/6 jeweils von dem mit Gas betriebenen Heizkessel sowie dem KWKK *chp2* gedeckt (s. Anlage 14). Anders als in dem vorangegangenen Szenario (Kapitel 5.5.2) wird nur ein minimaler Anteil gegenüber diesem („CO2 2030“: 138 GWh; „CO2 2050“: 24 GWh) in die *excess*-Senke des *oemof*-ES abgeleitet (s. Abbildung 11).

Vergleich von oemof und PyPSA der übertragenen Leistung in die *excess*-Senke in Szenario "CO2 2030" und "CO2 2050"

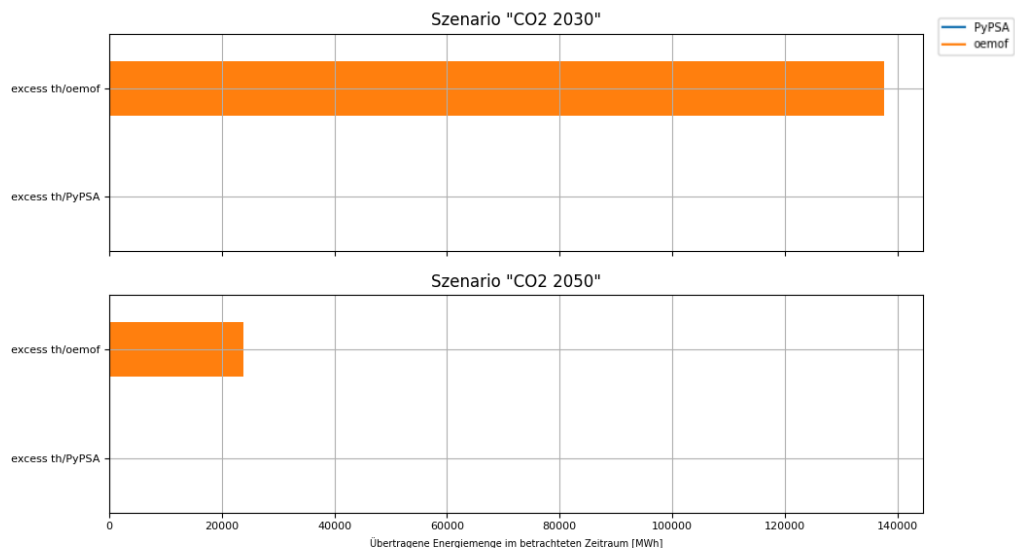


Abbildung 11: Vergleich von oemof und PyPSA der übertragenen Leistung in die *excess*-Senke in Szenario "CO2 2030" und "CO2 2050"

Die *PyPSA*-Simulation greift auch auf die PTH-Technologie zu, jedoch wird der größte Teil durch von außerhalb des betrachteten Systems erzeugte, importierte, Wärme *imported_heat* gedeckt. Weitere Kraftwerke, die in *PyPSA* mitwirken sind die beiden KWKK *chp2* und *chp3* (s. Anlage 15). Allerdings wird keine Wärme aus dem System durch die excess-Senke abgeführt (s. Abbildung 11).

5.6 Interpretation der Simulationsergebnisse

Wie in Kapitel 5.5 aufgezeigt, sind trotz derselben verwendeten Eingabedaten sehr unterschiedliche Ergebnisse der ES-Optimierungen in den drei Szenarien zu beobachten. Beim direkten Vergleich aller Szenarien fällt auf, dass trotz identischer Parametrisierung, erhebliche Unterschiede festzustellen sind. Dies gilt sowohl für die Zielgröße (ermittelte Kosten), als auch für den zeitlichen Umfang sowie die, von den Modellen, generierte Anzahl an Beschränkungen. Dieser Unterschied lässt sich zunächst einmal mit damit erklären, dass sich die Komponenten in den beiden ESMF nicht exakt gleich darstellen lassen (s. dazu Kapitel 5.3 *Modellkomponenten*). So können durch die Verwendung des *dict*-Datentyps der In- und Output in *oemof* deutlich einfacher definiert werden, besonders dann, wenn es mehr als einen Ein- oder Ausgang zu definieren gibt. In *PyPSA* müssen alle ES-Komponenten mit mehr als einem In- bzw. Output als *Link*-Komponente modelliert werden, wozu zunächst eine Erweiterung der Parameterliste für diese Komponente definiert werden muss (s. Kapitel 5.3.4). Hierdurch können nichtkonvexe-Parameter wie minimale Stillstandszeit, Anfahrkosten etc. nicht modelliert werden, wodurch die Kraftwerksmodelle an Detailgrad verlieren. Außerdem lässt sich durch die nicht-generische Form der *Link*-Komponente die Optimierung dieser nicht vollends nachvollziehen, was die deutlichen Unterschiede zwischen den ESM von *PyPSA* und *oemof* verursacht haben könnte. Für die Vergleichbarkeit wurden die nicht modellierbaren Parameter der *Link*-Komponente in dem *oemof*-ES ebenfalls nicht berücksichtigt.

Besonders auffällig in der Analyse der Zeitreihen des Lastganges der einzelnen Komponenten ist, dass *oemof* im Referenzszenario deutlich weniger Kraftwerke aktiv hält und diese somit im direkten Vergleich mit der Komponente in *PyPSA* aufsummiert mehr Energie erzeugen. *Oemof* scheint in der Optimierung die Gesamtsystemkosten deutlich besser zu optimieren, was sich auch in Tabelle 5 niederschlägt. *PyPSA* hingegen investiert bereits im Referenzszenario in die Speicherkomponente und erneuerbare Energien, obwohl dies deutlich teurer im Vergleich dazu ist, bestehende Kraftwerke zu nutzen. In den Szenarien mit einer Beschränkung des CO₂-Limits hingegen investiert *oemof* deutlich mehr in Windkraft („CO₂ 2030“ und „CO₂ 2050“) und PV („CO₂ 2050“), als *PyPSA* und installiert eine deutlich höhere Leistungskapazität. *PyPSA* agiert jedoch etwas zurückhaltender bei diesen beiden Szenarien, was die Investitionen in die sehr gering CO₂-ausstoßenden Technologien angeht. Dies liegt u.a. daran, dass *PyPSA* die excess-Senke in keinem der Szenarien weder im Strom- noch im Wärmesektor nutzt und so in die Speichertechnologien investiert, sowie die erzeugte Strom- und Wärmemenge möglichst geringhält. In Szenario „CO₂ 2050“ nutzt *PyPSA* sogar die mit sehr hohen Kosten verbundene Quelle *imported heat* nutzt (s. Anlage 14).

Die Laufzeitunterschiede der Berechnung lassen sich u.a. mit der Verwendung des *pyomo*-Moduls [105] in *PyPSA* erklären, welches laut der Dokumentation erhöhte Leistungsanforderungen an den PC stellt und längere Simulationszeiten erfordert [106]. Hinzu kommt, dass durch die höhere Anzahl von

Beschränkungen mehr Iterationsschritte zur Optimierung nötig sind, was die Laufzeit der Simulation ebenfalls erhöht.

Zu den genannten Modellunterschieden ist außerdem nicht auszuschließen, dass die vorgenommene Parametrierung, in ihrer Auswirkung auf die Modelle, nicht exakt identisch ist. Durch die teilweise erheblich unterschiedlichen Schnittstellen, zwischen Oemof und PyPSA, kann nicht mit abschließender Sicherheit ausgeschlossen werden, dass die Unterschiede, zumindest zum Teil, darin ihre Ursache finden. Die hohe Komplexität des modellierten Energiesystems begünstigt diesen Umstand zudem.

6 Zusammenfassung

Die Zusammenfassung umfasst das Fazit aus den Erkenntnissen der Literaturrecherche bezüglich der Charakterisierung von ESM(F/G), der daraus resultierenden Entwicklung eines Datenblattes und der Anwendung dessen in dem Fallbeispiel des Vergleiches von *oemof* und *PyPSA*. In dem Fallbeispiel wird der Vergleich beider Frameworks sowohl mit genanntem Datenblatt als auch anhand der Modellierung des ES Hamburg vollzogen. Zusätzlich zeigt dieses Kapitel wie die Synergiepotenziale von Frameworks genutzt werden können und gibt einen Ausblick auf die Forschung bezüglich FOSS im Bereich der ES-Modellierung.

6.1 Fazit

Ziel der Arbeit war es zunächst mögliche Vergleichskriterien von ESM aus der gängigen Literatur zusammenzufassen und aus diesen ein Datenblatt zu entwickeln. Hierbei wurde die Literatur chronologisch von den Anfängen der ESM bis heute durchlaufen und die Zusammenhänge und Unterschiede der Veröffentlichungen anhand der Vergleichsparameter aufgeführt.

Die Literaturrecherche bezüglich der Klassifizierung zeigte zunächst, dass es auf diesem Gebiet der Klassifizierung und dem Vergleich von ESM-Programmen eine große Forschungscommunity gibt. So wurden bereits in den 1980er bzw. 1990er Jahren Klassifizierungsmerkmale u.a. von N.M.J.P. van Beek [29] oder R. Pandey [30] entwickelt, die eine wichtige Basis für diesem Bereich geschaffen haben. Im Laufe der Recherche kam es dazu, dass die Vergleichsparameter voneinander abwichen, da bei aktuelleren Veröffentlichungen die Parameter nicht mehr für die klassischen ESM entwickelt wurden, sondern sich auf ESMF oder ESGM bezogen. Daher ist die Abgrenzung dieser Begriffe außerordentlich wichtig für diese Thesis. So gewinnen die Begriffe ESMF bzw. ESGM in Hinsicht auf die OS-Bewegung in den letzten Jahren deutlich an Bedeutung hinzu. Dies könnte daran liegen, dass durch die Verbreitung im Internet und die verbesserten Speichermöglichkeiten von Anwendungen nicht mehr nur ein von dem Entwickler vorgegebener Rahmen an Möglichkeiten zur Modellierung eines ES, sondern auch ganze Baukästen (sog. *Frameworks*), die zuvor nur den Entwicklern selbst zugänglich waren, publiziert werden (können). Die Bedeutsamkeit der Verbindung der OS-Bewegung und der ESM(F/G) stellt bspw. Pfenninger in seinen Veröffentlichungen dar und drängt auf eine verstärkte Zusammenarbeit der Forschenden in der Modellierung von ES [4, 19, 88, 89].

Die Grundstruktur für das in dieser Thesis entwickelte Datenblatt für ESMF basiert auf den Publizierungen von Ringkjøb, Haugan et al. [36] und Amerighi, Ciorba et al. [32]. In Kapitel 2.4 werden diese zusammengefasst. Um das Datenblatt in einem konkreten Beispiel anwenden und die Aussagekraft dessen überprüfen zu können, wurden im zweiten Teil dieser Arbeit die beiden ESMF *oemof* und *PyPSA* anhand des generierten Datenblattes verglichen. Für diesen Vergleich mussten zunächst die beiden Frameworks mit Hilfe der Dokumentation und den von den Entwicklern veröffentlichten Berichten analysiert werden und die Merkmale der Vergleichskriterien in das Factsheet eingetragen werden. Der zweite Teil des Vergleiches beschäftigte sich mit der Modellanwendung auf das ES von Hamburg in der Strom- und Wärmeerzeugung, um die Erkenntnisse aus dem Datenblatt verifizieren zu können. Hierzu mussten die Eingabeparameter dieser verglichen

werden, um das ESM in dem jeweiligen Framework gleich zu gestalten. Der Vergleich wurde auf drei unterschiedliche Szenarien angewendet, welche sich jeweils in dem Limit der ausgestoßenen CO₂-Menge unterscheiden.

Die beiden Frameworks sind sich, laut des theoretischen Vergleiches, von den Basisinformationen sehr ähnlich (s. Kapitel 5.1): Beide wurden bzw. werden als FOSS von Forschergruppen aus Deutschland in der Programmiersprache Python entwickelt, können durch eine Minimierungsgleichung ein Energiesystem optimieren und wurden zur Energiesystemanalyse programmiert.

Der Vergleich durch das Datenblatt hat jedoch auch gezeigt, dass sich die scheinbar große Ähnlichkeit, die die Basisinformationen aufzeigen, nicht unbedingt in den weiteren Kategorien des Factsheets widerspiegelt. Denn in diesem Punkt besitzen *oemof* und *PyPSA* zueinander größere Abweichungen. Der größte Unterschied der beiden Frameworks findet sich in den Zielen der Entwickler. *PyPSA* wurde zur Analyse der Netzinfrastruktur entwickelt, wobei der Detailgrad der Modellierung einzelner Kraftwerke, die in dem ES simuliert werden, eine untergeordnete Rolle gegenüber derjenigen bei *oemof* einnehmen. Mit *PyPSA* können spezifisch zum elektrischen Übertragungsnetz Notfallanalysen zum (partikulären) Netzausfall, Spannungsamplituden, Spannungswinkel und Blindleistung des ES analysiert werden [2]. *Oemof* hingegen ist nach Hilpert, Kaldemeyer et al. ein ESMF, welches sich besonders für komplexe Sektorenübergreifende ES-Modellierung von urbanen ES, der minimalen geographischen Auflösung, hin zu transnationalen Verflechtungen des Energiesystems eignet [7].

Die Ergebnisse des Datenblattes wurden durch das Fallbeispiel bekräftigt, in welchem sich die *oemof*-Kraftwerksmodelle wesentlich einfacher zu modellieren und der Wärme- und Stromsektor ohne großen Aufwand in den Zusammenhang setzen ließen. Letzteres liegt an dem größeren Umfang bzw. der besseren Möglichkeit, den *oemof* gegenüber *PyPSA* bezüglich der Integration von mehreren Ein- und Ausgängen einer Energieumwandlungskomponente sowie der Modellierung von Lastgängen (*oemof-demandlib*), anbietet. Fluktuierende erneuerbare Energieerzeuger können sowohl mit der *oemof-feedinlib* als auch in *PyPSA* mit *atlite* realisieren werden.

Somit lässt sich abschließend resümieren, dass das in dieser Arbeit entwickelte Datenblatt in der Anwendung dem Benutzer einen sinnvollen Überblick geben kann. Einzelne Parameter, wie etwa die Berechnungsdauer oder die Benutzerfreundlichkeit hängen jedoch stark von den Voraussetzungen und Wissen des Anwenders ab und haben daher keine Anwendung in dem Datenblatt gefunden.

