Forschungsbericht 2023-15

UNSEEN
Evaluation of Uncertainties in Linear-Optimizing Energy System Models Compendium

Projektleitung:
Benjamin Fuchs und Karl-Kiên Cao

Deutsches Zentrum für Luft- und Raumfahrt
Institut für Vernetzte Energiesysteme
Stuttgart

DLR Deutsches Zentrum
für Luft- und Raumfahrt
German Aerospace Center

Forschungsbericht 2023-15

UNSEEN
Bewertung der Unsicherheiten in linear optimierenden Energiesystemmodellen
Kompendium

Projektleitung:
Benjamin Fuchs und Karl-Kiên Cao

Deutsches Zentrum für Luft- und Raumfahrt
Institut für Vernetzte Energiesysteme
Oldenburg/Stuttgart

98 Seiten
32 Bilder
17 Tabellen
80 Literaturstellen

DLR Deutsches Zentrum
für Luft- und Raumfahrt

Herausgeber:

Deutsches Zentrum
für Luft und Raumfahrt e. V.
Wissenschaftliche Information
Linder Höhe
D-51147 Köln

ISSN 1434-8454
ISRN DLR-FB-2023-15
Erscheinungsjahr 2023

DOI: 10.57676/w2rq-bj85

Energiesystemanalyse, Szenarien, Optimierung, High-Performance-Computing, PIPS-IPM++, REMix, AMIRIS, Modellkopplung

Projektleitung: Benjamin Fuchs und Karl-Kiên Cao
DLR, Institut für Vernetzte Energiesysteme, Stuttgart

UNSEEN
Bewertung der Unsicherheiten in linear optimierenden Energiesystemmodellen
Kompendium

Für die Energiesystemforschung sind Softwarie-Modelle ein Kernelement zur Analyse von Szenarien. Das Forschungsprojekt UNSEEN hatte das Ziel eine bisher unerreichte Anzahl an modellbasierten Energieszenarien zu berechnen, um Unsicherheiten – vor allem unter Nutzung linear optimierender Energiesystem-Modelle - besser bewerten zu können. Hierfür wurden umfangreiche Parametervariationen auf Energieszenarien angewendet und das wesentliche methodische Hindernis in diesem Zusammenhang adressiert: die rechnerische Beherrschbarkeit der zu lösenden mathematischen Optimierungsprobleme. Im Vorläuferprojekt BEAM-ME wurde mit der Entwicklung und Anwendung des Open-Source-Lösers PIPS-IPM++ die Grundlage für den Einsatz von High-Performance-Computing (HPC) zur Lösung dieser Modelle gelegt. In UNSEEN war dieser Löser die zentrale Komponente eines Workflows, welcher zur Generierung, Lösung und multi-kriteriellen Bewertung von Energieszenarien auf dem Hochleistungscomputer JUWELS am Forschungszentrum Jülich implementiert wurde. Zur effizienten Generierung und Kommunikation von Modellinstanzen für Methoden der mathematischen Optimierung auf HPC wurde eine weitere Workflow-Komponente von der GAMS Software GmbH entwickelt: der Szenariogenerator. Bei der Weiterentwicklung von Lösungsalgorithmen für linear optimierende Energie-Systemmodelle standen gemischt-ganzzahlige Optimierungsprobleme im Fokus, welche für die Modellierung konkreter Infrastrukturen und Maßnahmen zur Umsetzung der Energiewende gelöst werden müssen. Die in diesem Zusammenhang stehenden Arbeiten zur Entwicklung von Algorithmen wurden von der Technischen Universität Berlin verantwortet. Bei Design und Implementierung dieser Methoden wurde sie vom Zuse Instituts Berlin unterstützt.

energy system analyis, scenarios, optimization, high-perfomance-computing, PIPS-IPM++, REMix, AMIRIS, model coupling

Project coordination: Benjamin Fuchs and Karl-Kiên Cao
DLR, Institute of Networked Energy Systems, Stuttgart

UNSEEN
Evaluation of Uncertainties in Linear-Optimizing Energy System Models
Compendium

In energy system research, software models are a key element for the analysis of scenarios. The aim of the UNSEEN research project was to compute an unprecedented number of model-based energy scenarios in order to better evaluate uncertainties - especially using linear energy system optimization models. Therefore, extensive parameter variations have been applied to energy scenarios and the most important methodological obstacle in this context has been addressed: the computational manageability of the mathematical optimization problems to be solved. In the precursor project BEAM-ME, the development and application of the open-source solver PIPS-IPM++ laid the foundation for the use of high-performance computing (HPC) to solve these problems. In UNSEEN, this solver was the central component of a workflow that was implemented for the generation, solution, and multi-criteria assessment of energy scenarios on the high-performance computer JUWELS at Forschungszentrum Jülich. For efficient generation and communication of model instances for mathematical optimization methods on HPC, another workflow component was developed by GAMS Software GmbH: the scenario generator. The further development of solution algorithms for linear optimizing energy system models focused on mixed-integer optimization problems, which are to be solved for the modeling of concrete infrastructures and measures for the implementation of the energy transition. The Technical University of Berlin was responsible for the development of algorithms in this context. The Zuse Institute Berlin supported the design and implementation of these methods.

Gefördert durch:

Bundesministerium
für Wirtschaft
und Klimaschutz

aufgrund eines Beschlusses
des Deutschen Bundestages

UNSEEN
Evaluation of Uncertainties in Linear-Optimizing Energy System Models

Compendium

Lovis Anderson[2], Aileen Böhme[4], Thomas Breuer[5], Jan Buschmann[1], Karl-Kien Cao[1], Frederik Fiand[4], Ulrich Frey[1], Benjamin Fuchs[1], Nils-Christian Kempke[3], Kai von Krbek[1], Wided Medjroubi[1], Judith Riehm[1], Shima Sasanpour[1], Sonja Simon[1], Charlie Vanaret[2], Manuel Wetzel[1], Menghzu Xiao[1], and Janina Zittel[2]

[1]Deutsches Zentrum für Luft- und Raumfahrt
[2]Zuse Institute Berlin
[3]Technische Universität Berlin
[4]GAMS Software GmbH
[5]Forschungszentrum Jülich GmbH (Jülich Research Centre)

September 6, 2023

Acknowledgement

The UNSEEN project was funded by the German Federal Ministry for Economic Affairs and Energy under grant number FKZ 03EI1004A. The authors gratefully acknowledge the Gauss Centre for Supercomputing e. V. (www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS (Jülich Supercomputing Centre, 2021) at Jülich Supercomputing Centre (JSC).

Contents

Acronyms and Abbreviations

ESOM	Energy System Optimization Model
RES	Renewable Energy Sources
RE	Renewable Energy
IAM	Integrated Assessment Model
VRES	Variable Renewable Energy Sources
LP	Linear Programming/Linear Program
MIP	Mixed-Integer Programming/Mixed-Integer Linear Program
ASM	Active Set Method
IPM	Interior-Point Method
NN	Neural Network
MPI	Message Passing Interface
OpenMP	Open Multi-Processing
HPC	High Performance Computing
POI	Points of interest
GDX	Gams Data eXchange
cdf	cumulative distribution function
DE	Deterministic Equivalent
EMP	Extended Mathematical Programming
MGA	Modeling to Generate Alternatives
UC	Unit Commitment
FPR	Fix-propagate-repair
MPI	Message Passing Interface

Chapter 1

Introduction

1.1 Motivation

In the energy sector, modeling tools have been widely used to address the challenge of reducing greenhouse gas emissions. These tools are employed to analyze and optimize future energy system designs that enable an affordable, secure, and sustainable low-carbon energy supply. One crucial aspect in such optimization models is to manage the computational burden. Therefore, modelers have adopted strategies to reduce the temporal or spatial scale by analyzing representative snapshots or using clustered network models. In the quest for faster model execution, we refer to such approaches as model-based strategies, particularly heuristic methods. However, their potential for speeding-up computation time is limited as they are typically implemented on shared memory computer architectures. To address this limitation, solver-based speed-up approaches, exemplified by the newly developed PIPS-IPM++ solver, have emerged. PIPS-IPM++ is specifically designed for parallel solving a ESOM on distributed memory computers, including High-Performance Computers. Leveraging the block structure of a linear program's coefficient matrix, this solver offers significant speed-up potential for Linear Programming/Linear Program (LP). However, due to the requirement of providing domain-specific information to the solver, implementing these approaches may entail higher costs.

Given the great potentials of solver-based approaches like the application of PIPS-IPM++ for achieving significant speed-up in model execution, several new research directions arise. On the one hand, it is clear that there is a trade-off between the potential benefits and the associated implementation costs which has to be further investigated. On the other hand, the capability to solve large-scale ESOMs with good performance allows the consideration of uncertainties in energy scenario analysis by more sophisticated techniques than comparing only a hand full of worst-, best- and business-as-usual cases: ensemble calculations. Moreover, in combination with suitable heuristics for integer fixing, a performant solver for LP represents the basis for enhancning the performance for solving a mathematical problem class which is even more relevant beyond the scope of energy systems: Mixed-Integer Programming/Mixed-Integer Linear Program (MIP). This is the point where the research project UNSEEN started.

1.2 The Research Project UNSEEN

The UNSEEN research project tackled the crucial issue of reliability in the research domain of energy systems analysis. The project aimed to overcome the limitations of previous modeling efforts by exploring an extensive parameter space (see 4.2) that encompassed a wide range of scenarios. A significant focus of the project was to develop a comprehensive understanding of the interplay between different parameters within the energy system by leveraging the capabilities of high-performance computers and parallel computations (see chapter 2). To achieve this, a massive number of scenarios were automatically generated and evaluated using a newly developed parameter sampling technique (see chapter 3) and state-of-the-art modeling tools. Through this process, UNSEEN aimed to identify robust scenarios that would enable a sustainable and resilient energy future.

In this context, a key aspect was to address the limitations of traditional least-cost sceanrio assessments. We incorporated a broad range of factors represented by inidicators from the triangle of energy supply objectves: affordability, reliability and sustainability (see section 4.4.1). Furthermore, the project emphasized the integration of modeling tools, such as the REMix optimization model and the AMIRIS agent-based simulation. This multifaceted approach enabled a deeper understanding of the complex dynamics and trade-offs within the energy system. By considering multiple models and indicators, UNSEEN was able to set up a modeling workflow for more nuanced analyses to answer complex questions that had previously remained unanswered.

With regrad to the challenge of improving solvers for MIP in energy systems research, the project emphasized the integration of deep neural networks and reinforcement learning techniques. The corresponding REMix optimization models has been extended to generate scalable model instances that capture the discrete decision variables in energy system analysis in order to test the combination of neural networks and optimization algorithms for reducing the computational burden.

In conclusion, the contributions to the field of energy system modeling and analysis are the following. First, exploring a wide parameter space, second, integrating multiple modeling tools, and third leveraging cutting-edge technologies to generate robust insights into uncertain energy futures in order to assess and navigate the diverse development paths of the energy transition, leading to more informed and effective policy decisions.

1.3 Structure and authorships of this document

This compendium summarizes the research activities conducted within the UNSEEN project. Its remainder is structured by the focus of research of the cooperating institutions: At the beginning, we take the overall perspective and introduce the workflow that integrates all of the relevant parts of UNSEEN (section 2). It has been implemented at the supercomputer JUWELS by Thomas Breuer from the Jülich Research Center (FZJ). The first part of this HPC-Workflow is the scenario driver which has been developed by the GAMS Software GmbH (section 3) represented by Aileen Böhme and Fred Fiand. The outputs from the scenario driver are used to create instances of the ESOM REMix (section 4.1). This work was mainly the responsibility of Shima Sasanpour and Jan Buschmann who work with the German Aerospace Center (DLR), Institute of Networked Energy Systems. DLR was also responsible for preparing the inputs to the scenario driver. The corresponding work is documented in section 4.2 by Sonja Simon and Mengzhu Xiao. Based on the optimization results observed for the solved REMix instances Kai von Krbek, Jan Buschmann and

Ulrich Frey implemented an evaluation approach consisting of the statistical analysis of indicators (section 4.4.1) which have been processed by specific indicator models (section 4.4). The results of this analysis are presented in section 4.5 by Ulrich Frey. In section 4.6, we present the enhancements to the REMix framework accomplished by Manuel Wetzel and Shima Sasanpour. One reason for these enhancements has been the provision of scalable instances for both LP and MIP based ESOMs. On the one hand, they were required for the further development of the solver PIPS-IPM++ (section 5) by Nils-Christian Kempke from Zuse Institute Berlin (ZIB). The corresponding performance analyses have been conducted by Thomas Breuer and are presented in section 6. On the other hand, a new framework for solving MIP instances based on heuristics has been developed by Charlie Vanaret from Technical University of Berlin (TUB). This work is presented in section 7 together with the findings of the attempt to use reinforcement learning strategies for this purpose, initiated by Lovis Anderson.

Chapter 2

Workflow and data management

2.1 Workflow

2.1.1 Overview

The goal of this project was to develop a workflow that couples different models (REMix and AMIRIS) by automatically generating and solving a large number of scenarios (see Figure 2.1). For this purpose, existing software packages were further developed and additional packages were newly developed. This ensured that, on the one hand, the interaction of the individual components of the workflow works on the technical level and, on the other hand, scientific knowledge can be gained. Thus, the individual components can essentially be divided into two categories:

1. Applications: Scenario driver (3), REMix (4.1), PIPS-IPM++ (5), AMIRIS (4.3.1), indicator models (4.4.1)
2. Linking tools: JUBE (2.1.3)

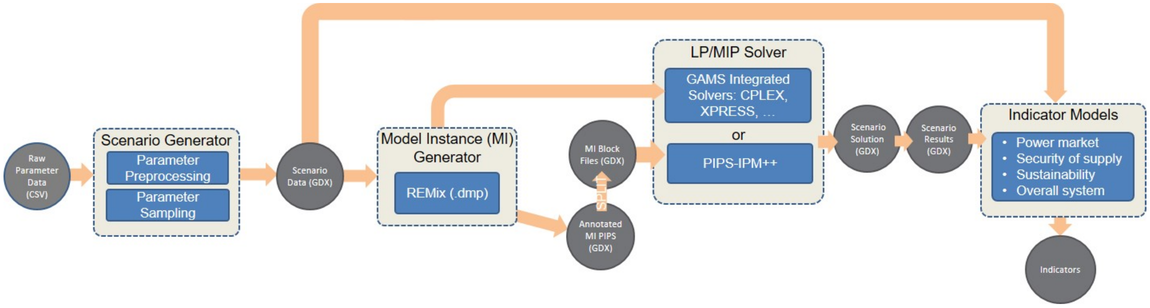


Figure 2.1: Overview of HPC Workflow

2.1.2 HPC-Workflow

The components of the workflow to be developed had never been used together in this constellation before. Therefore, the technical and content requirements of the individual programs had to be collected and analyzed first. It turned out that, in addition to the existing software stack on the HPC system, several other packages would have to be installed on which the workflow components depend. The challenge here was to install an environment that would allow the different components to run together. In particular, it was important to ensure that the versions of the Python packages were compatible. This had the positive side effect that the software packages under our control were updated to the latest version. However, every time a software change was made, the impact of these changes on the entire workflow had to be considered so that it could continue to run robustly. This continuous development could only be ensured by a clear definition of the interfaces between the components and implicit assumptions within the software packages and by very good communication between the developers involved. This was certainly the biggest challenge, because first of all a common language had to be found; e.g. an energy system modeler understands something different by a "node" than a computer scientist in his everyday professional use of this word.

The requirements analysis has also shown that a data and folder structure needs to be developed (see section 2.2) and that the selected linking tool JUBE provides a very good basis for controlling the workflow process (see section 2.1.3).

2.1.3 JUBE

While the individual applications to be coupled in the workflow are described in other chapters, in this section we will take a closer look at the tool JUBE (Breuer, Lührs, Klasen, & Wellmann, 2022), which links these applications together.

JUBE is a generic, lightweight, configurable environment for systematically executing, monitoring and analysing application execution. It provides a script-based framework that is commonly used to run benchmarks and complex workflows on, but not limited to, high performance computers, and to evaluate the results.

2.1.3.1 Main concepts

The functionality of JUBE is essentially based on three main concepts which are independent of the UNSEEN project but used for the concrete implementation of the UNSEEN-workflow:

- Parameter space and workflow creation
- Substitution of placeholders
- Unique directory structure

Parameter space and workflow creation JUBE allows in a very simple way to create a large parameter space on the basis of which the desired workflow is configured. To do this, the user can define *parametersets* which contain a list of *parameter*s. Each *parameter* can be assigned a single value or a list of values, whereby this list can be separated by any character (in the standard case, the comma serves as a separator for a list). In the generic example 2.1, the parameter *const* is assigned a single value and the parameters *p1* and *p2* are each assigned a list of values. However, these *parameter*s are only defined in the *parametersets* and are not further processed at this point. The actual workflow generation takes place in the so-called *step*s. These *step*s include the *parameterset*s with the *use* argument and form the basis for the individual work packages that make up the workflow. Furthermore, the *step*s also contain the commands specified by the user which are to be executed for this work package (indicated by the *do* keyword). The combination of *parametersets* and *steps* results in the work packages shown in Figure 2.2. For the *step preprocess*, one work package is created as there is only one parameter with one value. However, since in the *compile* step the parameter *p1* has two values, two work packages are created by this step, one for each parameter value. Due to the dependency of the *execute*-step (on *preprocess* and *compile*) specified by the *depend* argument, on the one hand the parameters from their work packages are available and on the other hand it is also ensured that the associated work packages are also executed before the execution of the work packages of the *execute*-step is started. Another special feature of this step is that work packages are generated for all combinations of parameter values. Thus, a cross product of all values is formed. In this way, by including parameters in steps and defining dependencies between these steps, JUBE can generate a large number of work packages and execute them systematically one after the other. The main concept for parameter space and workflow creation presented here in a general example forms the basis for setting up the complex multi-step workflow in the UNSEEN project.

```
parameterset:
  - name: preset
    parameter: {name: const, _: 0}
  - name: compset
    parameter: {name: p1, _: a,b}
  - name: execset
    parameter: {name: p2, _: 1,2}

step:
  - name: preprocess
    use: preset
    do: ./run_prep_cmd
  - name: compile
    use: compset
    do: ./run_comp_cmd
  - name: execute
    depend: preprocess, compile
    use: execset
    do: ./run_exec_cmd
```

Listing 2.1: JUBE YAML code example: parameter space and workflow creation

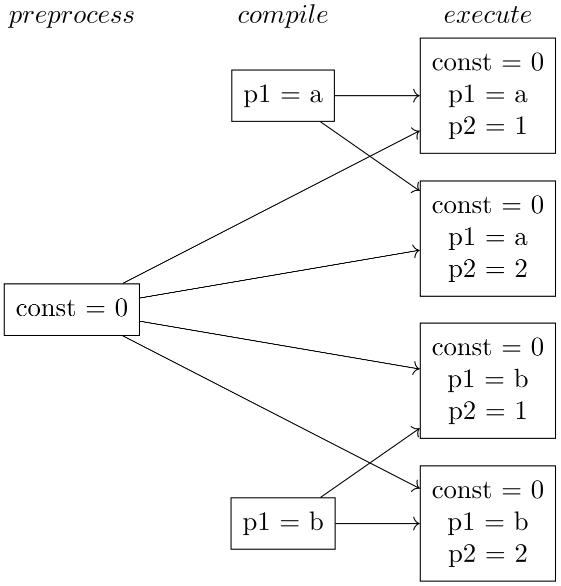


Figure 2.2: Schematic representation of the work packages created by listing 2.1

Unique directory structure A structured execution of programs with different configurations can only function reliably if, among other things, the individual combinations are not executed in the same directory so that, on the one hand, they do not influence each other by overwriting data, for example, and so that, on the other hand, it can also be retroactively traced in which configuration the respective results were achieved. Furthermore, in such a complex use case as the workflow developed in this project, in which several steps build on each other in a dependent manner and these steps are executed in parallel for different configurations (in this case scenarios), it must be ensured that both the data of the preceding steps can be accessed consistently in the sequence of steps for calculating a scenario and that scenarios running in parallel can be found unambiguously and that their data do not influence each other. JUBE solves this problem by executing each work package (see previous section) in its own folder and, as soon as a dependency is defined between work packages, symbolic links are automatically created in the respective directories so that, for example, data or programs from the preceding steps can be accessed via the links. In addition to these directories, a few files are also created, which, for example, store the concrete configuration with which the respective work packages were executed, or which enable JUBE to continue the workflow after the optimization of the scenarios has been carried out on the computing nodes of the supercomputer. In addition, these individual directoriess are combined for each JUBE run under one main folder, which is named with an ascending index. In this way, a hierarchical folder structure is created, which is exemplified in figure 2.3.

Substitution of placeholders Another main concept of JUBE is the replacement of placeholders in text files, usually using the values of the set parameters. For this purpose, first the path to the

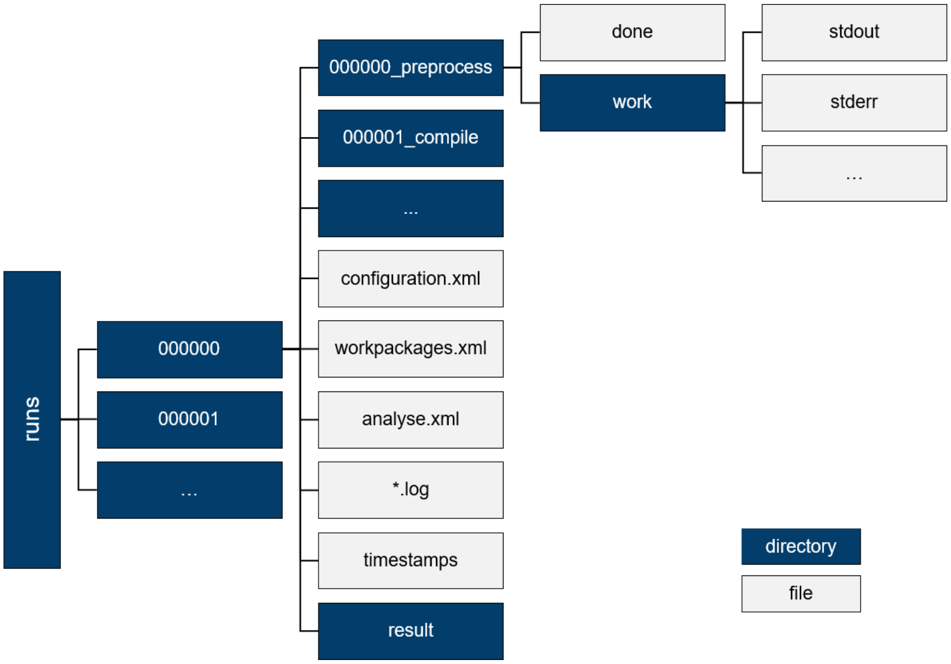


Figure 2.3: Schematic representation of a JUBE directory structure

file whose content is to be replaced can be specified in a *fileset* in order to copy this input file into the working folder of the corresponding work package. In a *substituteset* the file name is specified in which to search for placeholders and additionally the file name of the file resulting from the parameter substitution. Depending on how the name of the output file is chosen, either the input file is replaced if input and output file have the same name, or a new file is created so that the input file remains unchanged. The example 2.2 shows how the placeholder *#NUMBER#* in the file *file.in* (see 2.3) is replaced by the value *1*, creating the file *file.out* (see 2.4).

```
parameterset:
  name: param_set
  parameter: {name: number, type: int,  _: 1}

fileset:
  name: files
  copy: {source_dir: blub, _: file.in}

substituteset:
  name: substitute
  iofile: {in: file.in, out: file.out}
  sub: {source: ”#NUMBER#”, dest: $number}

step:
  name: substitute_step
```

```
    use:
      - param_set      #use existing parameterset
      - files          #use existing fileset
      - substitute     #use existing substituteset
    ...
```