6.2 Synergiepotential

Letztendlich kommt es auf die Anwendung des Frameworks an, welches der verglichenen Frameworks passender für den jeweiligen Modellierungsansatz ist. *oemof* erwies sich in der Fallstudie der ES-Modellierung, welche für die Thesis entwickelt wurde, als anwenderfreundlicher und passender von den Eigenschaften des Frameworks. Dies liegt vor allem daran, dass die spezifischen Emissionen direkt in der jeweiligen ES-Komponente deklariert werden und beliebig viele Ein- und Ausgangsbusse zu einer ES-Komponente hinzugefügt werden können. Zudem erwies sich die Modellierung des ES mit einem hohen Anteil an erneuerbaren Energien als ausgereifter und realistischer als ebendiese mit Hilfe des Frameworks *PyPSA*.

Wäre eine Analyse der elektrischen Netzinfrastruktur eines klar definierten Energiesystems das Ziel der Untersuchung durch die Fallstudie gewesen, so hätte sich *PyPSA* als geeigneter gegenüber *oemof* herausgestellt. So kommt die Frage auf, warum beide Frameworks nicht synchron zur Modellierung des ES genutzt werden können, sodass die Ergebnisse direkt verglichen und die Stärken der ESMF kumuliert werden. Die Intention mehrere Aspekte eines Energiesystems mit Hilfe der jeweiligen Stärken der verglichenen Frameworks inspizieren zu können, erfordert jedoch eine einheitliche Ein- und Ausgabestruktur. Hierdurch könnte etwa das Problem der Implementierung der Anzahl von Ausgängen der Generator-Komponente (s. Kapitel 5.3.4 Transformer) umgangen werden, indem dieses durch eine standardisierte Oberfläche zur Parametrierung des ESM Anwenderfreundlich kreiert wird. Durch den unterschiedlichen Fokus beider Frameworks bzw. mehrerer ESMF kann so ein Tool entstehen, welches eine Vielzahl von Forschungsfeldern der ES-Analyse abdecken kann. Dieses Synergiepotential wird etwa durch die FOSS *tessif* abgerufen, indem es für beide Frameworks eine einheitliche Ein- und Ausgabestruktur bietet und so das Potenzial beider bzw. mehrerer Frameworks nutzen kann, was dem Nutzer ein breiteres Feld an Möglichkeiten zur ES-Modellierung darlegt.

6.3 Ausblick

Durch immer komplexer werdende Netzwerk- und Energiesystemstrukturen mit neuen Technologien und der Sektorenkopplung etwa durch Elektroautos, *Smart Grids*, dezentrale Stromversorgung, Einspeisung und Abhängigkeit von fluktuierenden erneuerbaren Energien, muss die Simulation und Analyse von ES ständig angepasst und flächendeckend vorangetrieben werden. Dies wird u.a. von FOSS-Angeboten wie *oemof* und *PyPSA* erfüllt, da sich durch den freien Zugang die Hürde des Erwerbs solcher Software deutlich verringert und durch die umfangreiche Nutzung und Vernetzung mit Hilfe von Plattformen wie *openmod*, *Open Power System Data*, *Open Energy Platform*, *FfE Open Data Portal* oder *Open Power System Data* [3, 20–22, 41, 86] eine effizientere Zusammenarbeit von Forschern weltweit gewährleistet werden kann. Da es jedoch zu den zahlreichen Forschungsrichtungen in der Energietechnik eine große Zahl an entwickelter Software gibt, ist es notwendig FOSS zu sichten, durch eine Evaluierung der jeweiligen Software mächtigere Programme zu erschaffen und die Synergien mehrerer Programme zu nutzen, wie es u.a. das Projekt *open_MODEX* [5] tut. Hierzu sind geeignete Vergleichskriterien ein wichtiger Punkt, an dem diese Thesis einen Mehrwert bietet. Durch das in dieser Arbeit entwickelte Factsheet sowie das der *openmod*, Amerighi, Ciorba et al. und *Open Energy Platform* [3, 15, 32], können die bereits entwickelten Programme besser eingeordnet werden und ineffiziente Dualität in der Entwicklung von FOSS zur ESA verhindert werden, wie es Pfenninger, DeCarolis et al. in „*The importance of open data and software: Is energy research lagging behind?*“ [4] anprangern.

Im Zuge der Modellierungsprozesses wurde festgestellt, dass die Abbildung vergleichsweiser komplexer Energiesysteme, vor allem durch unterschiedliche Frameworks, Schwierigkeiten mit sich bringen. Für die zukünftige Modellierung ähnlich komplexer Energiesysteme empfiehlt sich daher, den Umfang inkrementell zu erhöhen. Dadurch kann die Wahrscheinlichkeit verringert werden, dass etwaige Unstimmigkeiten in der Parametrisierung auf Grund unterschiedlicher Schnittstellen, zu unerwarteten Differenzen in den Ergebnissen führen.

Quellen

- [1] K. van den Bergh, E. Delarue und W. D'haeseleer, „Improved zonal network models in generation scheduling“ in *2014 Power Systems Computation Conference (PSCC)*, Wrocław, Poland, 18.08.2014 - 22.08.2014, S. 1–7, doi: 10.1109/PSCC.2014.7038465.
- [2] T. Brown, J. Hörsch und D. Schlachtberger, „PyPSA: Python for Power System Analysis“, *Journal of Open Research Software*, Jg. 6, Nr. 3, S. 12, 2018, doi: 10.5334/jors.188.
- [3] *openmod - Open Energy Modelling Initiative*. [Online]. Verfügbar unter: <https://openmod-initiative.org/manifesto.html> (Zugriff am: 30. April 2021).
- [4] S. Pfenninger, J. DeCarolis, L. Hirth, S. Quoilin und I. Staffell, „The importance of open data and software: Is energy research lagging behind?“, *Energy Policy*, Jg. 101, S. 211–215, 2017, doi: 10.1016/j.enpol.2016.11.046.
- [5] M. Leich, „open_MODEX“, *Reiner Lemoine Institut*, 21. Feb. 2019, 2019. [Online]. Verfügbar unter: https://reiner-lemoine-institut.de/open_modex/. Zugriff am: 15. Oktober 2021.
- [6] Oemof Developer Group, *oemof.solph - a model generator for energy systems - v0.4.1*. Zenodo, 2020. [Online]. Verfügbar unter: <https://zenodo.org/record/3906081#.YDEMROhKjic>
- [7] S. Hilpert, C. Kaldemeyer, U. Krien, S. Günther, C. Wingenbach und G. Plessmann, „The Open Energy Modelling Framework (oemof) - A new approach to facilitate open science in energy system modelling“, *Energy Strategy Reviews*, Jg. 22, S. 16–25, 2018, doi: 10.1016/j.esr.2018.07.001.
- [8] *tum-ens/urbs: urbs v1.0.1*. Zenodo, 2019.
- [9] C. Bussar, P. Stöcker, L. Moraes, K. Jacqué, H. Axelsen und D. U. Sauer, „The Long-Term Power System Evolution – First Optimisation Results“, *Energy Procedia*, Jg. 135, S. 347–357, 2017, doi: 10.1016/j.egypro.2017.09.526.
- [10] T. Burandt, K. Löffler und K. Hainsch, „GENeSYS-MOD v2.0 - Enhancing the Global Energy System Model: Model improvements, framework changes, and European data set“, Berlin: Deutsches Institut für Wirtschaftsforschung (DIW), DIW Data Documentation 94, 2018. [Online]. Verfügbar unter: <https://www.econstor.eu/handle/10419/180395>.
- [11] F. Wiese *et al.*, „Baltimore open source energy system model“, *Energy Strategy Reviews*, Jg. 20, S. 26–34, 2018, doi: 10.1016/j.esr.2018.01.003.
- [12] *oemof | publications*. [Online]. Verfügbar unter: <https://oemof.org/publications/> (Zugriff am: 10. Mai 2021).
- [13] *pypsa | publications*. [Online]. Verfügbar unter: <https://pypsa.org/publications/> (Zugriff am: 10. Mai 2021).
- [14] F. Wiese, S. Hilpert, C. Kaldemeyer und G. Pleßmann, „A qualitative evaluation approach for energy system modelling frameworks“ (En;en), *Energy Sustain Soc*, Jg. 8, Nr. 1, S. 1–16, 2018, doi: 10.1186/s13705-018-0154-3.
- [15] OEP Community, *Open Energy Platform (OEP)*. [Online]. Verfügbar unter: <https://openenergy-platform.org/> (Zugriff am: 26. April 2021).
- [16] *History of the OSI | Open Source Initiative*. [Online]. Verfügbar unter: <https://opensource.org/history> (Zugriff am: 27. April 2021).
- [17] C. DiBona, M. Stone und D. Cooper, *Open Sources 2.0: The Continuing Evolution*. Sebastopol: O'Reilly Media, Inc, 2008.
- [18] *The Open Source Definition | Open Source Initiative*. [Online]. Verfügbar unter: <https://opensource.org/docs/osd> (Zugriff am: 14. Mai 2021).

- [19] S. Pfenninger *et al.*, „Opening the black box of energy modelling: Strategies and lessons learned“, *Energy Strategy Reviews*, Jg. 19, S. 63–71, 2018, doi: 10.1016/j.esr.2017.12.002.
- [20] *Data Platform – Open Power System Data*. [Online]. Verfügbar unter: <https://data.open-power-system-data.org/> (Zugriff am: 29. Mai 2021).
- [21] *opendata.ffe.de – The FfE Open Data Portal*. [Online]. Verfügbar unter: <http://opendata.ffe.de/> (Zugriff am: 29. Mai 2021).
- [22] *OEP - Database*. [Online]. Verfügbar unter: <https://openenergy-platform.org/dataedit/schemas> (Zugriff am: 29. Mai 2021).
- [23] J. Nagel, *Optimization of Energy Supply Systems: Modelling, Programming and Analysis*. Cham: Springer International Publishing, 2019.
- [24] K. Mainzer, *Analyse und Optimierung urbaner Energiesysteme - Entwicklung und Anwendung eines übertragbaren Modellierungswerkzeugs zur nachhaltigen Systemgestaltung*. Karlsruhe, 2019.
- [25] E. Worrell, S. Ramesohl und G. Boyd, „ADVANCES IN ENERGY FORECASTING MODELS BASED ON ENGINEERING ECONOMICS“, *Annu. Rev. Environ. Resour.*, Jg. 29, Nr. 1, S. 345–381, 2004, doi: 10.1146/annurev.energy.29.062403.102042.
- [26] A. Herbst, F. Toro, F. Reitze und E. Jochem, „Introduction to Energy Systems Modelling“ (En;en), *Swiss J Economics Statistics*, Jg. 148, Nr. 2, S. 111–135, 2012, doi: 10.1007/BF03399363.
- [27] M. Greenberger und R. Richels, „Assessing Energy Policy Models: Current State and Future Directions“, *Annu. Rev. Energy.*, Jg. 4, Nr. 1, S. 467–500, 1979, doi: 10.1146/annurev.eg.04.110179.002343.
- [28] M. Grubb, J. Edmonds, P. ten Brink und M. Morrison, „The Costs of Limiting Fossil-Fuel CO₂ Emissions: A Survey and Analysis“, *Annu. Rev. Energy. Environ.*, Jg. 18, Nr. 1, S. 397–478, 1993, doi: 10.1146/annurev.eg.18.110193.002145.
- [29] N.M.J.P. van Beeck, „Classification of Energy Models“ (English), 1999. [Online]. Verfügbar unter: <https://research.tilburguniversity.edu/en/publications/classification-of-energy-models>
- [30] R. Pandey, „Energy policy modelling: agenda for developing countries“, *Energy Policy*, Jg. 30, Nr. 2, S. 97–106, 2002, doi: 10.1016/S0301-4215(01)00062-3.
- [31] D. Connolly, H. Lund, B. V. Mathiesen und M. Leahy, „A review of computer tools for analysing the integration of renewable energy into various energy systems“, *Applied Energy*, Jg. 87, Nr. 4, S. 1059–1082, 2010, doi: 10.1016/j.apenergy.2009.09.026.
- [32] O. Amerighi, U. Ciorba und M. C. Tommasino, „Analysing Transition Planning and Systemic Energy Planning Tools for the implementation of the Energy Technology Information System: Models Characterization Report“, ENEA, Italien, Okt. 2010. [Online]. Verfügbar unter: http://www.cres.gr/atest/pdf/D_2_1_Models_Characterisation_Report.pdf. Zugriff am: 25. April 2021.
- [33] P. Lopion, P. Markewitz, M. Robinius und D. Stolten, „A review of current challenges and trends in energy systems modeling“, *Renewable and Sustainable Energy Reviews*, Jg. 96, S. 156–166, 2018, doi: 10.1016/j.rser.2018.07.045.
- [34] L. M. Hall und A. R. Buckley, „A review of energy systems models in the UK: Prevalent usage and categorisation“, *Applied Energy*, Jg. 169, S. 607–628, 2016, doi: 10.1016/j.apenergy.2016.02.044.
- [35] K. Bouw, K. J. Noorman, C. J. Wiekens und A. Faaij, „Local energy planning in the built environment: An analysis of model characteristics“, *Renewable and Sustainable Energy Reviews*, Jg. 144, S. 111030, 2021, doi: 10.1016/j.rser.2021.111030.