Listing 2.2: JUBE YAML code example: text substitution

```
Number: #NUMBER#
```

Listing 2.3: content of the file *file.in*

```
Number: 1
```

Listing 2.4: content of the file *file.out* after the value has been substituted

2.1.3.2 New functionality

With the main concepts described in chapter 2.1.3.1 and the use of JUBE for several years in different supercomputer environments, JUBE already brings essential functionalities for use in this project. However, JUBE has so far mainly been used for a small to medium number of work packages that were processed sequentially one after the other. However, since a large number of scenarios are to be processed in this project and these scenarios each require several data processing and calculation steps, a high number of work packages that have to be processed with JUBE is to be expected. In order to accelerate this execution and to simplify the further processing of the results, the extensions described in the following two sections were implemented in JUBE especially for this project and released after a successful test phase both in the project and for all other JUBE users.

parallel execution of workpackages In the standard case, JUBE creates a list of work packages step by step and processes them sequentially. However, there are many use cases in which these work packages can be processed at least partially in parallel, as they are independent of each other. Thus, in our case, there are two steps at the beginning of the workflow that prepare general data and are executed once for all scenarios. This is followed by four more steps that build on each other and are executed separately for each scenario, but have no dependencies between the scenarios. Therefore, the work packages of a step can be carried out in parallel. To make this possible, JUBE was parallelised within the framework of this project. The implementation of this parallelisation was more complex than initially assumed, as JUBE is implemented in Python and the internal data processing is designed for serial processing. The *multiprocessing* package [1] was chosen for the parallelisation, as it is already included in the Python distribution and thus the dependency on another library could be avoided. The first prototype developed was already convincing in small test cases. However, when used with a large use case, it had to be determined that the memory requirement of the new parallel variant of JUBE increased disproportionately with an increasing number of work packages, so that JUBE stopped at the memory limit. After a thorough analysis, the cause of this increase was found and successfully fixed, so that this extension can be published as open source and can also be used for large use cases.

[1] https://docs.python.org/3/library/multiprocessing.html

Using the UNSEEN HPC workflow as an example, the advantages of this parallel implementation of JUBE in practice will be shown below. In the first step, *prepscen*, data is prepared once, which is required, among other things, for parameter sampling in the step *genscen*, in which e.g. 100 scenarios are then prepared. Subsequently, these scenarios are annotated in the step *gen2GDX* and split up by means of *gmschk* in order to prepare them for the solver *PIPS-IPM++*. After the scenarios have been solved individually, they are finalised by the steps *postsolve* and *indicators*. In the sequential flow of the original JUBE variant, the work packages for the individual scenarios are processed one after the other, so that the total runtime of the workflow results from the sum of the runtimes of all work packages (see figure 2.4).

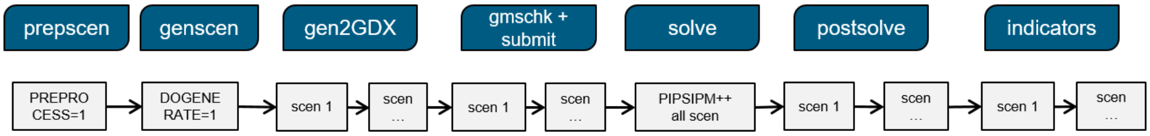


Figure 2.4: Sequential processing of the UNSEEN workflow by JUBE

In contrast, in the new parallel variant of JUBE, the independent work packages of a step can be processed in parallel. In Figure 2.5, 20 parallel processes were selected for the corresponding steps for this purpose, so that each process only had to process $\frac{100}{20} = 5$ of the total of 100 scenarios. The new runtime is thus calculated from the sum of the runtimes for processing five scenarios and is thus reduced by a factor of about 20. Depending on the selected configuration, this factor can be correspondingly smaller or larger.

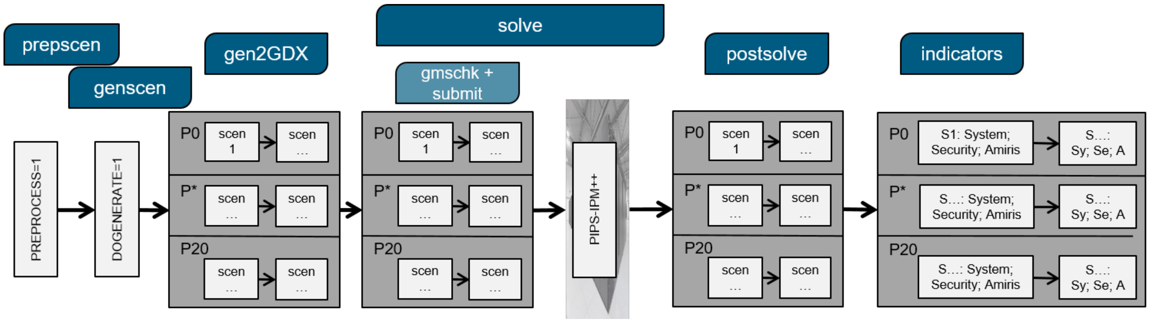


Figure 2.5: Parallel processing of the UNSEEN workflow by JUBE with 20 processes

result table as database Another function of JUBE is the generation of result tables based on the defined parameters and the reading of data from files that were generated during execution, for example. These tables can be output either in csv format or formatted for better clarity. Both output forms are simple text files that have to be completely rewritten as soon as entries are subsequently changed or these tables are viewed again and again during the execution of JUBE, e.g. to observe the progress of the execution or to see the first results. By extending JUBE with the possibility to write these result data into a database, this problem can be circumvented, as individual entries can be adjusted via the primary keys of a database without having to recreate the entire database. The implementation was done by integrating the *sqlite3* library [2], which is also included in the Python distribution. After a successful test phase of the implementation, this extension was published as open source and thus made available to all JUBE users. This provides a new feature that has already been requested by several JUBE users. Within the UNSEEN workflow, it is used, among other things, to save the paths to the individual files, the content of which is further processed in the downstream statistical analysis.

2.2 Data management

Due to the scope of the multi-stage HPC workflow, the integration of different software packages that have never worked together in this form before, and the large amounts of data to be expected, it became clear at the beginning of the project during the requirements analysis that a data and folder structure would have to be developed that clearly defined where which data originates and thus also where the individual programs find the data they need. Although the concrete implementation of the workflow was still unclear at this early stage, in retrospect it proved particularly helpful to have dealt with data management at an early stage. Thus, by the time of writing, hundreds of thousands of folders and several million files had already been evaluated to generate results, not including the data generated in the implementation and testing phases.

The essential design decision was to store the scenario data and results separately from the log files, as the latter are only needed for the development phase and the execution of the workflow, but do not necessarily need to be kept for the long-term storage of the remaining data. For this reason, in addition to the already existing folder structure required by JUBE, another hierarchical structure was developed to store the scenarios and results. Figure 2.6 shows the general structure as an example. In the folders *preprocess* and *gams_wdir*, mainly data is stored that can be used to trace which data sets were used to create the respective scenarios. The individual scenarios and their associated results are grouped in blocks in further subfolders (*scen_1_X*, where X is e.g. 100). To avoid having to search through the individual subfolders for further processing, JUBE creates a database with paths to all relevant files. This structure also enables JUBE to pass the correct paths to the corresponding input and output files to the programs used in the individual steps.

[2] https://docs.python.org/3/library/sqlite3.html

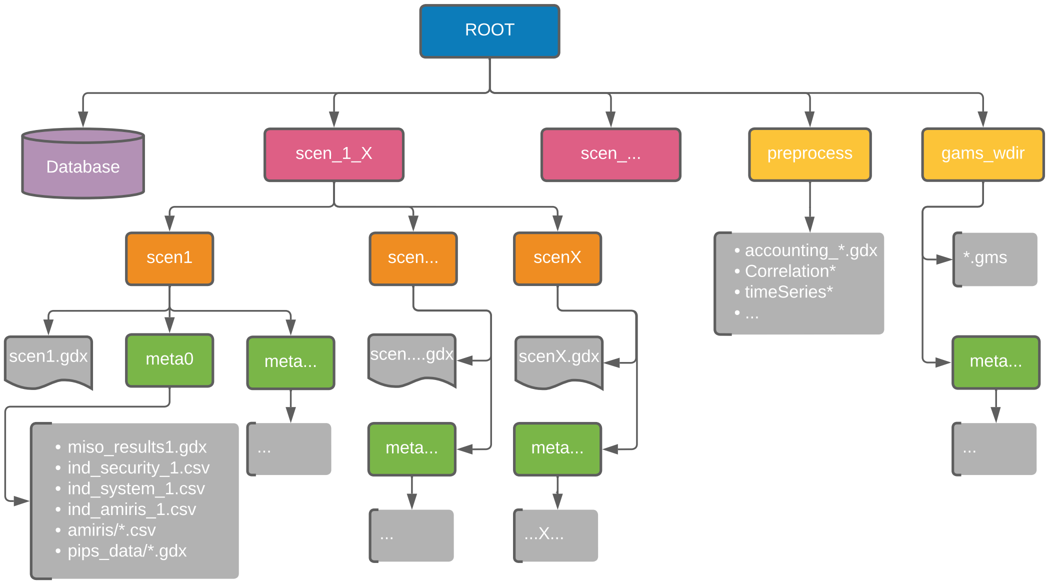


Figure 2.6: Hierarchical directory and file structure on which the workflow is based

Chapter 3

Scenario Driver

3.1 Definitions

3.1.1 Scenario

We define a scenario as a possible and plausible state of a future energy system. In the following, we refer to scenarios in two regards. On the one hand *data scenarios*, which represent a plausible combination of input data to energy system optimization modeling (with REMix). On the other hand, *solved scenarios* describe the composition of a future energy system in terms of installed capacities for power generation, transmission and storage. To observe a *solved scenario* REMix models process a large amount of input data.

3.1.2 Parameter Space

For a subset of the input data *solved scenarios* are considered to be sensitive. For these parameters we defined a so-called parameter space out of possible parameter values from the literature. Remaining parameters stay fixed. Parameters that belong to the parameter space can be varied within the predefined ranges. In this context, a *data scenario* is a sampled point in the parameter space, i.e., all parameters in the parameter space have a particular value assigned.

3.1.3 Scenario Driver

The Scenario Driver is a tool implemented in GAMS and Python used for generating *solved scenarios* from *data scenarios* using REMix.

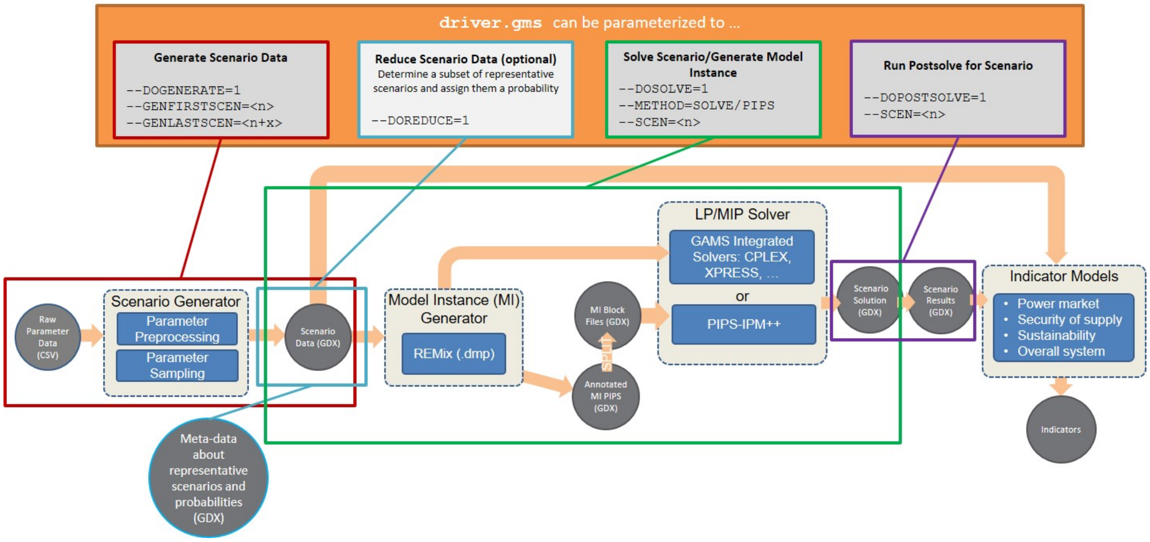
3.2 Functionalities of the Scenario Driver

The Scenario Driver provides many functionalities (see Figure 3.1). The Scenario Driver allows to

- generate *data scenario for REMix* (Scenario Generator)
- reduce *data scenarios* using forward or backward type algorithms (optional)
- solve REMix scenarios based on previously *data scenarios* using a GAMS integrated solver or generate REMix model instances for PIPS-IPM++

- run the postsolve part of REMix to process *solved scenarios* (REMix results)

Each functionality is implemented in an independent module that can be activated or deactivated when running a single file *driver.gms*. User-parameters to specify input/output files and folders (e.g., model and data file, working directory, folder for scenario data, etc.) provide high flexibility.


driver.gms can be parameterized to ...
Generate Scenario Data
Reduce Scenario Data (optional)
Determine a subset of representative scenarios and assign them a probability
Solve Scenario/Generate Model Instance
Run Postsolve for Scenario
Raw Parameter Data (CSV)
Scenario Generator
Parameter Preprocessing
Parameter Sampling
Scenario Data (GDX)
Meta-data about representative scenarios and probabilities (GDX)
Model Instance (MI) Generator
REMix (.dmp)
MI Block Files (GDX)
SPLIT
Annotated MI PIPS (GDX)
LP/MIP Solver
GAMS Integrated Solvers: CPLEX, XPRESS, ...
or
PIPS-IPM++
Scenario Solution (GDX)
Scenario Results (GDX)
Indicator Models
Power market
Security of supply
Sustainability
Overall system
Indicators

```
--DOGENERATE=1
--GENFIRSTSCEN=<n>
--GENLASTSCEN=<n+x>

--DOREDUCE=1

--DOSOLVE=1
--METHOD=SOLVE/PIPS
--SCEN=<n>

--DOPOSTSOLVE=1
--SCEN=<n>
```

Figure 3.1: Functionalities of the Scenario Driver

3.2.1 Scenario Generator

The Scenario Generator is able to generate correlated scenario data based on different probability distributions[1]. For this purpose, min/max/mean/median specifications for parameters-to-be-sampled as well as information about the interrelations among these parameters must be provided (see Section 3.3). In an independent preprocessing step the provided inputs are processed:

- CSV files are converted to compressed Gams Data eXchange (GDX) (Gams Data eXchange) files. GDX is the standard, high performance binary GAMS data exchange format.

- min/max/mean/median specifications are checked for data consistency. Errors (e.g., min>max) and warnings (e.g., min=max) are documented in a *preprocessing status file*. Constant parameter (i.e., min=max) are excluded from sampling and saved to a separate GDX file.

- A pseudo-correlation matrix is constructed from the provided parameter interrelations. This pseudo-correlation matrix is likely not to be a correct correlation matrix, i.e., is not positive semi-definite. Therefore, a positive semi-definite correlation matrix is computed that is as close as possible to the pseudo-correlation matrix. The process of constructing a positive semi-definite correlation matrix is fully documented in the *preprocessing status file*.

[1] Accordingly, for the POI analysis in section 4.5, we distinguish results based on parameter sampling with a truncated normal distribution or uniform distribution.

After preprocessing the inputs, the Scenario Generator can be parameterized to sample correlated scenario data based on the truncated normal distribution (default) or the uniform distribution. The following sections provide more details on the sampling method.

3.2.1.1 Sampling Method

The scenario data is generated using a Gaussian copula (Nelsen, 2006), (Joe, 2016). A copula is a multivariate distribution for which the marginal probability distribution of each variable is uniform on the interval [0,1]. These variables may be completely independent, deterministically related, or anything between. Because of the possibility for dependence among variables, copulas are used to construct a multivariate distribution for dependent variables. By transforming each of the variables in the copula separately using the inversion method, the resulting distribution can have arbitrary marginal distributions.

Currently, the Scenario Generator uses a Gaussian copula to generate correlated scenario data as follows: First, a multivariate Gaussian with mean vector zero and the computed positive semi-definite correlation matrix is constructed and scenario data is sampled. Second, the standard normal cumulative distribution function (cdf) is applied to get unif(0,1) random variables. Third, the uniform marginals are transformed by applying the inverse cdf of the truncated normal distribution. If no mean/median is provided for a parameter the corresponding uniform marginals are transformed to unif(min,max). Figure 3.2 shows the three step procedure for three exemplary parameters.

The Scenario Generator also randomly selects a time series file (e.g., containing time series for demand, wind, photovoltaic) from a given discrete set of time series files (see Section 3.3).

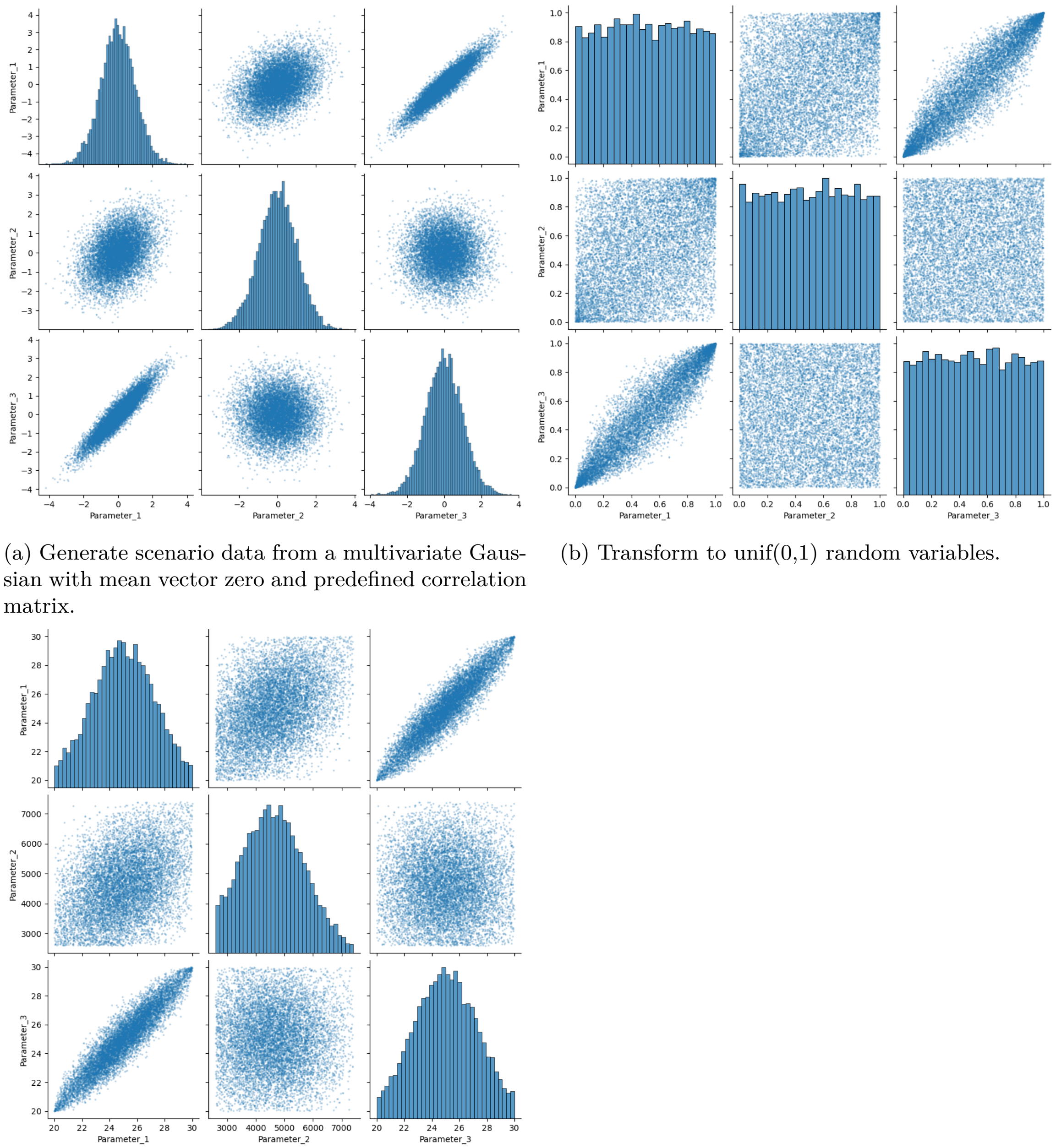
3.2.1.2 Validation of the Sampling Method

The Scenario Generator can be parameterized to run an extensive test that allows validating the described sampling method. The sampling method test generates a test sample of selected size (per default 50.000 test scenarios), calculates min/max/mean values and the correlation matrix of the test sample, and compares the results to the input, i.e., the provided min/max/mean/median specifications and computed positive semi-definite correlation matrix. Note that test scenarios will not be saved. Results of the sampling method test are documented in the *preprocessing status file*.

Running the sampling method test for 100.000 test scenarios yields the following results. All parameters are sampled within their given range [min, max]. The average relative difference between the sample mean and given mean amounts to 4.83 %. High relative differences up to 444.28 % occur where the given mean is close to the lower/upper end of the truncated normal distribution and, thus, leads to a heavily skewed truncated normal distribution. However, 98.17 % of the sample means deviate less than 10 % from the given means and 49.35 % of the sample means deviate less than 1 % from the given means. Generated histograms show the frequency distributions of sampled parameters (see Figure 3.3).

The sample correlation matrix shows that parameters are correlated as expected. 99.83 % of the sample correlations deviate less than 0.01 from the given correlations. The maximum difference is 0.0375 and the average difference amounts to 0.0025.

In summary, the test results show that the implemented sampling method allows to reliably generate correlated scenario data based on the truncated normal distribution.



(a) Generate scenario data from a multivariate Gaussian with mean vector zero and predefined correlation matrix.

(b) Transform to unif(0,1) random variables.

(c) Transform each unif(0,1) random variable to truncated normal.