- [36] H.-K. Ringkjøb, P. M. Haugan und I. M. Solbrekke, „A review of modelling tools for energy and electricity systems with large shares of variable renewables“, *Renewable and Sustainable Energy Reviews*, Jg. 96, S. 440–459, 2018, doi: 10.1016/j.rser.2018.08.002.
- [37] L. Kriechbaum, G. Scheiber und T. Kienberger, „Grid-based multi-energy systems—modelling, assessment, open source modelling frameworks and challenges“, *Energ Sustain Soc*, Jg. 8, Nr. 1, 2018, doi: 10.1186/s13705-018-0176-x.
- [38] S. Auer, J. Liße, S. R. Mandha und C. Horn, „Power-flow-constrained asset optimization for off-grid power systems using selected Open-Source frameworks“ in *Hybrid Power Systems Workshop 2019*, 2019.
- [39] S. Lechtenboehmer, T. Pregger und A. Palzer, „RegMex - Modellexperimente und -vergleiche zur Simulation von Wegen zu einer vollstaendig regenerativen Energieversorgung“, 2018. [Online]. Verfügbar unter: https://epub.wupperinst.org/frontdoor/deliver/index/docId/7096/file/7096_RegMex.pdf
- [40] J. Després, N. Hadjsaid, P. Criqui und I. Noirot, „Modelling the impacts of variable renewable sources on the power sector: Reconsidering the typology of energy modelling tools“, *Energy*, Jg. 80, S. 486–495, 2015, doi: 10.1016/j.energy.2014.12.005.
- [41] *Open Models - wiki.openmod-initiative.org*. [Online]. Verfügbar unter: https://wiki.openmod-initiative.org/wiki/Open_Models (Zugriff am: 13. Mai 2021).
- [42] EnArgus, *Solver*. [Online]. Verfügbar unter: https://www.enargus.de/pub/bscw.cgi/d11774-2/*/*/*Solver.html?op=Wiki.getwiki&search=model (Zugriff am: 21. Mai 2021).
- [43] B. Meindl und M. Templ, *Analysis of commercial and free and open source solvers for linear optimization problems*, 2013. [Online]. Verfügbar unter: https://www.researchgate.net/profile/matthias_templ/publication/265117825_analysis_of_commercial_and_free_and_open_source_solvers_for_linear_optimization_problems
- [44] J. Keirstead, M. Jennings und A. Sivakumar, „A review of urban energy system models: Approaches, challenges and opportunities“, *Renewable and Sustainable Energy Reviews*, Jg. 16, Nr. 6, S. 3847–3866, 2012, doi: 10.1016/j.rser.2012.02.047.
- [45] J.C. Hourcade *et al.*, „Estimating the costs of mitigating greenhouse gases“ (English), *Climate change 1995. Economic and social dimensions of climate change*, S. 263–296, 1996. [Online]. Verfügbar unter: <https://orbit.dtu.dk/en/publications/estimating-the-costs-of-mitigating-greenhouse-gases>
- [46] J.-C. Hourcade, M. Jaccard, C. Bataille und F. Ghersi, „Hybrid Modeling: New Answers to Old Challenges Introduction to the Special Issue of The Energy Journal“, *EJ*, SI2006, Nr. 01, 2006, doi: 10.5547/ISSN0195-6574-EJ-VolSI2006-NoSI2-1.
- [47] C. Bataille, *Design and application of a technologically explicit hybrid energy-economy policy model with micro and macro economic dynamics*. Ottawa: Library and Archives Canada = Bibliothèque et Archives Canada, 2006.
- [48] P. Fortes, R. Pereira, A. Pereira und J. Seixas, „Integrated technological-economic modelling platform for energy and climate policy analysis“, 17. Juli 2014. [Online]. Verfügbar unter: <http://dx.doi.org/10.1016/j.energy.2014.06.075>. Zugriff am: 30. Mai 2021.
- [49] T. Ommen, W. B. Markussen und B. Elmegaard, „Comparison of linear, mixed integer and non-linear programming methods in energy system dispatch modelling“, *Energy*, Jg. 74, S. 109–118, 2014, doi: 10.1016/j.energy.2014.04.023.
- [50] *Energy terminology: A multilingual glossary = Terminologie de l'energie : dictionnaire multilingue = Energieterminologie : ein mehrsprachiges Wörterbuch = Terminologia de la energia : un glosario multilingüe*, 2. Aufl. Oxford: Pergamon Press, 1986.

- [51] B. Pfluger, *Assessment of least-cost pathways for decarbonising Europe's power supply: a model-based long-term scenario analysis accounting for the characteristics of renewable energies*. Karlsruhe: KIT Scientific Publishing, 2014.
- [52] M. Ventosa, Á. Baíllo, A. Ramos und M. Rivier, „Electricity market modeling trends“, *Energy Policy*, Jg. 33, Nr. 7, S. 897–913, 2005, doi: 10.1016/j.enpol.2003.10.013.
- [53] R. Villasana, L. Garver und S. Salon, „Transmission Network Planning Using Linear Programming“, *IEEE Trans. on Power Apparatus and Syst.*, PAS-104, Nr. 2, S. 349–356, 1985, doi: 10.1109/TPAS.1985.319049.
- [54] W. Medjroubi, U. P. Müller, M. Scharf, C. Matke und D. Kleinhans, „Open Data in Power Grid Modelling: New Approaches Towards Transparent Grid Models“, *Energy Reports*, Jg. 3, S. 14–21, 2017, doi: 10.1016/j.egy.2016.12.001.
- [55] S. Chatzivasileiadis, „Lecture Notes on Optimal Power Flow (OPF)“, 2. Nov. 2018. [Online]. Verfügbar unter: <http://arxiv.org/pdf/1811.00943v1>.
- [56] openmod initiative, *Property:Network coverage - wiki.openmod-initiative.org*. [Online]. Verfügbar unter: https://wiki.openmod-initiative.org/index.php?title=Property:Network_coverage&from=OSeMOSYS#SMWResults (Zugriff am: 21. Mai 2021).
- [57] GNU-Projekt - Free Software Foundation, *Copyleft. Pragmatischer Idealismus*. [Online]. Verfügbar unter: <http://www.gnu.org/philosophy/pragmatic.html> (Zugriff am: 15. Oktober 2021).
- [58] M. Lang, „Open-Source-Lizenzen rechtskonform einsetzen“, *Dev-Insider*, 5. Okt. 2017, 2017. [Online]. Verfügbar unter: <https://www.dev-insider.de/open-source-lizenzen-rechtskonform-einsetzen-a-604458/>. Zugriff am: 17. Mai 2021.
- [59] M. McNutt, „Journals unite for reproducibility“ (eng), *Science (New York, N.Y.)*, Jg. 346, Nr. 6210, S. 679, 2014, doi: 10.1126/science.aaa1724.
- [60] Universität Innsbruck, *Zitationsanalysen*. [Online]. Verfügbar unter: <https://www.uibk.ac.at/ulb/services/zitationsanalysen.html> (Zugriff am: 17. Mai 2021).
- [61] M. Leich, „oemof“, *Reiner Lemoine Institut*, 5. Jan. 2015, 2015. [Online]. Verfügbar unter: <https://reiner-lemoine-institut.de/oemof/>. Zugriff am: 14. Mai 2021.
- [62] GitHub, *oemof community*. [Online]. Verfügbar unter: <https://github.com/oemof> (Zugriff am: 14. Mai 2021).
- [63] C. Wingenbach, S. Hilpert und S. Günther, „Entwicklung eines Open Source Energie-Modells für Schleswig-Holstein (openMod. sh)“, Europa-Universität Flensburg, Flensburg, 30. Juni 2017. [Online]. Verfügbar unter: <https://www.uni-flensburg.de/fileadmin/content/abteilungen/industrial/dokumente/downloads/veroeffentlichungen/forschungsergebnisse/openmodsh-endbericht-1.pdf>.
- [64] The Atlite Authors, *Atlite: Convert weather data to energy systems data — atlite 0.2.3.dev6+g1476c43 documentation*. [Online]. Verfügbar unter: <https://atlite.readthedocs.io/en/latest/index.html> (Zugriff am: 24. Mai 2021).
- [65] Copernicus Climate Change Service, „ERA5-Land hourly data from 2001 to present“, 2019.
- [66] E. Kalnay *et al.*, „The NCEP/NCAR 40-Year Reanalysis Project“, *Bull. Amer. Meteor. Soc.*, Jg. 77, Nr. 3, S. 437–471, 1996, doi: 10.1175/1520-0477(1996)077%3C0437:TNYRP%3E2.0.CO;2.
- [67] National Centers for Environmental Prediction/National Weather Service/NOAA/U.S. Department of Commerce, *NCEP/NCAR Global Reanalysis Products, 1948-continuing*. [Online]. Verfügbar unter: <https://psl.noaa.gov/data/gridded/data.ncep.reanalysis.html> (Zugriff am: 15. Oktober 2021).

- [68] BAMS, *The NCEP/NCAR 40-Year Reanalysis Project*. [Online]. Verfügbar unter: <https://rda.ucar.edu/datasets/ds090.0/docs/publications/bams1996mar/?time=d> (Zugriff am: 15. Oktober 2021).
- [69] R. Müller, U. Pfeifroth, C. Träger-Chatterjee, R. Cremer, J. Trentmann und R. Hollmann, „Surface Solar Radiation Data Set - Heliosat (SARAH) - Edition 1“, 2015.
- [70] R. Müller, U. Pfeifroth, C. Träger-Chatterjee, J. Trentmann und R. Cremer, „Digging the METEOSAT Treasure—3 Decades of Solar Surface Radiation“, *Remote Sensing*, Jg. 7, Nr. 6, S. 8067–8101, 2015, doi: 10.3390/rs70608067.
- [71] Bundesnetzagentur, *MaStR*. [Online]. Verfügbar unter: <https://www.marktstammdatenregister.de/MaStR> (Zugriff am: 16. Oktober 2021).
- [72] M. N. I. Maruf, „A Novel Method for Analyzing Highly Renewable and Sector-Coupled Subnational Energy Systems—Case Study of Schleswig-Holstein“, *Sustainability*, Jg. 13, Nr. 7, S. 3852, 2021, doi: 10.3390/su13073852.
- [73] M. N. I. Maruf, „Open model-based analysis of a 100% renewable and sector-coupled energy system—The case of Germany in 2050“, *Applied Energy*, Jg. 288, S. 116618, 2021, doi: 10.1016/j.apenergy.2021.116618.
- [74] S. Hilpert, F. Dettner und A. Al-Salaymeh, „Analysis of Cost-Optimal Renewable Energy Expansion for the Near-Term Jordanian Electricity System“, *Sustainability*, Jg. 12, Nr. 22, S. 9339, 2020, doi: 10.3390/su12229339.
- [75] S. Hilpert, „Effects of Decentral Heat Pump Operation on Electricity Storage Requirements in Germany“, *Energies*, Jg. 13, Nr. 11, S. 2878, 2020, doi: 10.3390/en13112878.
- [76] C. Kaldemeyer, C. Boysen und I. Tuschy, „Compressed Air Energy Storage in the German Energy System – Status Quo & Perspectives“, *Energy Procedia*, Jg. 99, S. 298–313, 2016, doi: 10.1016/j.egypro.2016.10.120.
- [77] C. Boysen, H. Grotlüschen, H. Großer, C. Kaldemeyer und I. Tuschy, „Druckluftspeicherkraftwerk Schleswig-Holstein: Untersuchung zur Eignung Schleswig-Holsteins als Modellstandort für die Energiewende“, Europa-Universität Flensburg, Flensburg, ZNES-Reihe; Forschungsergebnisse, 2017. [Online]. Verfügbar unter: https://www.uni-flensburg.de/fileadmin/content/abteilungen/industrial/dokumente/downloads/veroeffentlichungen/forschungsergebnisse/dlsh-forschungsbeitraege5-20180126.pdf?sword_list=da&no_cache=1. Zugriff am: 30. Mai 2021.
- [78] J. Gorenstein Dedecca, S. Lumbreras, A. Ramos, R. A. Hakvoort und P. M. Herder, „Expansion planning of the North Sea offshore grid: Simulation of integrated governance constraints“, *Energy Economics*, Jg. 72, S. 376–392, 2018, doi: 10.1016/j.eneco.2018.04.037.
- [79] J. Hörsch, F. Hofmann, D. Schlachtberger und T. Brown, „PyPSA-Eur: An Open Optimisation Model of the European Transmission System“, *Energy Strategy Reviews*, Jg. 22, Nr. 4, S. 207–215, 2018, doi: 10.1016/j.esr.2018.08.012.
- [80] D. P. Schlachtberger, T. Brown, M. Schäfer, S. Schramm und M. Greiner, „Cost optimal scenarios of a future highly renewable European electricity system: Exploring the influence of weather data, cost parameters and policy constraints“, *Energy*, Jg. 163, Nr. 0, S. 100–114, 2018, doi: 10.1016/j.energy.2018.08.070.
- [81] M. Schlott, A. Kies, T. Brown, S. Schramm und M. Greiner, „The Impact of Climate Change on a Cost-Optimal Highly Renewable European Electricity Network“, *Applied Energy*, Jg. 230, Nr. 7609, S. 1645–1659, 2018, doi: 10.1016/j.apenergy.2018.09.084.

- [82] F. Hofmann, M. Schäfer, T. Brown, J. Hörsch, S. Schramm und M. Greiner, „Principal Flow Patterns across renewable electricity networks“, *EPL*, Jg. 124, Nr. 1, S. 18005, 2018, doi: 10.1209/0295-5075/124/18005.
- [83] J. Hörsch und J. Calitz, „PyPSA-ZA: Investment and operation co-optimization of integrating wind and solar in South Africa at high spatial and temporal detail“, 30. Okt. 2017. [Online]. Verfügbar unter: <http://arxiv.org/pdf/1710.11199v1>.
- [84] G. Markus und G. Alexandre, „Impact of high renewable penetration scenarios on system reliability: two case studies in the Kingdom of Saudi Arabia“, 12. Sep. 2017. [Online]. Verfügbar unter: <http://arxiv.org/pdf/1709.03761v1>.
- [85] Stromnetz Hamburg GmbH, *Energieportal Hamburg*. [Online]. Verfügbar unter: <https://www.energieportal-hamburg.de/> (Zugriff am: 24. Mai 2021).
- [86] Open Power System Data, *Open Power System Data – A platform for open data of the European power system*. [Online]. Verfügbar unter: <https://open-power-system-data.org/> (Zugriff am: 26. Mai 2021).
- [87] S. Pfenninger und I. Staffell, *Renewables.ninja*. [Online]. Verfügbar unter: <https://www.renewables.ninja/> (Zugriff am: 26. Mai 2021).
- [88] S. Pfenninger und I. Staffell, „Long-term patterns of European PV output using 30 years of validated hourly reanalysis and satellite data“, *Energy*, Jg. 114, S. 1251–1265, 2016, doi: 10.1016/j.energy.2016.08.060.
- [89] I. Staffell und S. Pfenninger, „Using bias-corrected reanalysis to simulate current and future wind power output“, *Energy*, Jg. 114, S. 1224–1239, 2016, doi: 10.1016/j.energy.2016.08.068.
- [90] M. M. Rienecker *et al.*, „MERRA: NASA’s Modern-Era Retrospective Analysis for Research and Applications“, *Journal of Climate*, Jg. 24, Nr. 14, S. 3624–3648, 2011, doi: 10.1175/JCLI-D-11-00015.1.
- [91] Stromnetz Hamburg GmbH, *Energieportal Hamburg - Stromwetter*. [Online]. Verfügbar unter: <https://www.energieportal-hamburg.de/stromwetter> (Zugriff am: 24. Mai 2021).
- [92] Wärme Hamburg, *Erzeugungsanlagen*. [Online]. Verfügbar unter: <https://waerme.hamburg/fernwaermesystem/erzeugungsanlagen> (Zugriff am: 26. Mai 2021).
- [93] *Wedel | Vattenfalls Kraftwerke*. [Online]. Verfügbar unter: <https://web.archive.org/web/20120618113607/http://kraftwerke.vattenfall.de/powerplant/wedel> (Zugriff am: 26. Mai 2021).
- [94] Müllverwertung Borsigstraße GmbH, *MVB*. [Online]. Verfügbar unter: <http://www.mvb-hh.de/> (Zugriff am: 26. Mai 2021).
- [95] Stadtreinigung Hamburg, *Verbund Hamburger Abfallverwertungsanlagen*. [Online]. Verfügbar unter: <https://www.stadtreinigung.hamburg/export/sites/default/download/PDF/Verbund-MVA.pdf> (Zugriff am: 26. Mai 2021).
- [96] J. Weibezahn, R. Weinhold, C. Gerbaulet und F. Kunz, *Conventional power plants list*. [Online]. Verfügbar unter: https://data.open-power-system-data.org/conventional_power_plants/ (Zugriff am: 26. Mai 2021).
- [97] Umweltbundesamt, *Datenbank "Kraftwerke in Deutschland"*. [Online]. Verfügbar unter: <https://www.umweltbundesamt.de/dokument/datenbank-kraftwerke-in-deutschland> (Zugriff am: 26. Mai 2021).
- [98] V. Quaschnig, *Regenerative Energiesysteme: Technologie - Berechnung - Klimaschutz*, 10. Aufl. München: Hanser, 2019.
- [99] T. Schwarzböck, „Bestimmung der fossilen Kohlendioxidemissionen aus Österreichischen Müllverbrennungsanlagen (BEFKÖM)“, Institut für Wassergüte, Ressourcenmanagement und

- Abfallwirtschaft, Wien, 2015. [Online]. Verfügbar unter:
<https://www.wien.gv.at/umweltschutz/nachhaltigkeit/pdf/schwarzboeck-2015.pdf>. Zugriff am: 26. Mai 2021.
- [100] Umweltbundesamt, *Energiebedingte Emissionen*. [Online]. Verfügbar unter:
<https://www.umweltbundesamt.de/daten/energie/energiebedingte-emissionen#energiebedingte-emissionen-durch-stromerzeugung> (Zugriff am: 26. Mai 2021).
- [101] Statistische Ämter des Bundes und der Länder | Gemeinsames Statistikportal, *Umweltökonomische Gesamtrechnungen der Länder: Daten zum Emissionen von Gasen in den Bundesländern*. [Online]. Verfügbar unter:
<https://www.statistikportal.de/de/ugrdl/ergebnisse/gase#alle-ergebnisse> (Zugriff am: 26. Mai 2021).
- [102] Fraunhofer-Institut für Solare Energiesysteme ISE, *Studie: Stromgestehungskosten erneuerbare Energien - Fraunhofer ISE*. [Online]. Verfügbar unter:
<https://www.ise.fraunhofer.de/de/veroeffentlichungen/studien/studie-stromgestehungskosten-erneuerbare-energien.html> (Zugriff am: 24. Februar 2021).
- [103] U. Krien, P. Schönfeldt, J. Launer, S. Hilpert, C. Kaldemeyer und G. Pleßmann, *User's guide — oemof.solph 0.4.4 documentation: ExtractionTurbineCHP (component)*. [Online]. Verfügbar unter: <https://oemof-solph.readthedocs.io/en/v0.4.4/usage.html#extractionturbinechp-component> (Zugriff am: 30. September 2021).
- [104] T. Brown, *GitHub - PyPSA/PyPSA: PyPSA: Python for Power System Analysis*. [Online]. Verfügbar unter: <https://github.com/PyPSA/PyPSA/blob/master/examples/sector-coupling/chp-fixed-heat-power-ratio.py> (Zugriff am: 30. September 2021).
- [105] W. E. Hart *et al.*, *Pyomo — Optimization Modeling in Python*. Cham: Springer International Publishing, 2017.
- [106] *Optimal Power Flow — PyPSA 0.18.0 documentation*. [Online]. Verfügbar unter: https://pypsa.readthedocs.io/en/latest/optimal_power_flow.html#overview (Zugriff am: 6. September 2021).

Anlagen

Anlage 1

Berechnung der spez. CO₂-Emissionen von Hausmüll

$$23 \text{ t}_{\text{Müll}}/\text{h} \cdot 7,5 \text{ GJ}/\text{t}_{\text{Müll}} = 172500 \text{ GJ}/\text{h} = 47,92 \text{ GW} = 47916,67 \text{ MW} \quad (3)$$

$$\eta_{el} = \frac{29 \text{ MW}}{47916,67 \text{ MW}} = 0,0006 \quad (4)$$

$$\eta_{th} = \frac{70 \text{ MW}}{47916,67 \text{ MW}} = 0,0015 \quad (5)$$

Spezifische CO₂-Emissionen von Hausmüll:

$$42,6 \text{ Kg}_{\text{CO}_2}/\text{GJ} = 0,0426 \text{ t}_{\text{CO}_2}/\text{GJ} \quad [99]$$

$$\text{CO}_2\text{Emissions}_{\text{per hour, Müll}} = 172500 \text{ GJ}/\text{h} \cdot 0,0426 \text{ t}_{\text{CO}_2}/\text{GJ} = 7328,5 \text{ t}_{\text{CO}_2}/\text{h} \quad (6)$$

$$\text{CO}_2\text{Emissions}_{\text{spez, Müll}} = \frac{7348,5 \text{ t}_{\text{CO}_2}/\text{h}}{47916,67 \text{ MW}} = 0,15336 \text{ t}_{\text{CO}_2}/\text{MWh} \quad (7)$$

Anlage 2

Emissionen ausgewählter Treibhausgase in Deutschland nach Kategorien in Tsd. t Kohlendioxid-Äquivalenten*																							
Kategorie	Stoff	1990	1991	1992	1993	1994	1995	2000	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	
Energiebedingte Emissionen	CO ₂	989.470	955.417	910.148	900.266	881.456	881.166	839.481	811.815	823.022	797.731	803.380	746.979	784.388	760.739	766.024	783.628	744.818	749.275	752.475	735.109	704.109	
	CH ₄	40.523	37.898	34.588	35.433	31.897	30.609	25.014	15.309	13.763	12.896	12.838	11.523	11.740	13.111	12.693	11.774	12.146	11.665	11.702	11.665	11.702	10.738
	N ₂ O	7.055	6.713	6.448	6.372	6.243	6.254	5.558	4.953	5.042	5.146	5.146	5.214	4.984	5.311	5.410	5.485	5.497	5.380	5.508	5.581	5.565	5.437
	Summe	1.037.048	1.000.029	951.184	942.071	919.597	918.029	870.053	832.077	841.827	815.772	821.431	763.487	801.508	777.890	784.620	801.819	761.973	766.929	769.721	752.376	720.284	
Anteil an Gesamtemissionen		82,9 %	83,0 %	82,4 %	82,3 %	81,7 %	81,7 %	83,3 %	83,8 %	84,1 %	83,8 %	84,1 %	84,0 %	85,1 %	84,6 %	84,9 %	85,2 %	84,4 %	84,6 %	84,7 %	84,1 %	83,9 %	
Industrieprozesse	CO ₂	59.695	55.878	53.138	53.599	56.238	55.788	57.496	52.248	52.663	51.225	48.844	40.626	45.956	46.052	45.295	44.989	44.927	43.471	45.262	48.622	48.340	
	CH ₄	351	345	371	407	444	449	566	595	579	580	545	513	531	529	513	507	527	529	534	542	512	
	N ₂ O	23.392	24.052	26.517	24.417	27.119	25.210	6.480	8.627	8.459	10.976	9.626	9.974	1.867	1.480	1.201	1.234	1.140	1.173	1.107	1.101	1.093	
	Summe	13.395	12.832	13.300	16.084	16.487	17.080	13.278	14.163	14.100	14.178	14.206	14.658	14.251	14.401	14.586	14.628	14.662	15.069	15.201	15.363	14.846	
Anteil an Gesamtemissionen		96,833 %	93,106 %	93,325 %	94,507 %	100,288 %	98,527 %	77,819 %	75,633 %	75,801 %	76,960 %	73,221 %	65,771 %	62,606 %	62,462 %	61,595 %	61,358 %	61,256 %	60,242 %	62,105 %	65,628 %	64,792 %	
Landwirtschaft	CO ₂	3.184	2.905	2.604	2.368	1.936	2.222	2.803	2.325	2.313	2.427	2.483	2.590	2.325	2.636	2.574	2.699	2.940	3.071	2.773	2.924	2.914	
	CH ₄	43.453	38.432	37.477	37.287	37.607	37.372	35.420	33.161	32.558	32.719	33.117	33.224	33.038	32.806	32.959	33.550	33.844	33.759	33.434	33.147	32.543	
	N ₂ O	32.668	30.209	29.260	28.814	27.285	28.564	30.041	28.703	28.531	27.780	29.471	28.462	28.263	29.824	29.354	29.857	30.689	31.136	30.284	30.000	28.108	
	Summe	79.305	71.545	68.941	68.469	66.829	68.158	68.264	64.189	63.402	62.927	65.071	64.276	63.626	65.265	64.886	66.106	67.472	67.966	66.491	66.070	63.565	
Anteil an Gesamtemissionen		6,2 %	5,9 %	6,0 %	6,0 %	5,9 %	6,1 %	6,5 %	6,3 %	6,2 %	6,5 %	6,7 %	7,1 %	6,8 %	7,1 %	7,0 %	7,0 %	7,3 %	7,5 %	7,3 %	7,4 %		
Abfallwirtschaft	CO ₂	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	CH ₄	36.864	38.344	39.000	38.918	38.137	37.183	27.606	20.303	18.695	17.374	16.153	14.925	13.803	12.992	12.222	11.468	10.846	10.334	9.874	9.348	8.849	
	N ₂ O	1.437	1.334	1.253	1.146	1.129	1.105	922	996	795	806	796	801	794	826	824	818	842	850	862	875	879	
	Summe	38.302	39.678	40.253	40.064	39.267	38.288	28.528	21.299	19.490	18.180	16.949	15.726	14.598	13.818	13.046	12.287	11.688	11.183	10.735	10.222	9.728	
Anteil an Gesamtemissionen		3,1 %	3,3 %	3,5 %	3,3 %	3,5 %	3,4 %	2,7 %	2,1 %	2,1 %	1,9 %	1,7 %	1,5 %	1,5 %	1,4 %	1,3 %	1,3 %	1,2 %	1,2 %	1,1 %	1,1 %		
Gesamtemissionen		1.251.488	1.204.358	1.154.107	1.145.111	1.125.980	1.123.001	1.044.664	993.199	1.000.520	973.839	976.672	909.260	942.338	919.435	924.147	941.870	902.389	906.320	909.032	894.256	858.369	
Anteil der Stoffe an den Gesamtemissionen		CO ₂	84,09 %	84,21 %	83,69 %	83,51 %	83,45 %	83,63 %	86,13 %	87,23 %	87,75 %	87,43 %	86,97 %	88,26 %	88,04 %	88,07 %	88,29 %	87,84 %	87,81 %	88,06 %	87,96 %	88,00 %	
		CH ₄	9,66 %	9,55 %	9,66 %	9,78 %	9,60 %	9,40 %	8,48 %	8,98 %	8,56 %	8,53 %	8,41 %	8,62 %	8,28 %	8,32 %	8,36 %	8,32 %	8,26 %	8,11 %	8,12 %	8,13 %	
		N ₂ O	5,16 %	5,17 %	5,50 %	5,31 %	5,49 %	5,44 %	4,12 %	4,36 %	4,28 %	4,59 %	4,62 %	4,86 %	4,85 %	4,08 %	3,99 %	3,97 %	4,22 %	4,27 %	4,16 %	4,14 %	
		F-Gase	1,07 %	1,07 %	1,15 %	1,40 %	1,46 %	1,52 %	1,27 %	1,43 %	1,41 %	1,46 %	1,45 %	1,61 %	1,51 %	1,57 %	1,58 %	1,62 %	1,66 %	1,67 %	1,72 %	1,73 %	

* Emissionen ohne Berücksichtigung von Landnutzung, Landnutzungsänderungen und Forstwirtschaft

Quelle: Umweltbundesamt, Nationale Treibhausgasinventuren für die deutsche Bundesregierung einschließlich Emissionen von 1990, Emissionsentwicklung 1990 bis 2018 (Stand 12/2019)

Abbildung 12: Emissionen ausgewählter Treibhausgase in Deutschland nach Kategorien in Tsd. t Kohlendioxid-Äquivalenten (Quelle: https://www.umweltbundesamt.de/sites/default/files/medien/384/bilder/8_tab_thg-emi-kat_2020.png, Zugriff am: 12.04.2021)