Figure 3.2: Three steps to generate correlated scenario data using a Gaussian copula for three exemplary parameters.

3.2.2 Scenario Reduction

The Scenario Driver provides a module for Scenario Reduction that uses the tool SCENRED2.[2] The reduction algorithms developed in (Dupačová, Gröwe-Kuska, & Römisch, 2003) and (Heitsch & Römisch, 2003) determine a scenario subset (of prescribed cardinality or accuracy) and assign optimal probabilities to the preserved scenarios. All deleted scenarios have probability zero. The reduction method (forward or backward) is set automatically, i.e. to the method with the best expected performance with respect to running time.

3.2.3 Solve Scenario

Via the Scenario Driver one is able to either solve a scenario using GAMS integrated solve or generate a model instance (e.g., an annotated model instance in GDX format for PIPS-IPM++).

3.2.4 Postsolve Scenario

The Scenario Driver allows running the postsolve directly at the end of the solve or separately.

[2] See SCENRED2 documentation at `https://www.gams.com/latest/docs/T_SCENRED2.html`.

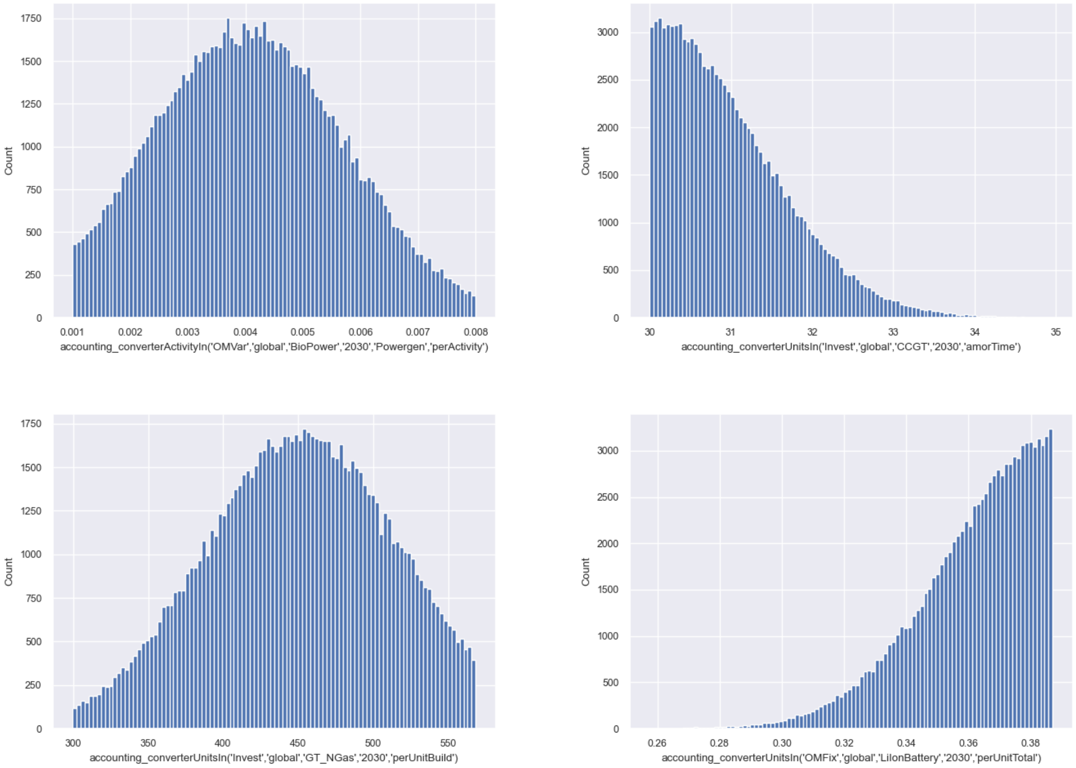


Figure 3.3: Frequency distribution of 4 selected parameters. Note that histograms are generated for all 764 sampled parameters.

3.3 Scenario Driver Inputs

Each module of the Scenario Driver processes inputs:

- **Scenario Generator:** The Scenario Generator takes a folder (the raw parameter data) that contains different types of files as input:

    1. CSV files with min/max/mean/median information. Each file contains data for a specific REMix parameter[3] (parameters are extended by a **last** dimension containing the min/max/mean/median specification). Note that the REMix parameter name is extracted from the file name, i.e., CSV files must have the following format *<parametername>.csv*. Scenario data generated for a REMix parameter will be merged into the basic dataset (at compile time), therefore, it is crucial that the name of the CSV file corresponds to the correct REMix parameter in the basic dataset.

    2. A CSV file containing (non-zero) parameter interrelations in list format. The Scenario Generator initializes a zero-correlation matrix and inserts information from the CSV file, i.e., constructs a pseudo-correlation matrix. The Scenario Generator computes a positive-semidefinite correlation matrix that is as close as possible to the pseudo-correlation matrix.

    3. CSV files containing time series. Each time series CSV file contains time series for a specific REMix parameter. For each REMix parameter multiple time series CSV files can be provided and labelled by a unique ID. Note that the REMix parameter name as well as the ID are extracted from the file name, i.e., CSV files must have the following format *timeSeries_<parametername>_<ID>.csv*. For each scenario, the Scenario Generator selects a file ID per REMix parameter and saves this information into the corresponding scenario file. Before the solve this information is used to merge the selected time series into the basic dataset.

- **Scenario Reduction:** Scenario GDX files of all scenarios that should be considered for reduction.

- **Solve Scenario:** A scenario GDX file, i.e., *scen<n>.gdx* and a dmp file (self-contained model and data file).

- **Postsolve Scenario:** A solution point GDX file, i.e. *<modelname>_p<n>.gdx* or *pips_data/<modelname>_jacobian<n>_sol.gdx*, a scenario GDX file, i.e. *scen<n>.gdx*, and a dmp file.

3.4 Scenario Driver Outputs

Each module of the Scenario Driver produces outputs:

- **Scenario Generator:** Numbered scenario GDX files, i.e., from *scen<n>.gdx* to *scen<n+x>.gdx*.

- **Scenario Reduction:** A SCENRED2 output file *SR2_OUTPUT.gdx* that contains the assigned probabilities.

[3] A REMix parameter contains data for the REMix model.

- **Solve Scenario:** The output can be either a solution point or a model instance. Per default the scenario is solved using CPLEX and a solution point GDX file numbered according to the input scenario GDX file is written, i.e., *<modelname>_p<n>.gdx*. Moreover, the Scenario Driver can be parameterized to output an annotated model instance in GDX format (i.e., *<modelname>_jacobian<n>.gdx*), a model instance in cip format (i.e., *<modelname><n>.cip*) or a model instance in lp format (i.e., *<modelname><n>.lp*).
- **Postsolve Scenario:** A result GDX file numbered according to the input solution point GDX file, i.e., *results<n>.gdx*.

Based on the numbering, solution point and result files can be precisely assigned to the associated scenario file. All GDX files are generated in the compressed GDX format.

Chapter 4

Energy System Modeling

As central model within our workflow and as basis for the creation of ESOM instances we use REMix (Renewable Energy Mix for a sustainable energy supply) (Wetzel et al., in prep.). It is a framework typically used to create models that minimize the costs (i.e. operational and capital expenditures) of an energy system's components. Typical use cases of REMix are multi-regional system design studies for the integration of high shares of renewable power supply.
In the following, we describe different REMix instances that are derived from a basic model (see section 4.1). These instances differ in their spatial resolution and the type of optimization model.[1] To a certain degree input parameters, such as the set definition of considered technologies stay the same across all instances. However, inputs we considered to be relevant for sensitivity analyses, are sampled from the parameter space (see section 4.2). Furthermore, we introduce all indicators that have been derived from the outputs of REMix (see section 4.4.1) using a set of what we identify as indicator models (see section 4.4). The statistical evaluation across all these indicators for a significant high number of workflow executions allows the identification of what we refer to as Points of interest (POI).

4.1 Description of the REMix model instances

The basic REMix model instance is a detailed representation of the German power system at the transmission grid level. It consists of 365 substation and 112 grid junction locations. The regions can be grouped together into user-defined numbers of aggregated model nodes to provide model instances of different sizes, such as the instances shown in Table 4.1. The model also considers the energy imports and exports with eleven neighboring countries, which are fixed and predetermined.

The main focus of the model is on the power sector, where various types of power plants, energy storage systems, and grid technologies are considered. It assumes a scenario where either no existing power plants are present (greenfield scenarios) or with predefined capacities, e.g., of lignite and coal power plants (brownfield scenarios). Gas-fired power plants are considered as transitional technologies and can be expanded further. Prices for greenhouse gas emissions encourage investments in low-carbon technologies.

[1] An introduction to the required adaptions to the REMix framework is given in section 4.6.

The model takes into account historical weather data from 1995 to 2018 to determine the optimal dispatch of renewable energy sources. It also considers various techno-economic parameters such as the initial investment costs, fuel costs, costs associated with CO_2 emissions, efficiencies of the technologies, as well as fixed and variable operation and maintenance costs. These parameters are determined through a sampling approach 3.2.1).

Table 4.1: REMix LP and MIP model instances

Name	**Type**	**Nodes**	**Description**
Christolph	LP	488	477 German nodes and 11 neighboring nodes
Dagolpert	LP	54	DAMP2[a] region with high resolution (33 model nodes), 10 nodes for the rest of Germany and 11 neighboring nodes
Edelpraut	LP	109	North Rhine-Westphalia with high resolution (88 model nodes), 10 nodes for the rest of Germany and 11 neighboring nodes
Gunilpert	LP	270	North Rhine-Westphalia and half of Germany with high resolution (253 model nodes), 6 nodes for the rest of Germany and 11 neighboring nodes
Benemipt	MIP	12	One German node and 11 neighboring nodes
Clemip	MIP	13	Two German nodes and 11 neighboring nodes
Achmip	MIP	54	DAMP2[a] region with high resolution (33 model nodes), 10 nodes for the rest of Germany and 11 neighboring nodes
Emipraut	MIP	109	North Rhine-Westphalia with high resolution (88 model nodes), 10 nodes for the rest of Germany and 11 neighboring nodes
Gunmipert	MIP	270	North Rhine-Westphalia and half of Germany with high resolution (253 model nodes), 6 nodes for the rest of Germany and 11 neighboring nodes

[a] DAMP2: sub-region of North Rhine-Westphalia including the cities of Bottrop, Düsseldorf, Duisburg, Essen, Gelsenkirchen, Herne, Krefeld, Mülheim an der Ruhr, Oberhausen, Wesel, Bochum, Borken, Ennepe-Ruhr-Kreis, Kleve, Leverkusen, Mönchengladbach, Mettmann, Recklinghausen, Remscheid, Rhein-Kreis Neuss, Solingen, Viersen, Wuppertal

4.2 Definition of the Parameter Space

4.2.1 Parameter Space

One central objective of UNSEEN is to assess a broad range of scenarios, defined by a large set of parameters as input variables. In order to provide a meaningful and relevant scenario assessment, we created a parameter space that limits the input variables for REMix modeling. The parameter space provides upper and lower limits for a set of parameters to the scenario generator. Based on these limits the scenario generator defines a variety of scenarios input sets, which are eventually calculated with the improved REMix model. The results of these model runs are then analyzed extensively in section 4.5.

Basis of the parameter space is an extensive literature research on existing scenario studies, both on the German and European level. Additionally, we include expert knowledge from the previous energy scenario development exercises and projects within the DLR in the field of scenario analyses.