Anlage 3

Source Name	Block Type Name/ Type Name	Component	Reference	input bus	output bus	Efficiency (el)	Efficiency (th)	Latitude	Longitude
<i>Unique Identifier</i>			<i>ref</i>	<i>inp</i>	<i>outp</i>	<i>[ηel]</i>	<i>[ηth]</i>	<i>[°]</i>	<i>[°]</i>
Gas supply	-	source	gas supply	-	gas	-	-	-	-
Coal supply	-	source	coal supply	-	coal	-	-	-	-
Oil supply	-	source	oil supply	-	oil	-	-	-	-
Waste	-	source	waste	-	waste	-	-	-	-
Heizkraftwerk ADM HKW	ocgt pp	Mimo-transformer	chp1	gas	electricity, hot water	0,3773	0,3	53,51	9,94985
HKW Moorburg Block A Moorburg	hard coal pp	Mimo-transformer	pp1	coal	electricity	0,4625	-	53,489	9,949
HKW Moorburg Block B Moorburg	hard coal pp	Mimo-transformer	pp2	coal	electricity	0,4625	-	53,489	9,949
HKW Tiefstack GuD Tiefstack	ccgt pp	Mimo-transformer	chp2	gas	electricity, hot water	0,585	0,4	53,53	10,07
HKW Tiefstack Block 2 Tiefstack	hard coal pp	Mimo-transformer	chp3	coal	electricity, hot water	0,4075	0,4	53,53	10,06
Wedel GT A	mineral oil pp	Mimo-transformer	pp3	oil	electricity	0,3072	-	53,5662	9,72864
Wedel GT B	mineral oil pp	Mimo-transformer	pp4	oil	electricity	0,3072	-	53,5662	9,72864
HKW Wedel Block 1 Wedel	hard coal pp	Mimo-transformer	chp4	coal	electricity, hot water	0,4075	0,4	53,5667	9,72864
HKW Wedel Block 2 Wedel	hard coal pp	Mimo-transformer	chp5	coal	electricity, hot water	0,4075	0,4	53,5667	9,72864
MVR Müllverwertung Rugenberger Damm	waste pp	Mimo-transformer	chp6	waste	electricity, hot water	0,33	0,4		
Heizwerk Hafencity	ccgt hp	Mimo-transformer	hp1	gas	hot water	-	0,966666667	53,54106052	9,995900959
MVR Borsigstraße, Stelling Moor, Stapelfeld (Biomass)	-	source	Bm1	-	electricity, hot water	-	-	-	-
Combustion Plants sum (MASTR)	-	Mimo-transformer	cs1	-	-	-	-	-	-
Photovoltaik	Csi-Panel	source	Pv1	-	electricity	-	-	-	-
Wind (Onshore)	name='Vestas_V112_3MW'	source	Won1	-	electricity	-	-	-	-
Storage	electrical energy storage	storage	Su1	electricity	electricity	-	-	-	-
Imported electricity	-	source	imported el	-	electricity	-	-	-	-
Imported heat	-	source	imported heat	-	hot water	-	-	-	-
Electrical demand	-	sink	demand el	electricity	-	-	-	-	-
Heat demand	-	sink	demand th	hot water	-	-	-	-	-
Excess Electricity	-	sink	excess el	electricity	-	-	-	-	-
Excess Heat	-	sink	excess th	hot water	-	-	-	-	-
Gas bus	-	bus	gas pipeline	gas	gas	-	-	-	-
Coal bus	-	bus	coal supply line	coal	coal	-	-	-	-
Mineral oil bus	-	bus	oil supply line	oil	oil	-	-	-	-
waste bus	-	bus	waste supply	waste	waste	-	-	-	-
Electrical bus	-	bus	Powerline	electricity	electricity	-	-	-	-
Thermal bus	-	bus	district heating pipeline	hot water	hot water	-	-	-	-

Source Name	Region	Component2	Sector	carrier	Installed Capacity (el)	Minimum Capacity	Minimum Capacity Ratio	Installed Capacity (th)
Unique Identifier	Arbitrary string	string	P/H/M/C	string	[MW _{el}]	[MW _{el}]	[MW _{el} /MW _{el}]	[MW _{th}]
Gas supply	HH	source		gas	-	-	-	-
Coal supply	HH	source		coal	-	-	-	-
Oil supply	HH	source		oil	-	-	-	-
Waste	HH	source		waste	-	-	-	-
Heizkraftwerk ADM HKW	HH	transformer	Coupled	gas	22,5	6,75	0,300	35
HKW Moorburg Block A Moorburg	HH	transformer	power	coal	784	296	0,378	-
HKW Moorburg Block B Moorburg	HH	transformer	power	coal	768	296	0,385	-
HKW Tiefstack GuD Tiefstack	HH	transformer	Coupled	gas	123	51	0,415	180 (779)
HKW Tiefstack Block 2 Tiefstack	HH	transformer	Coupled	coal	188	68	0,362	293 (779)
Wedel GT A	SH	transformer	Power	oil	50,5	20,2	0,400	-
Wedel GT B	SH	transformer	Power	oil	50,5	20,2	0,400	-
HKW Wedel Block 1 Wedel	SH	transformer	Coupled	coal	130	57	0,438	130 (423)
HKW Wedel Block 2 Wedel	SH	transformer	Coupled	coal	118	44	0,373	88 (423)
MVR Müllverwertung Rugenberger Damm	HH	transformer	Coupled	waste	24	9,6	0,400	67 (70)
Heizwerk Hafency	HH	transformer	heat	gas	-	-	-	348
MVR Borsigstraße, Stelling Moor, Stapelfeld (Biomass)	HH	source	Power; Heat	biomass	48,4	18	0,372	126
Combustion Plants sum (MASTR)	HH	-	Power; Coupled	-	3181.5 (2278.5)	868,75	3181.5 (2278.5)	793 (1272)
Photovoltaik	HH	source	Power	solar	24,17	0	0	-
Wind (Onshore)	HH	source	Power	wind	122,44	0	0	-
Storage	HH	storage	Power	electricity	3,2333	3,2333	3,2333	-
Imported electricity	-	source	Power	electricity	-	-	-	-
Imported heat	-	source	Power	electricity	-	-	-	-
Electrical demand	HH	sink	Power	electricity	-	-	-	-
Heat demand	HH	sink	Heat	hot water	-	-	-	-
Excess Electricity	-	sink	Power	electricity	-	-	-	-
Excess Heat	-	sink	Heat	hot water	-	-	-	-
Gas bus	HH	bus		gas	-	-	-	-
Coal bus	HH	bus		coal	-	-	-	-
Mineral oil bus	HH	bus		oil	-	-	-	-
waste bus	HH	bus		waste	-	-	-	-
Electrical bus	HH	bus		electricity	-	-	-	-
Thermal bus	HH	bus		hot water	-	-	-	-

Source Name	Flow costs/ variable costs (el)	Flow costs/ variable costs (th)	Flow emissions	timeseries	expandable	expansion costs	expansion limits	Minimum Downtime	Time to start (cold start)
Unique Identifier	[€/MW]	[€/MW]	[tCO2/MW]	(yes=1, no=0)	(yes=1, no=0)	[annuity (€/MW)]	[MW]	[h]	[h]
Gas supply	-	-	-	0	0	0	-	0	-
Coal supply	-	-	-	0	0	0	-	0	-
Oil supply	-	-	-	0	0	0	-	0	-
Waste	-	-	-	0	0	0	-	0	-
Heizkraftwerk ADM HKW	90	21,6	0,756	0	0	0	-	0,16667	1
HKW Moorburg Block A Moorburg	82		0,3361	0	0	0	-	1	6
HKW Moorburg Block B Moorburg	82		0,3361	0	0	0	-	1	6
HKW Tiefstack GuD Tiefstack	90	18,9	0,756	0	0	0	-	0,833	4
HKW Tiefstack Block 2 Tiefstack	82	19,68	0,756	0	0	0	-	1	6
Wedel GT A	90	-	0,756	0	0	0	-	3	6
Wedel GT B	90	-	0,756	0	0	0	-	3	6
HKW Wedel Block 1 Wedel	82	19,68	0,756	0	0	0	-	1	6
HKW Wedel Block 2 Wedel	82	19,68	0,756	0	0	0	-	1	6
MVR Müllverwertung Rugenberger Damm	82	20	0,15336	0	0	0	-	2	7
Heizwerk Hafencity	-	20		0	0	0	-	0	-
MVR Borsigstraße, Stelling Moor, Stapelfeld (Biomass)	61	-	0,001	0	0	0	-	1	3
Combustion Plants sum (MASTR)	-	-	0,611756	-	-	-	-	-	-
Photovoltaik	74	-	0	1	1	52500	inf	0	0
Wind (Onshore)	61	-	0,007	1	1	91875	inf	0	0
Storage	-	-		0	1	0	inf	0	-
Imported electricity	-	-	0,401	1	0	0	inf	0	-
Imported heat	-	-	-	0	0	0	inf	0	-
Electrical demand	-	-	-	1	0	0	inf	0	-
Heat demand	-	-	-	1	0	0	inf	0	-
Excess Electricity	-	-	-	0	1		inf	0	-
Excess Heat	-	-	-	0	1		inf	0	-
Gas bus	-	-	-	0	0	0	-	0	-
Coal bus	-	-	-	0	0	0	-	0	-
Mineral oil bus	-	-	-	0	0	0	-	0	-
waste bus	-	-	-	0	0	0	-	0	-
Electrical bus	-	-	-	0	0	0	-	0	-
Thermal bus	-	-	-	0	0	0	-	0	-

Source Name	Cold Start costs
<i>Unique Identifier</i>	[€]
Gas supply	-
Coal supply	-
Oil supply	-
Waste	-
Heizkraftwerk ADM HKW	24
HKW Moorburg Block A Moorburg	49
HKW Moorburg Block B Moorburg	49
HKW Tiefstack GuD Tiefstack	40
HKW Tiefstack Block 2 Tiefstack	49
Wedel GT A	45
Wedel GT B	45
HKW Wedel Block 1 Wedel	49
HKW Wedel Block 2 Wedel	49
MVR Müllverwertung Rugenberger Damm	40
Heizwerk Hafencity	-
MVR Borsigstraße, Stellinginger Moor, Stapelfeld (Biomass)	57
Combustion Plants sum (MASTR)	-
Photovoltaik	0
Wind (Onshore)	0
Storage	-
Imported electricity	-
Imported heat	-
Electrical demand	-
Heat demand	-
Excess Electricity	-
Excess Heat	-
Gas bus	-
Coal bus	-
Mineral oil bus	-
waste bus	-
Electrical bus	-
Thermal bus	-