4.2.1.1 System boundaries

The scenario generator relies on the parameter space to limit the modeling input to meaningful data. Therefore, the first exercise was the joint definition of system boundaries within the project, covering geographical, temporal and technical aspects.

Area and countries REMix focuses on European and German level, setting the scene also for UNSEEN. With the focus on methodological improvements we further limited the approach specifically to Germany and its neighboring countries. Furthermore we distinguished between generic parameters, which are applicable for all (European) countries in the modeling exercise and parameters, which are defined each country or region. While all technology specific data is covered by a generic data set, the demand and supply related parameters as well as potential data were distinguished for the following countries and regions:

- Germany
- France
- Poland
- Denmark
- Switzerland
- Aggregated Czech Republic, Austria, Belgium, Netherlands (CZ, BE, NL, AT)
- EU total

Time range For the parameter space in the UNSEEN project a comprehensive data set for the year 2050 was collected. Additionally, we covered the complete time range before that date. Many parameters feature a specific development over time, e.g. for investment costs or power demand in the various sectors. Therefore, we provide additional information on parameter space limits from 2015 to 2050 years in 10-year steps. However, these data were not always available for each parameter. Additionally, a specific data set for variable renewable energy potential from solar and wind was provided for the years 1995-2018 with an hourly resolution.

4.2.1.2 Selection of parameters

In a first step, the REMix modelers provided a list of input parameters, which are essential or relevant for defining model runs, as well as desirable as information for the scenario evaluation procedure. The following table summarizes the selected input parameters.

There exists a broad range of literature for the generic, more technology focused parameters from which the parameter space can be selected (see section 4.2.1.3). However, for the country specific parameters it is a challenge to provide consistent data across all time steps and/or for all countries. We therefore aggregated parameter values from various studies, as described in section 4.2.1.4. Additionally, we screened those studies for parameters for the ex-post-evaluation, predominantly extracting power generation and capacity data.

Table 4.3: General and Country-specific Parameters

Parameters	**Unit**	**Comment**
General Parameters		
Specific Investment Costs	€/kW	for Grids: €/km
Operation & Maintenance costs fixed	€/kW*a	
Operation & Maintenance costs variable	€/GJ	
Efficiency	%	gross
Fuel costs	€/GJ	if not in O&M variable
CO_2 price	€/Gt	
Country-specific Parameters		
Power demand on sectoral level	PJ/a	by country and sector (power, heat, transport)
Biomass potential	PJ/a	by country
Wind onshore potential	PJ/a	EnDAT by different years
Wind offshore potential	PJ/a	EnDAT by different years
PV potential	PJ/a	EnDAT by different years

Specifications This literature review was then condensed to provide the boundaries and specifications for the parameter space. Basic information are the selection lower and upper limits for the selected parameters from literature. Additionally, we extracted key indicators on the distribution or frequency, which will assist the scenario generator in selecting appropriate increments for each parameter.

Our literature review provided us with a large data set of values for the above described parameters. However, neither the scenarios, nor the parameter values can be classified as independent or normally distributed. There are strong dependencies between various scenarios studies successively building on each other. Based on the objective, the scenarios differ in story-line, the underlying assumptions and level of ambition towards the energy transition. Also, the number of scenarios, available time steps, extractable values for each parameter and, in general, the transparency vary strongly between studies. Moreover, the scenarios feature different approaches: An exploratory versus a normative approach; an optimization versus a simulation approach; a technology, demand or infrastructure focused approach. Therefore, a straight forward statistical assessment of distribution and frequency is difficult. Thus we selected only a few descriptive key figures, first and foremost the minimum, maximum and median, which is less susceptible against non-comparable numbers of scenarios and missing data (see table 4.5). In order to evaluate this non-normal distribution, the median was calculated as well.

Specifically, the demand parameters between and within sectors are not independent. To represent the dependencies, we evaluated characteristics in terms of shares of a sector or a technology, e. g. the shares of heat pumps in total heat demand for buildings or the share of power demand for the production of synthetic fuels in total power demand. Here, we identified their minimum and

Table 4.5: Specifications for defining the parameter space

Parameter	Description
Min	Minimum of range from literature
Max	Maximum of range from literature
Mean	Mean value from literature
Ref	DLR Reference value (expert judgment)
Median	Median from literature
Max+/Min-	Outliers
Middle	One consistent scenario near the Median
Upper	One consistent scenario near the Max
Lower	One consistent scenario near the Min

maximum of shares respectively. The scenario studies are very diverse and demand development is subject to strong path dependencies. Therefore, we also identified consistent demand pathways from one scenario, representing middle, lower and upper demand developments.

4.2.1.3 Selection of relevant Technologies and techno-economic data

The REMix model covers a large variety of energy technologies which have to be covered in the parameter space. In order to model the energy system, the following technologies are selected and classified as conventional, decentral, central, grid and storage (see table 4.6). For these technologies we carried out an extensive literature research.

Table 4.6: Classification of Technologies for parameter space

Conventional	Decentral	Central	Grid	Storage
Open cycle gas turbines	Photovoltaic (PV)	Offshore wind	HVAC overhead transmission	Lithium batteries
Combined-cycle gas turbines	Onshore wind	Pumped hydro	HVDC overhead transmission	Vanadium redox flow batteries
Nuclear power plants	Run of river	Electrolyzers	HVAC underground cables	Compressed air batteries
Hard coal power plants	Solid biomass	Fuel cells	HVDC underground cables (land)	
Lignite power plants	Biogas	Carbon capture and storage (CCS)	HVDC underground cables (sea)	

Overview over technical-economic data collection The technical-economic parameters are generic for all countries including investment costs, fixed and variable operation & maintenance costs, efficiency and life time. The minimum, maximum, average and median values for the above parameters of selected technologies are defined according to the literature review. The technological costs data are converted into present value of Euro in 2018. The parameters for storage are split into storage units and inverter units. Assumptions (numbers in gray) are made when values are not available for all the scenario years. This is either an average or it is assumed constant if no variations from the available data is present. NA means that no data is available.

4.2.1.4 Selection of country specific parameters and data collection

For power demand and renewable energy potentials we extracted country specific data from literature, predominantly based on study reports and scientific papers. While solar, wind and hydrogen potentials are modeled endogenously within the project (see section 4.2.1.5), biomass potentials were provided externally.

Biomass Biomass potential analyses vary widely, depending on the covered primary or secondary biomass sources, the disaggregation to power heat or transport, the applicable technologies, and sustainability issues. While a variety of studies provide country level potentials, most cover different categories of the above mentioned aspects. We therefore selected one European wide study, provided by the European Joint Research Center. (Ruiz et al., 2015) cover a large range of biomass classes from energy crops over forest products to residues, according to the natural conditions in each European country. Due to the variability of biomass production, they also provide a range of scenarios addressing minimum, maximum and intermediate potentials for each country in Europe. Moreover, they assess prospective potentials until 2050. Thus, this study already provides a parameter space for biomass potentials which can be directly translated to the UNSEEN project. Based on this extensive data source, we disaggregated the primary biomass potentials to sectors, in order to derive a specific available for power production, which serves as an input to REMix. Even though many biomass classes provide input to power, heat and transport, we applied a simple rule of thumb to derive power potentials: All potentials specifically dedicated for biogas were clustered to a biogas potential, which is completely available for power and CHP applications. Additionally, direct and indirect forestry products, municipal waste, straw and industrial wood residues were aggregated and solid biomass potentials. According to the European Association for Biomass (AEBIOM, 2019), the share of solid biomass in large industrial applications is 30% in the EU. We used this share as an approximation for CHP applications, allocating 30% of solid biomass in each country as a potential for REMix.

Fuel Prices For a parameter range in fuel costs we mainly evaluated the World Energy Outlook Series from IEA from 2012 to 2019, identifying the 2014 and 2019 reports as the boundaries for the lowest and highest fuel prices for oil, coal and gas in Europe up to 2040 (IEA, 2014, 2019). We additionally evaluated fuel costs for Germany, finding considerably higher prices for industrial applications for oil, gas and coal from (Elsland et al., 2017), which we added to the parameter space under the Max+ specification. All selected studies (Elsland et al., 2017; IEA, 2014, 2019) only provide projections until 2040; therefore we assumed the same prices for 2050 for the lower boundaries and slightly higher prices for 2050 for the upper boundaries of the price spread.

Additionally, the REMix model requires prices for lignite, uranium and biomass. However due to the phasing out of nuclear and coal in Germany, the prices have a smaller influence on the optimization in 2050. We therefore selected available prices from previous projects (Nitsch et al., 2012) a reference value, which will then be varied by +/-25% for the parameter space.

Biomass is currently the only renewable energy source which requires an external price input in REMix. Biomass covers a broad range of substances, from residues to energy plants, which can be used in a similarly broad range of technologies. However, REMix aggregates this range for biogas on the one hand and solid fuels on the other hand. This leads to a high uncertainty for price assumptions for biomass, which is even increased by the interrelation of biomass and fuel prices (Lesche, 2022). Moreover, future biomass prices are strongly related to the available potential of the respective biomass type. We therefore selected currently available price assumptions for the largest section of the solid biomass - wood. We chose the current price range from waste wood to wood pellets as the boundaries of the parameter space, also for future scenario years. The price range is based on assumptions from 2019 by the DBFZ (German biomass research centre) (Thrän et al., 2019).

CO_2 Prices For the range of CO_2-Prices we refer to the (IEA, 2019), which provides an intermediate price range from 40-140 \$/t CO_2 in 2040. As these prices are rather low concerning the currently necessary goals in CO_2 mitigation, we added the price assumptions of 240 \$/t CO_2 as an upper range, based our own analysis from the MuSeKo project (Fette et al., 2020).

Power demand

- Basis for the parameter space in UNSEEN are two sets of scenario studies: One for Germany and one for Europe – were available on the country level.
- Procedure of literature review: Selection of Scenarios, extraction of ranges, Selection of specifications for additional information
- Demand data: Here we identified significant differences between scenarios with 8% CO_2 reduction and ambitious scenarios with 90-100% CO_2 reduction.

4.2.1.5 Time series for RE potentials

Renewable energy potentials for solar, wind and hydro result from the REMix-EnDAT model, which has been developed at the German Aerospace Center for more than a decade (Scholz, 2012; Stetter, 2014). Detailed time series have been modeled for the years 1995-2018.

4.2.2 Interrelation Matrix

The parameter space provides a large range of possible input data for scenario calculations. In order to enhance the consistency of these calculated scenarios we further limit the parameter space by identifying interrelations between the parameters. We identify parameters, which either are expected to have either a positive correlation (factor B is increasing if factor A is increasing) or a negative correlation (factor B is decreasing if factor A is increasing). This reduces the variability of the parameters, the number of possible scenarios and thus calculation time. Moreover, it avoids scenarios with contradicting parameter values and leads to a more consistent sets of parameters for the scenario generator.

We developed an interrelation matrix, which represents logical relations as identified by expert judgment. Disclaimer: This is not to be confused with a statistical correlation matrix, based on hard data and statistics – which are not available for future values. Our interrelation matrix instead is based on assumptions on future developments and expert assessment.

Basis of the interrelation matrix are the 79 parameters which were identified for the parameter space. The interrelations are based on information from a previous project (Garcia-Teruel et al., 2022), which assessed relations between parameters including a cross impact balance (Weimer-Jehle, 2023) and expert judgement for a) selecting related parameter pairs and b) for evaluating the relation between each pair quantitatively.

The complete data set for the evaluated interrelation matrix can be found in (Simon & Xiao, 2022).

We selected the following tags for the interrelations (table 4.7): The default value for all pairs is set to 0. Where no interrelation could be identified or the direction of interrelation is unclear we also selected 0 to allow for a complete variation between min – max. Only for values from -2 to 3 the variation will be reduced in the scenario generator according to the tags in table 4.7.