Anlage 4

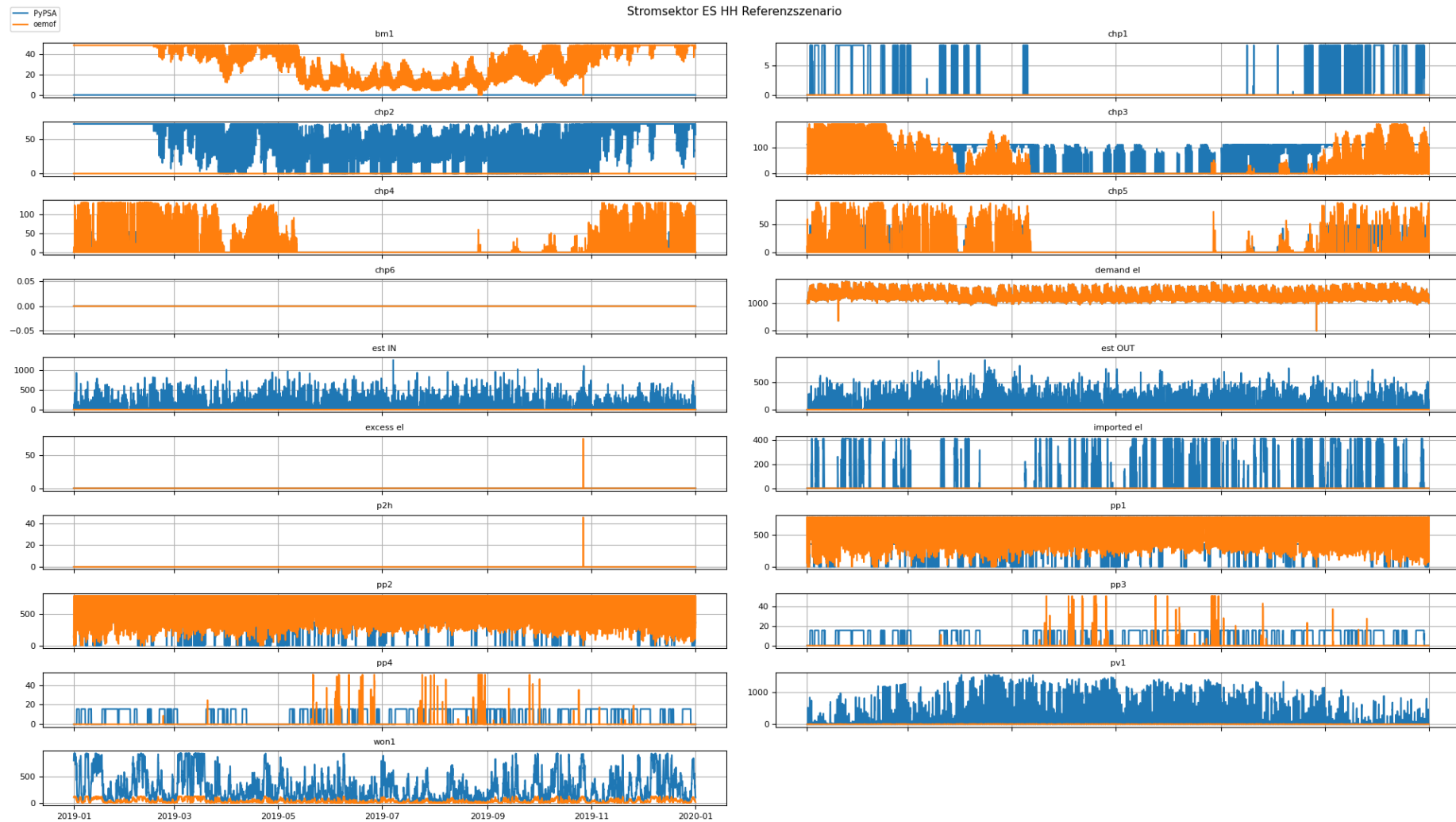


Abbildung 13: Lastgang Stromsektor ES HH Referenzszenario aller Komponenten

Anlage 5

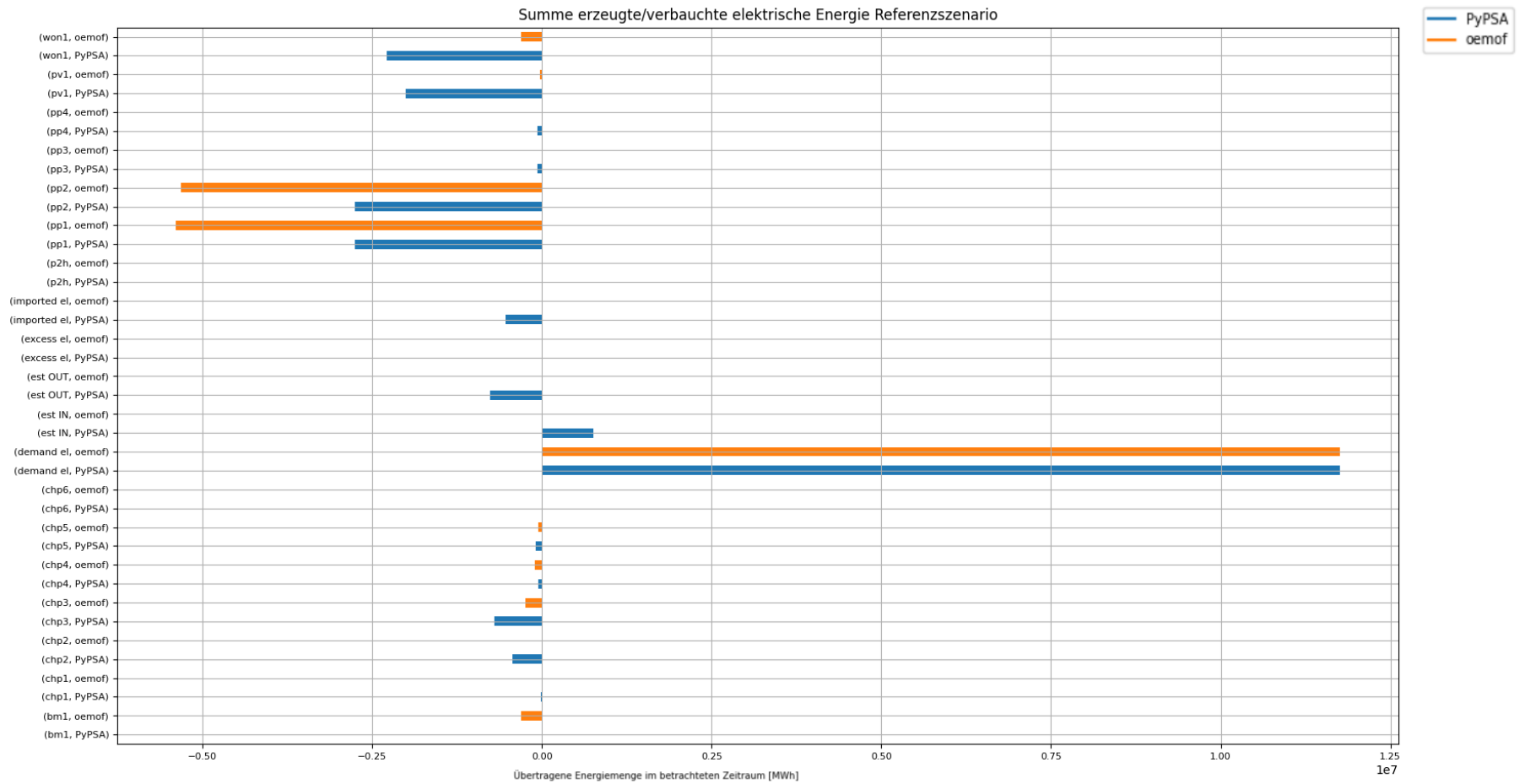


Abbildung 14: Summe erzeugter bzw. verbrauchter elektrischer Energie Referenzszenario aller Komponenten

Anlage 6

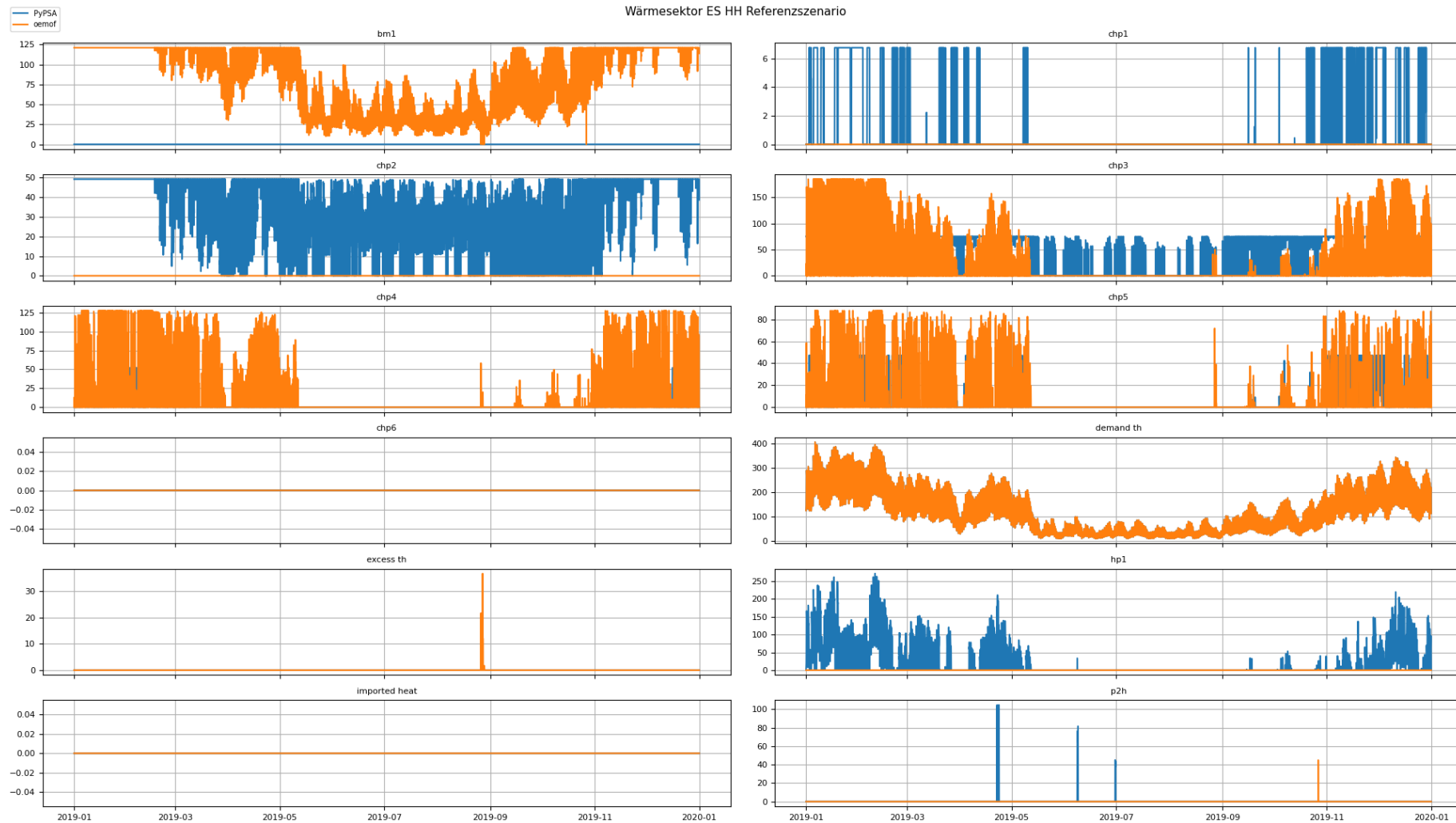


Abbildung 15: Lastgang Wärmesektor ES HH Referenzszenario aller Komponenten

Anlage 7

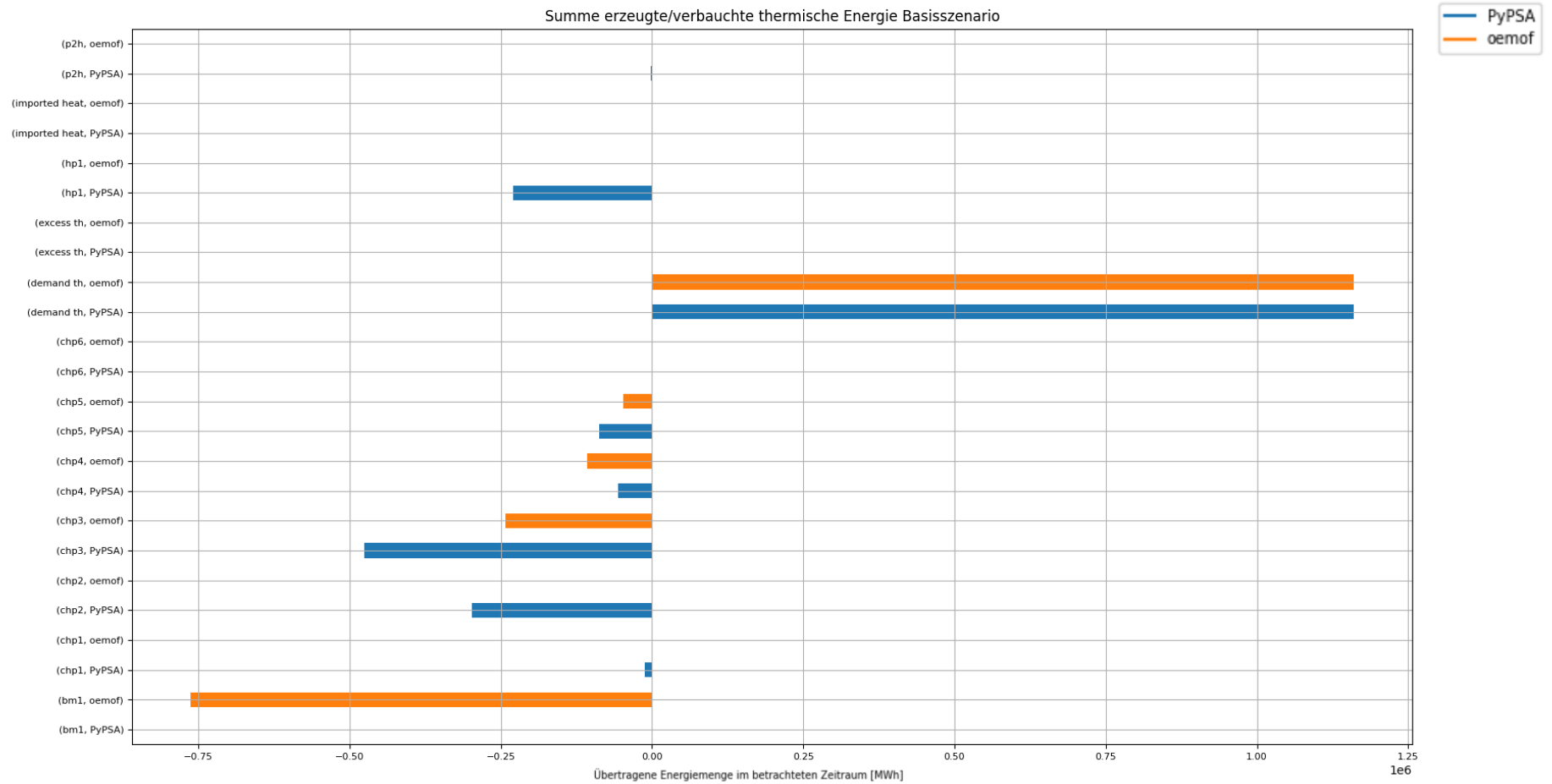


Abbildung 16: Summe erzeugter bzw. verbrauchter thermischer Energie Referenzszenario aller Komponenten