Table 4.7: Available tags for quantifying interrelations in the interrelation matrix

Tags	**Value**
Fixed to other parameter	5 (3)[b]
Strong positive relation	2
Weak positive relation	1
Weak negative relation	-1
Strong negative relation	-2
Not correlated	0
Relation unclear	0
No information available	0

[b] The original tag awarded was 5 for parameters that are fixed to another (e.g. oil price and gas price); for data integrity this value was later changed to 3 for calculation in the scenario generator, without changing the definition.

In a first step we selected the relevant parameter pairs for further quantification out of the 3,081 possible parameter pairs. A first selection was based on the previously analyzed parameters in paper). After analysing them, we selected the DSM-REMix for further details. We first reduced this matrix to the parameters that roughly matched our parameters (see table 4.8).

The matrix identified power demand parameters as most influential parameter on other parameters, next to fuel and CO_2 prices. The efficiency, lifetime and specific investment and variable costs showed no influence on other parameters, except a low influence on power demand. Thus, we included these parameters in our further quantification. Unfortunately, the matrix does not distinguish between positive or negative correlations. Therefore, we used it only as a guidance for the further expert evaluation of the interrelations, which focused on the parameters with high impacts and for selecting relevant interrelations for further assessment.

The UNSEEN project team selected additional parameter pairs, which are likely to have relations. Therefore, the matrix was assessed pair by pair by three energy system experts from DLR, relying the above mentioned paper and expert judgment. Tags were awarded for about 70 parameter pairs, where the experts agreed. Another set of parameter pairs with unclear relation were selected by the team for further analysis.

Expert survey workshop We selected another 48 parameter pairs with unclear relations for further discussion with experts from the energy systems analysis at DLR. We carried out 3 online surveys (S1, S2, S3), each covering 16 parameter pairs. The contributing experts awarded the same tags as mentioned in table 4.8 (from -2 to 5) for each pair. The survey results support that there are several options for logical relations between the selected parameter pairs. However, for an

Table 4.8: Specification matrix

	Efficiency	Life time	Investment costs and fixed operation costs	Fuel prices	CO_2 certificate prices	Annual power demand of additional consumers	Annual conventional power demand	Variable costs
Annual conventional power demand	3.86	3.86	3.86	0.67	0.94	33	47.2	0
Annual power demand of additional consumers	2.75	2.75	2.75	0.67	0.94	29.8	34.8	0
Fuel prices	1.28	1.28	1.28	0.67	0.44	5.81	11.9	0
CO_2 certificate prices	1.28	1.28	1.28	0.67	0.44	6.06	12.5	0
Efficiency	0	0	0	0	0	2.5	2.5	0
Life time	0	0	0	0	0	2.5	2.5	0
Investment costs and fixed operation costs	0	0	0	0	0	2.5	2.5	0
Variable costs	0	0	0	0	0	0	0	0

additional 15 pairs of parameters, the surveys provided a clear picture. Here, more than half of the participants (n > 13) agreed on the direction of the relation (negative, positive) or even the tags. All other parameter pairs either received an unclear assessment or were assessed as neutral and hence were awarded a 0. However, for the following 11 parameter pairs, we added a quantitative evaluation to the matrix, based on the majority of given answers by the participants.

The completed interrelation matrix, which includes quantitative values for 81 parameter pairs was provided as input to the scenario generator for further data processing, including a consistency check of the selected tags. The final version of the interrelation matrix is available in (Simon & Xiao, 2022).

4.3 Indicators

The systematic assessment of hundreds of scenarios poses a new challenge to the field of energy system analysis. In most studies only a handful of scenarios are analyzed and compared to each other. While this can still be done with a large number of scenarios, additional methods to filter and categorize scenarios are required. To address this challenge, we identified several indicators for the quantification of key aspects that characterize individual scenarios. Some of the indicators are motivated by political and technical constraints of the system. However, the indicators are limited due to the simplifications considered in ESOMs. By correlating indicators and systematical searching for outliers, we identify points of interest in the parameter space. The points of interest can then be further analyzed in order to draw conclusions on what input data or methodological constraints caused them. This allows modelers in a subsequent step to adjust the model constraints accordingly to either avoid or focus on these areas in the solution space.

4.3.1 Power Market Indicators

The following paragraphs give a short explanation of indicators to be calculated with the agent-based model AMIRIS (Schimeczek et al., 2023). They focus on economic stability of the system, thus indicating whether the given solution of solved scenario is feasible for all market participants, or hint in which direction the given solution might require additional policy measures to stabilize. To calculate these indicators, AMIRIS is simulating the dispatch decisions of the market participants based on the power plant park and further inputs provided by REMix. However, AMIRIS includes strategic behavior and bounded rationality of its actors leading to a dispatch deviating from the original findings of REMix. Thus, all dispatch-related values might differ from the original solutions found for an optimized energy system.

Market-driven total Carbon Emissions A change in the dispatch of conventional plants will also affect the total emissions of carbon dioxide. The sum of all emitted carbon dioxide found in AMIRIS can be compared to the same number from REMix optimizations. Their relative and absolute change can be used to estimate the stability of carbon emissions with respect to the behavior of actors in the energy system.

Market-driven energy not served The sum of all inelastic energy demand not served during one simulation year. The term “inelastic” refers to energy demand that cannot be shifted and, if not served, would lead to an unplanned disconnection of energy consuming devices from the electricity grid in order to maintain the grid frequency (brown out).

4.3.1.1 Supply-based Indicators

One possible goal to optimize an energy system is to minimize its system cost. This often causes energy systems to be designed without a margin for errors. Introducing such errors with an agent-based simulation can then lead to energy and power shortages. The amplitude and frequency of such shortages are an important indicator of reserve capacities required to operate an energy system in a stable way.

Missing Peak Production The maximum of inelastic energy demand not served from all hours of one simulation year. This directly indicates what additional power would be required to operate the energy system without any brown-out events considering the sub-optimal behavior of the energy system actors.

4.3.1.2 Price-based Indicators

Strategic bidding, bounded-rationality and missing information of energy system actors are reflected in the bids and asks at the energy exchange and thus directly impact electricity prices. Those, however, are connected to household and industry electricity prices and the revenue potential of market participants.

Hours with maximum Prices There is an upper limit for prices at the energy exchange. Typically, this price is reached only in case the inflexible load cannot be served in total. The count of all hours where the exchange's upper price limit is reached indicates how often brown-outs may occur and thus provide insights regarding system adequacy. Additionally, revenues of flexibility operators can rise significantly in these hours. If one assumes that a brown-out will be prevented using (not modeled) power generation reserve capacities, revenues for flexibility providers might drop significantly. Thus, the count of hours with maximum prices is an indicator for the revenue stability of flexibility option providers. It is calculated by using data of the AMIRIS simulation in a post-processing script.

Average Electricity Price The amount-weighted electricity price at the energy exchange averaged over one simulation year is correlated to the household and industry electricity prices and is thus an ideal indicator for consumer costs of electricity. It is calculated in AMIRIS.

Electricity Price Volatilities The volatility of electricity prices is an important indicator for the revenue potential of flexibility option providers. The consideration of the storage's time scale (i.e. energy to power ratio) of the flexibility options, however, is crucial for this indicator to have any meaning. Since different flexibility option technologies may operate on different time scales multiple price volatility indicators will be generated for varying storage times from one hour to one year. Valuable conclusions about cannibalization effects of flexibility options can be drawn from this indicator, especially when compared to their technology-specific profits. It is calculated by using data of the AMIRIS simulation in a post-processing script.

Hours with Negative Prices A larger amount of hours with negative electricity prices can be a hint for possible market policy inefficiencies. Especially, when considering an integrated European electricity system with strong market coupling, an extensive count of hours with negative electricity prices can lead to unwanted shifts in behavior of connected neighboring actors not explicitly modeled in the energy system simulation. Thus, this value is weak but valuable indicator the (in-)stability of the German electricity market zone in its international context. It is calculated by using data of the AMIRIS simulation in a post-processing script.

4.3.1.3 Profit-related indicators

One condition for a micro-economically stable energy system is that all of its actors can operate profitably. If, under a given policy regime, actors cannot operate profitably, they will withdraw from the market to minimize their losses.

Profitability by technology The profitability (i.e. revenues minus expenditures over one simulation year) will be analyzed individually for each generation technology and each type of flexibility option.

Share of economic curtailment The economically curtailed Variable Renewable Energy Sources (VRES) can be used as a valuable addition to the previous mentioned indicator. Technology-specific curtailment statistics allow to infer the cause in case of low profitability of the corresponding VRES technology. This indicator will be given as relative number, i.e. the share of curtailed energy over the total potential production of that technology. Note, that currently there is no compensation mechanism in place for economical curtailment, in contrast to grid-related curtailment.

4.3.1.4 Cost-based indicators

Costs for operating the energy system are the most important aspect considered in energy system optimization models. However, cost caused by policy instruments in place are also of relevance but are often neglected. Both are to be considered as indicators and are most meaningful when compared.

Dispatch-related system costs In AMIRIS, this indicator covers all the operational costs of power generation, i.e. fuel costs, costs for co2-emission certificates, wear-and-tear expenditures and other fixed and variable operation expenditures. Costs from electricity transport and power plant setup are not considered here.

Policy instrument costs Funding of the state for active policy instruments within the simulation, e.g. reduced tax income or paid-out subsidies over the course of one simulation year. This indicator indicates if a given power plant setup is harder and/or more costly to implement in a more realistic simulation setup.

4.3.2 Overall System Indicators

System costs System costs is an important indicator in energy system modeling that quantifies the economic expenses associated with the operation, maintenance, and infrastructure development of an energy system. It encompasses the costs of fuel procurement, power generation, transmission, as well as energy storage. These costs are directly calculated by REMix.

Annual demand Annual demand is a key indicator in an energy system model that represents the total amount of energy required by consumers within a given year. The annual demand serves as a fundamental input in energy system modeling, facilitating the assessment of resource adequacy, infrastructure planning, and policy formulation. The higher the annual demand, the higher the generator capacities needed to serve that demand.

4.3.2.1 Renewable Energy Assessment Indicators

The percentage of renewable energy sources is constantly increasing in current and future (decarbonized) energy systems. Renewable-based indicators reflect this increase by considering the total share of renewable energy generation, the alignment between generation and demand time series as well as the curtailment of feed-in electricity. The indicators considered take into account the geographical and technological dispersion of renewable energy resources. They are therefore strongly influenced by the feed-in time series and have a system-wide effect both on the power network as well as system security of supply.

Renewable Energy Penetration Index (REPI) The Renewable Energy Penetration Index (REPI) is defined by the average annual demand covered by renewable energy sources (Dagoumas, 2021; Michi et al., 2019, 2018; Poncela, Purvins, & Chondrogiannis, 2018) 4.1 as follows:

$$\mathrm{REPI} = \frac{\sum_t R(t)}{\sum_t L(t)} \tag{4.1}$$

where R(t) is the sum of W(t) and S(t) representing the wind and solar generation time series; respectively. L(t) is the demand time series and t is time. The REPI index is often a key indicator for policy makers and other stakeholders (Blanco, Spisto, Hrelja, & Fulli, 2016) (such as Transmission System Operators) for setting goals and for scenario characterization as it indicates the total penetration of renewable energy sources in the power system. A higher REPI value indicates an ESOM with a high penetration of VRES. As an example, Poncela et al. (Poncela et al., 2018) evaluated REPI along with three other indicators for 25 different meteorological years in order to assess energy system flexibility requirements. In (Wyrwa et al., 2022), the authors calculate REPI and other indicators to evaluate three main scenarios to design a pathway to carbon neutrality for the power system in Poland.