Anlage 8

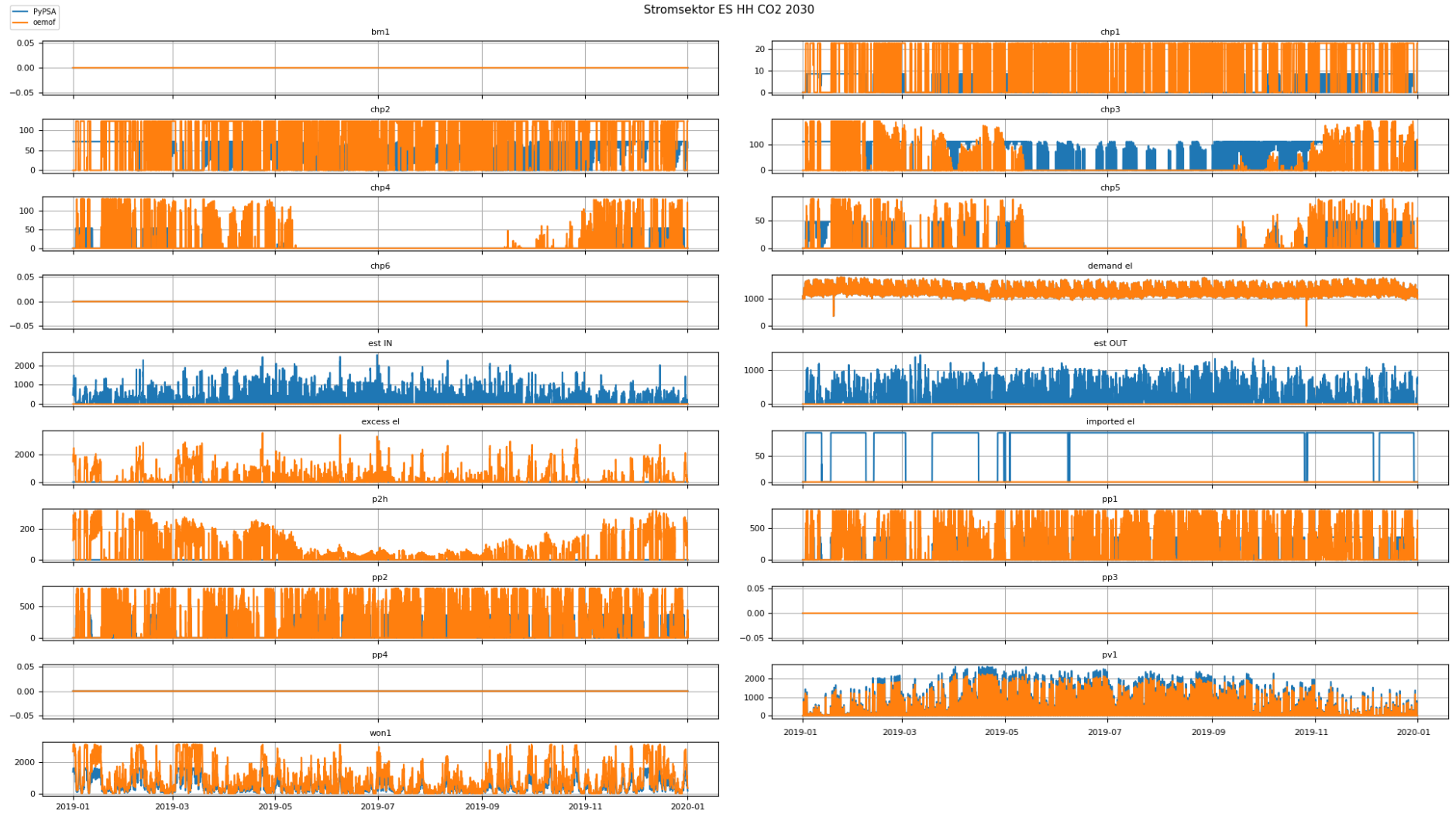


Abbildung 17: Lastgang Stromsektor ES HH Szenario "CO2 2030" aller Komponenten

X

Anlage 9

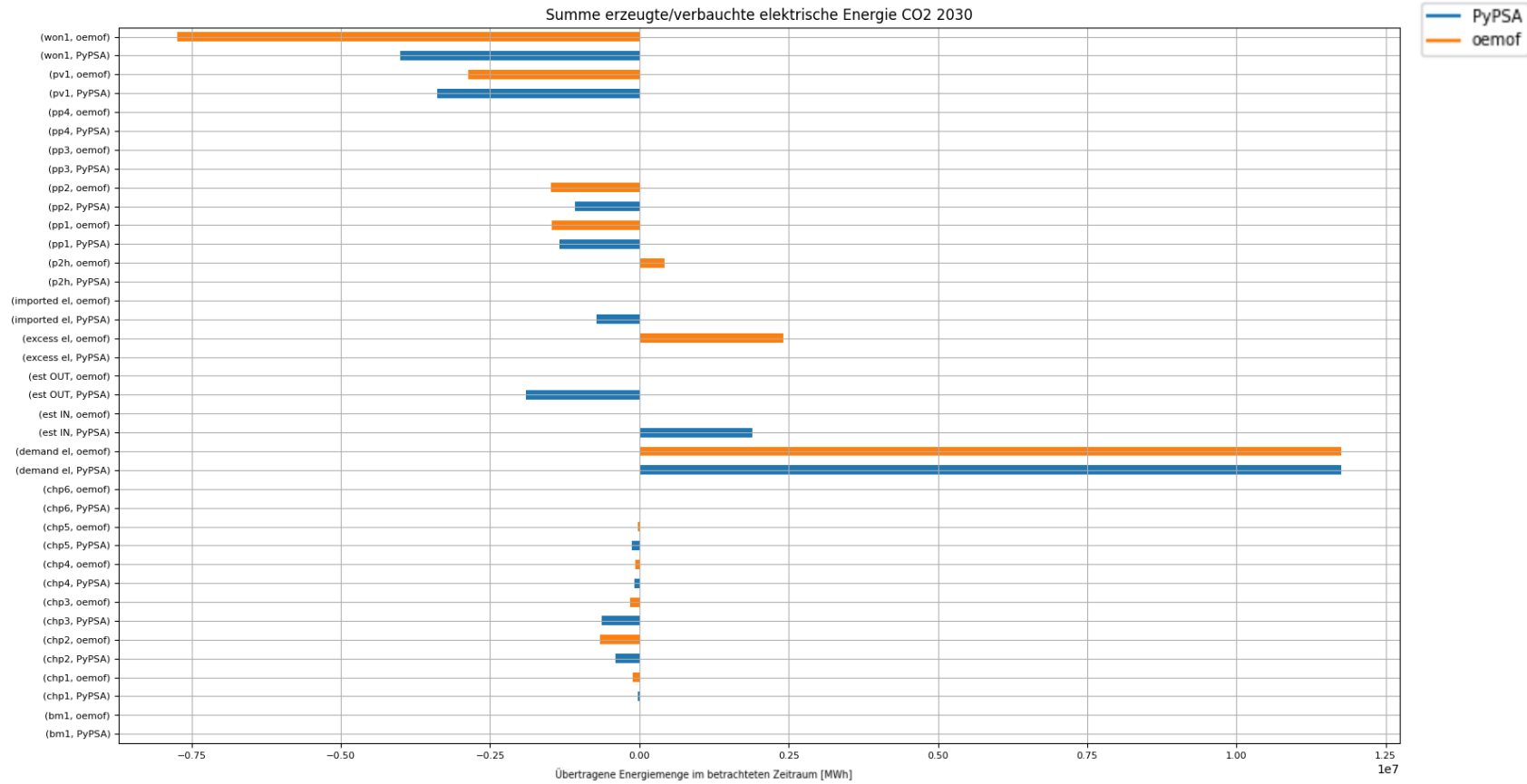


Abbildung 18: Summe erzeugter bzw. verbrauchter elektrischer Energie Szenario "CO2 2030" aller Komponenten

Anlage 10

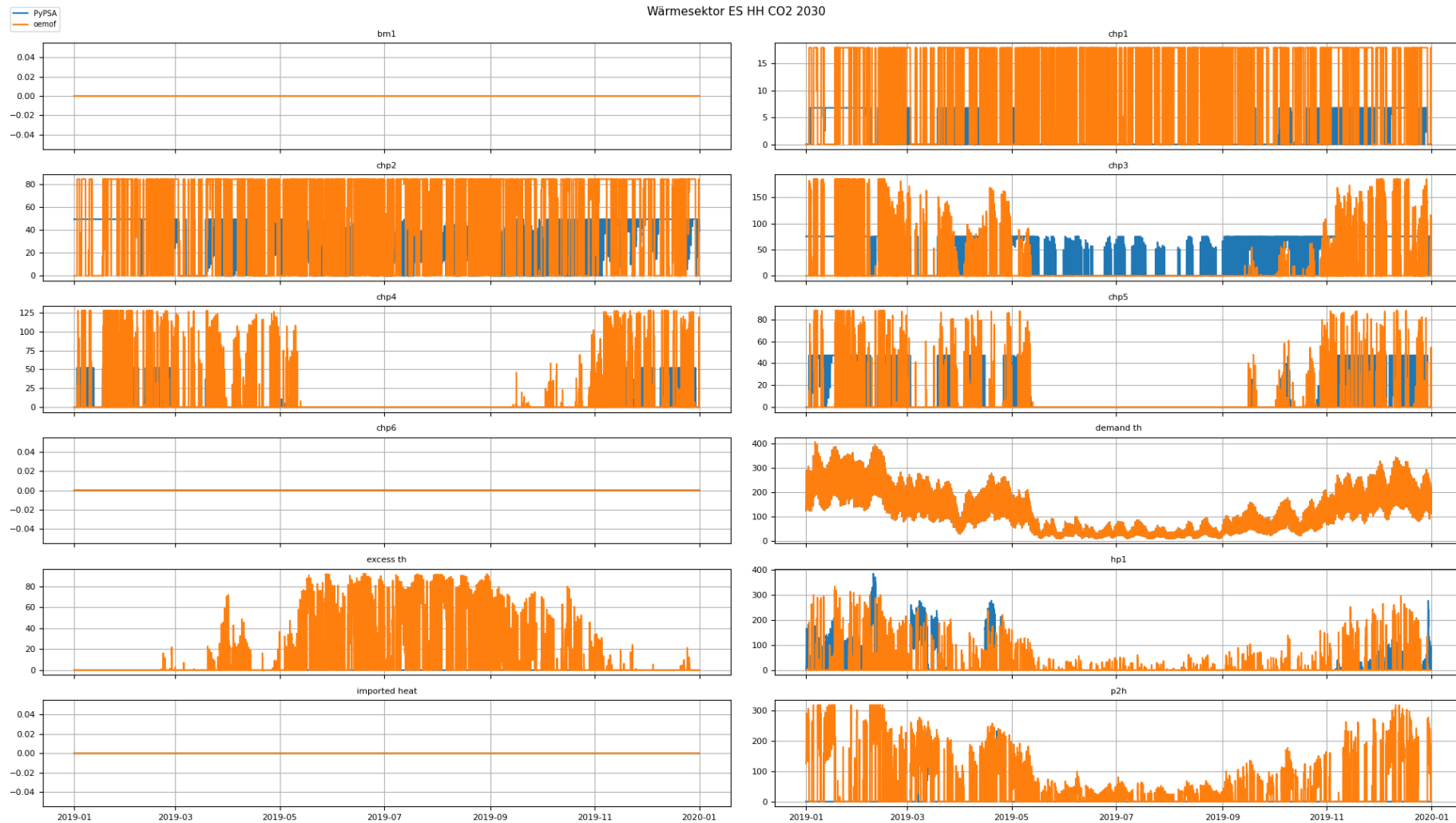


Abbildung 19: Lastgang Wärmesektor ES HH Szenario "CO2 2030" aller Komponenten

Anlage 11

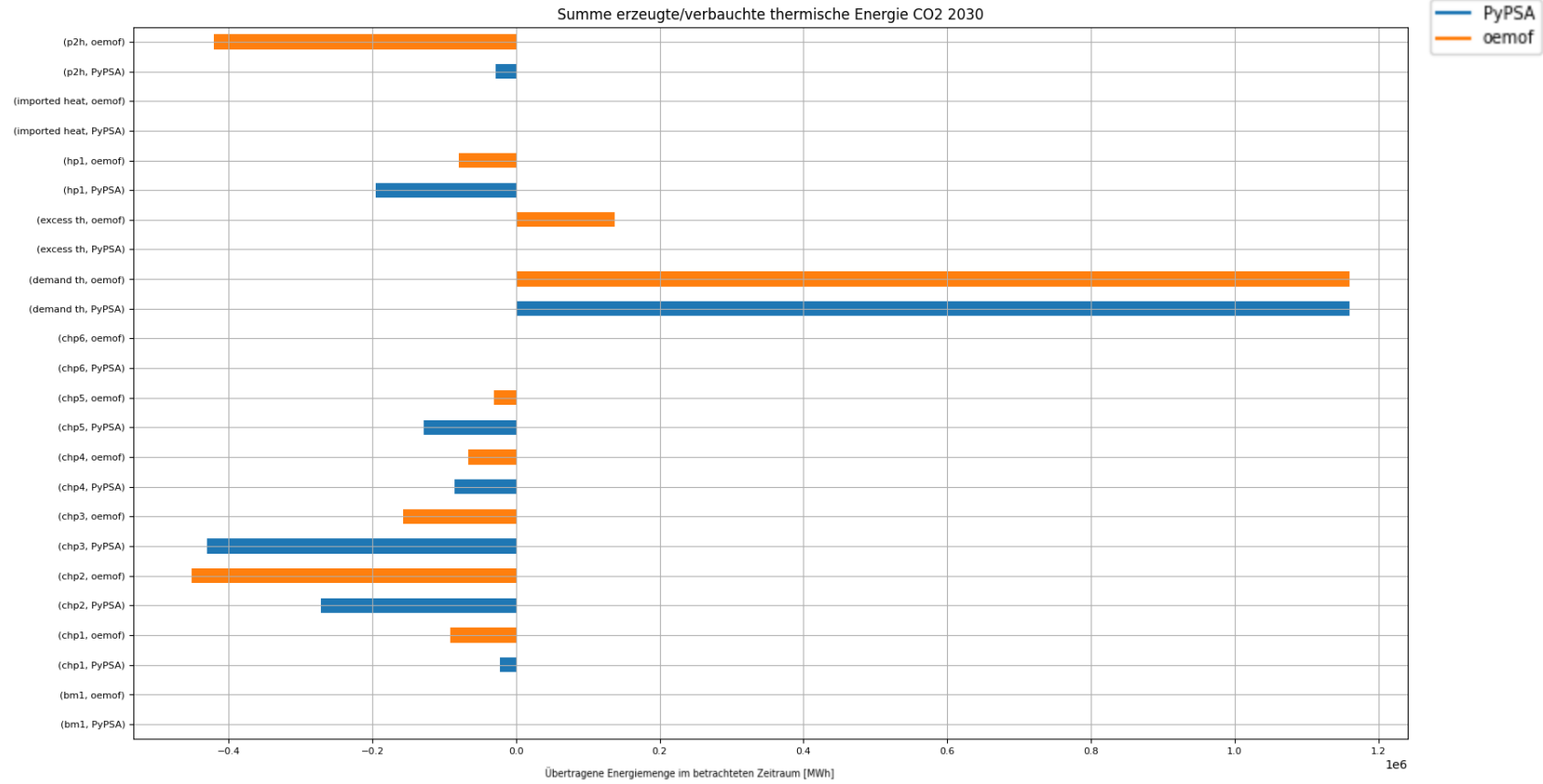


Abbildung 20: Summe erzeugter bzw. verbrauchter thermischer Energie Szenario "CO2 2030" aller Komponenten

Anlage 12

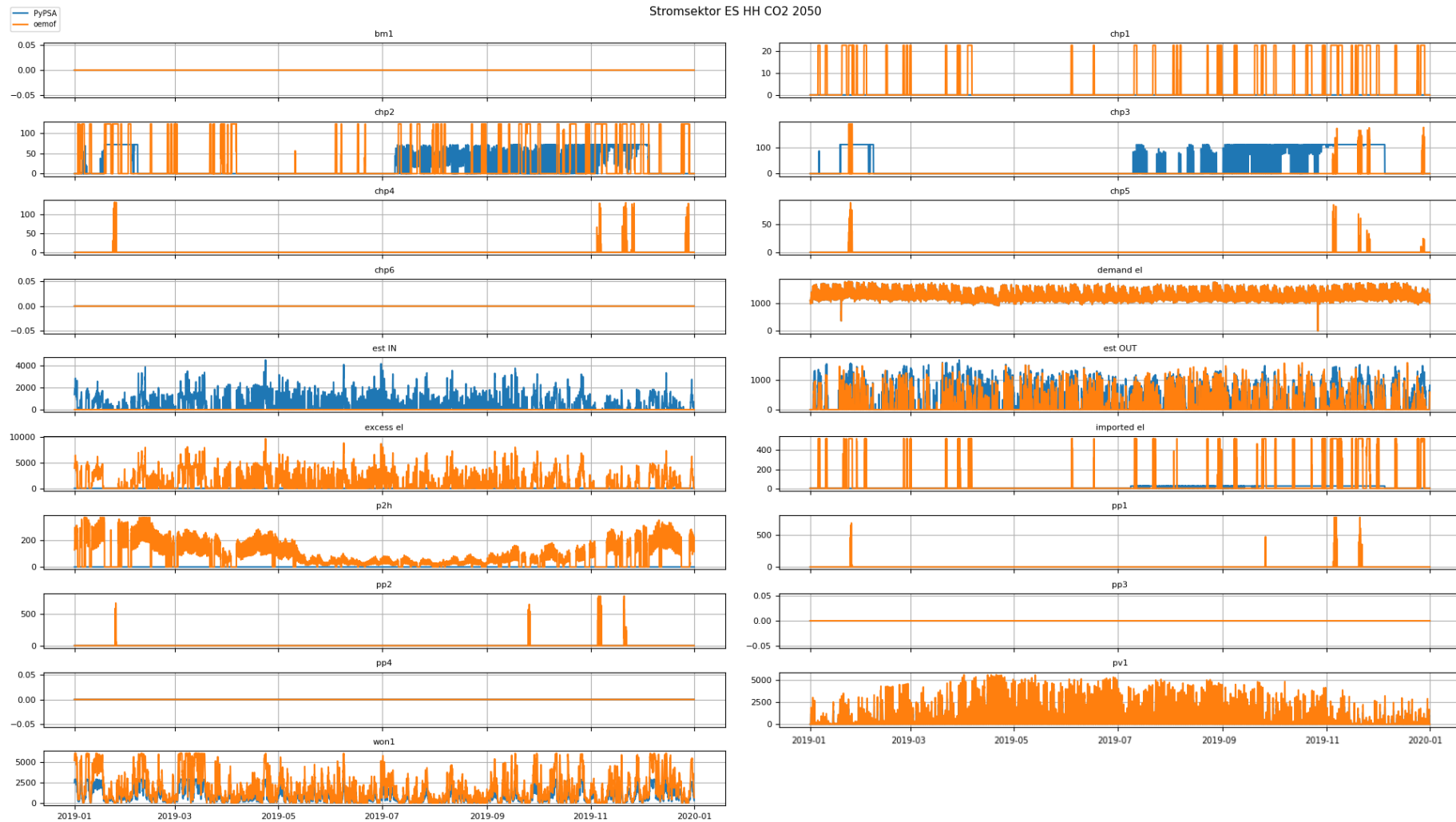


Abbildung 21: Lastgang Stromsektor ES HH Szenario "CO2 2050" aller Komponenten

Anlage 13

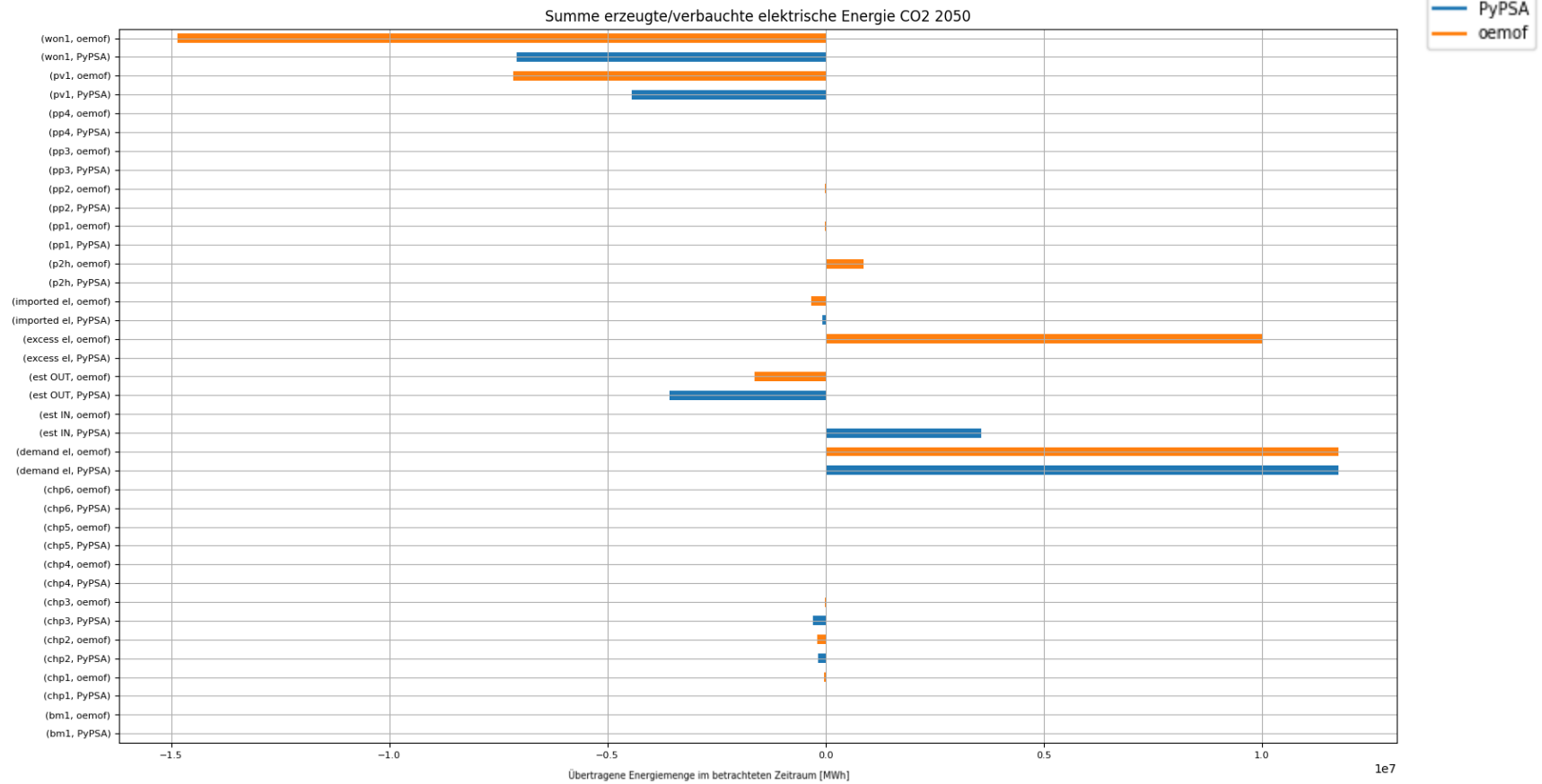


Abbildung 22: Summe erzeugter bzw. verbrauchter elektrischer Energie Szenario "CO2 2050" aller Komponenten

Anlage 14

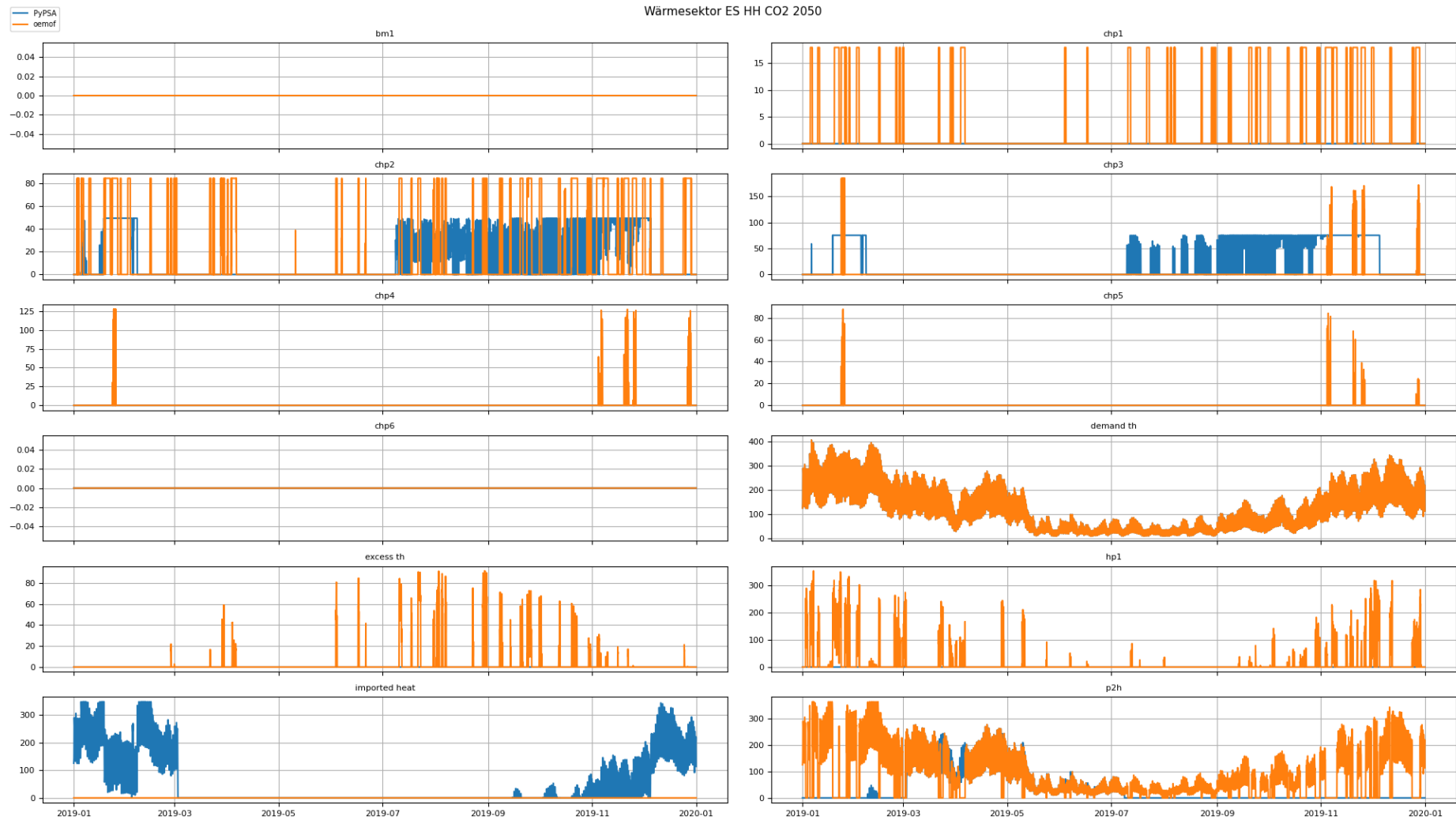


Abbildung 23: Lastgang Wärmesektor ES HH Szenario "CO2 2050" aller Komponenten

Anlage 15

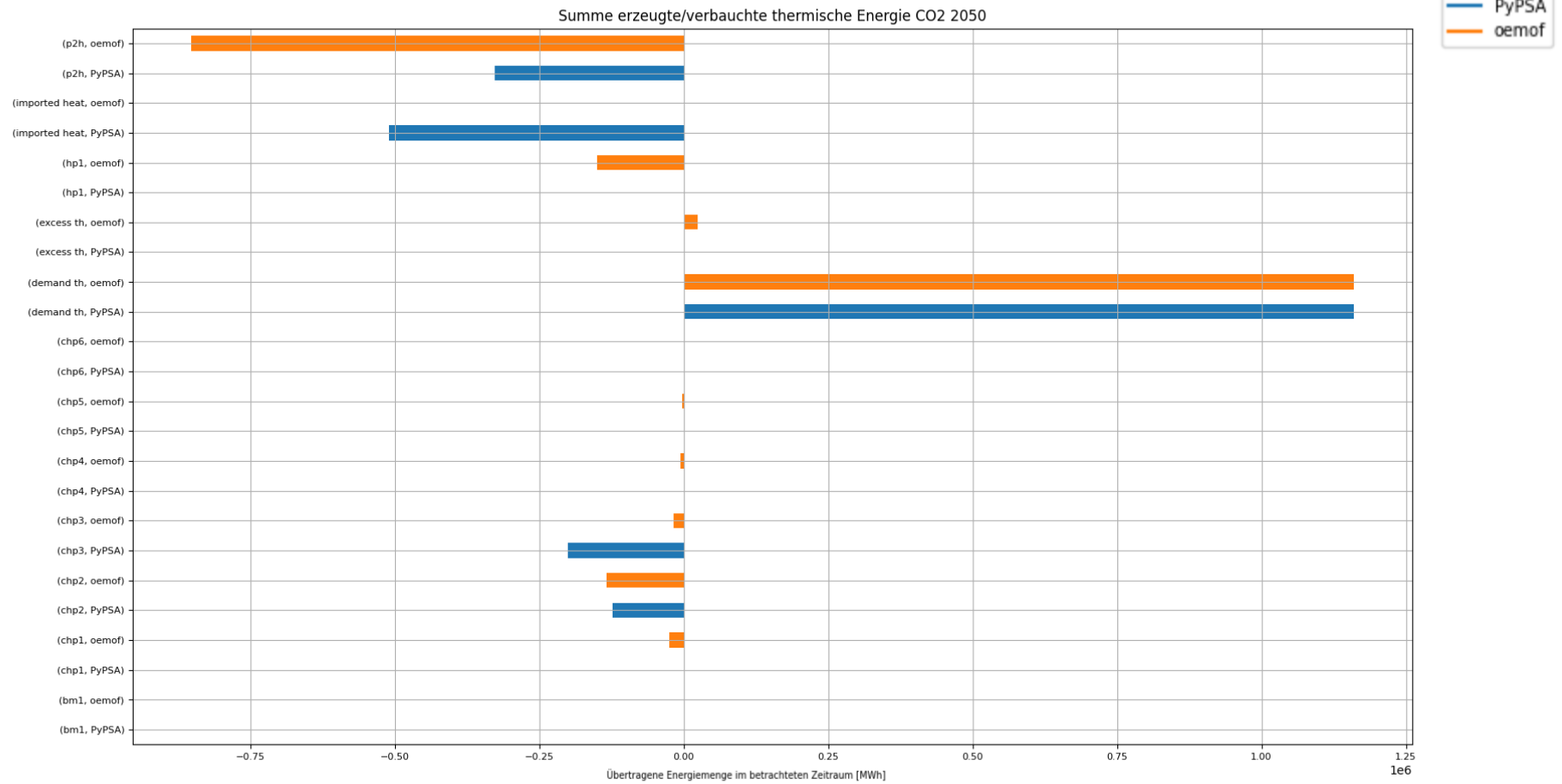


Abbildung 24: Summe erzeugter bzw. verbrauchter thermischer Energie Szenario "CO2 2050" aller Komponente