Share of Renewable Energy in Energy Mix The share of renewable energy in the energy mix was suggested among other by Du et al(Du et al., 2020). This value is often a political target, where the share of harvested renewable energy should exceed a given target.

$$\mathrm{RE}_{\mathrm{share}} = \frac{\sum_t \sum_\tau R_\tau(t)}{\sum_t P_{\mathrm{generated}}} \qquad \forall \tau \in \{\mathrm{RE}_{\mathrm{technologies}}\} \tag{4.2}$$

where τ are technologies, in this case restricted to renewable technologies, P is the total power generated, R is the power generated by an individual renewable technology.

RES Load Penetration Index (RLPI) The RES Load Penetration Index (RLPI) defines the maximum hourly coverage of load by VRES in relation to the total demand (Dagoumas, 2021; Michi et al., 2019; Poncela et al., 2018) as follows:

$$\mathrm{RLPI} = \max_t \left(\frac{\sum_t (W(t) + S(t))}{\sum_t L(t)} \right) \tag{4.3}$$

The RLPI index provides an indication of the amount of over capacities of VRES independent of the actual demand at a certain time period. RLPI also indicates the strain on the flexibility option of the system such as storage or curtailment of VRES in order to deal with the over capacity. The spatial distribution is not considered for this indicator as transmission adds another flexibility options, thus this indicator is one value per scenario.

Examples of RLPI calculations are shown in (ENTSO-E, 2014; Michi et al., 2019, 2018; Poncela et al., 2018), where RLPI is used together with REPI in order to assess VRES integration and power system flexibility requirements for various scenarios. RLPI values higher than 50% indicate higher VRES penetration. For RLPI values higher than 100% there is a risk of VRES curtailment (Blanco et al., 2016).

Geographical Dispersion of VRES Generation and Demand (GDVRES) Renewable power generation is highly dependent on geographical factors which results in VRES potential varying regionally and temporally, additionally are VRES not dispatchable, since they depend on the solar irradiation and the wind speed. The cumulative difference between the generation and the load gives an indication for this.(Taibi et al., 2018)

To assess the need for local flexibility options, three different indicators based on the geographical dispersion of VRES generation and demand are calculated. We call the indicators in the following $\mathrm{GDVRES_{st}}$ for spatial and temporal dispersion and $\mathrm{GDVRES_s}$ and $\mathrm{GDVRES_t}$ for spatial and temporal dispersion respectively. The total dispersion can be calculated as follows:

$$\mathrm{GDVRES_{st}} = \frac{\sum_{r,t} |W(t,r) + S(t,r) - L(t,r)|}{\sum_{t,r} L(r,t)} \qquad \text{with } r \in \{\text{regions}\} \tag{4.4}$$

where r indicates a certain region.

For the spatial and temporal dispersion, the temporal and spatial sum is taken as the absolute value. The spatial dispersion shows non-local production which needs transmission expansion while the temporal dispersion shows the need for storage as flexibility options.

$$\begin{aligned}
\mathrm{GDVRES_s} &= \tfrac{\sum_r \left|\sum_t (W(t,r)+S(t,r)-L(t,r))\right|}{\sum_{t,r} L(r,t)} && \text{with } r \in \{\text{regions}\} && (4.5)\\
\mathrm{GDVRES_t} &= \tfrac{\sum_t \left|\sum_s (W(t,r)+S(t,r)-L(t,r))\right|}{\sum_{t,r} L(r,t)} && \text{with } r \in \{\text{regions}\} && (4.6)\\
\mathrm{GDVRES_{max,t}} &= \max_t \tfrac{\sum_r |W(t,r)+S(t,r)-L(t,r)|}{\sum_r L(r,t)} && \text{with } r \in \{\text{regions}\} && (4.7)
\end{aligned}$$

Renewable Energy Curtailment Renewable energy curtailment describes the amount of VRES that had to be curtailed due to missing flexibility options such as storage or transmission capacity. Smaller values of VRES curtailments show a more efficient system which adequate flexibility options.

Curtailment of renewable energy has been used as an indicator for seizing battery storage systems (Yang, Bremner, Menictas, & Kay, 2018). An example is shown in Guo et al. (Guo, Wei, Chen, Dong, & Mei, 2020) where the authors used the renewable energy curtailment indicator to assess a two stage optimization model. The model was developed to determine the capacity of battery storage aimed at to reducing wind energy curtailment. Other examples are shown for both wind and solar energy curtailment indicators in (Bird, Cochran, & Wang, 2014; Rodrigues et al., 2015; Yang et al., 2018).

Renewable Energy Generation Curtailment Risk (RCR) The renewable energy curtailment risk describes the number of time-steps (amount of time) in a year during which VRES have to be curtailed (Poncela et al., 2018) and is defined as follows:

$$\mathrm{RCR} = \frac{\sum_t [\mathrm{RL} < 0]}{8760} \tag{4.8}$$

where RL indicates the residual load at hour t. The residual load is defined as follows:

$$\mathrm{RL(t)} = \mathrm{L(t)} - \mathrm{W(t)} - \mathrm{S(t)} \tag{4.9}$$

Smaller RCR values reflect an energy system where sufficient flexibility options are installed or the VRES coincides well with electricity demand. RCR values higher than zero represent also the number of hours per year where RLPI is greater than 1. This is reflected in negative residual load values, meaning that VRES generation is larger than the load ((Blanco et al., 2016)). Burke et al. (Burke & O'Malley, 2011) use RCR to evaluate the various factors influencing wind energy curtailment in wind farms. A general discussion of VRES curtailment, including its costs as well the extend of current curtailment in Germany and Denmark are presented by Jacobsen and Schröder (Jacobsen & Schröder, 2012).

4.3.2.2 Diversity Indicators

The technical diversity in the energy system is another category which impacts system robustness and resilience. Geographical and technological concentrations of individual technologies may result from the assumptions regarding the investment costs and resource potentials during least-cost optimization. This least-cost system however comes at the drawback of reduced security of supply in the case of different feed-in time-series such as shifts from solar to wind resources in individual weather years. This can be addressed by systematically analyzing the correlation between system diversity and security of supply indicators.

Herfindahl Index (HHI) The Herfindahl-Hirschman Index is a popular tool originating from economy used here to estimate the diversity of the power generation. An index close to one would imply that there is a strong concentration on one technology while an indicators close to zero means the power production is based on a wide portfolio.

$$H = \frac{\sum_\tau \sum_t P(t)_\tau^2}{\sum_\tau \sum_t P(t)_\tau} \qquad \forall \tau \in \text{generation technologies} \tag{4.10}$$

for all generation technologies as well as import of power. For the generation the actual generation based on the optimization model is taken, not the potential generation.

Shannon-Wiener-Index (SWI) The Shannon-Wiener diversity index is based on the concept of entropy. The larger the index is, the higher is the diversity. The application of this index to energy systems was proposed first by Stirling(Stirling, 1994).

$$H = -\sum_\tau p_\tau \cdot \ln p_\tau \qquad \forall \tau \in \text{generation technologies} \tag{4.11}$$

where p_τ is the proportionally reliance on the τ-th option to produce a certain power. A high value indicates that the system is not depending on a single energy source, thus is more robust towards technological development or change in the availability of a singular energy source.

Energy Self Sufficiency Rate (ESSR) This indicator describes the dependency of the energy system on power and fuel imports. The reference used for the definition of ESSR (Du et al., 2020) used an Integrated Assessment Model (IAM), not an ESOM. ESOMs often do not include fuel imports and thus would not be suitable for calculating this indicator.

$$\text{ESSR} = \sum_t \frac{E_{\text{import}} - E_{\text{export}}}{L(t)} \tag{4.12}$$

where E is transferred energy across borders both in chemical as well as electrical form. A high rate of self sufficiency means that the energy system is more stable towards external influences. The rate of self sufficiency is also often a criteria for policy makers.

4.3.3 Sustainability Indicators

The set of sustainability indicators aim to assess not only the global warming potential into account but also the impact on the ecosystems, the human health and resource depletion. This method aims to provide a more holistic approach as well as to show possible unexpected side-effects of the researched scenarios.

The sustainability indicators are based on common life cycle assessment indicators that based upon the analysis of Junne et. al.(Junne, Cao, Miskiw, Hottenroth, & Naegler, 2021), who looked into the power system expansion in Europe and Northern Africa with a multi-objective optimization. However, for this application, the life-cycle indicators where only assessed after the optimization to further investigate the system.

4.3.3.1 Climate Change: Global Warming Potential

The global warming potential is assessed and denoted as CO_2 equivalent (kg $CO_{2\text{eq}}$).

4.3.3.2 Ecosystem Quality

The following indications describe both direct adverse effects on the ecosystem or infusion of growth enhancing compounds that changes the balance in ecosystems.

Freshwater and terrestrial acidification (fw__terr__acid) The freshwater and terrestrial acidification gives a measurement of how high the acidification is which can adverse effects on the ecosystem quality and human health. The acidification is measured in mol H+_{eq}

Freshwater ecotoxicity (fw__ecotox) The freshwater ecotoxicity (in Comparative Toxic Unit for ecosystems) gives a measure of the impact of toxic substances on freshwater organisms. The toxic substances include metal ions as well as organic compounds.

Freshwater eutrophication (fw__eutr) The freshwater eutrophication gives a measure (kg P_{eq} – Phosphor) of the enrichment of the freshwater with nutritional elements which may lead to changes in the ecosystem balances.

Marine eutrophication (mar_eutr) The marine eutrophication measures (in kg N_{eq} – Nitrogen) the enrichment of marine waterbodies with nutritional elements which can lead to algae bloom and similar influences on the marine ecosystem.

Terrestrial eutrophication (ter_eutr) The terrestrial eutrophication (in mol N_{eq}) measures the enrichment of terrestrial ecosystems with nitrogen based compounds.

4.3.3.3 Human Health

These indicators describe adverse effects on human health.

Non-carcinogenic effects (HH_nc) This indicator aggregates non carcinogenic effects on human health (in Comparative Toxic Units for human health CTUh).

Carcinogenic effects (HH_c) This indicator aggregates carcinogenic effects on human health (in Comparative Toxic Units for human health CTUh).

Ionizing radiation (HH_ion) This indicator measures the damage on human health due to the emission of radio-nucleids and the resulting radiation (in kg U^{235}_{eq}).

Ozone layer depletion (HH_od) An indication of the atmospheric Ozone layer depletion leading to higher terrestrial UV radiation(in kg CFC-11_{eq}).

Photochemical ozone creation (HH_poc) The photo-chemical ozone creation gives an indication of ozone in the lower layer of the atmosphere (also known as smog) due to catalysis with sunlight (in kg $NMVOC_{eq}$).

Respiratory effects, inorganics (HH_resp) The effects on the human respiratory system (in disease incidences) indicate further toxic emissions especially particulate matter.

4.3.3.4 Resources

Fossils including uranium (RES_fossils) The fossils sum up the excavated fossil energy as well as uranium (in MJ) and show the depletion of those fossil energy carriers.

Land use (RES_land_use) The land use (in points) show the land usage by the energy generation plants which may conflict with other land usages such as agriculture.

Minerals and metals (RES_minerals_metals) The depletion of minerals and metals (in kg Sb_{eq}) indicate the total resource usage over all metals and minerals.

Dissipated water (RES_water) The dissipated water (in m^3 $water_{eq}$) indicates the water requirements during production and operation in the energy system.

4.3.3.5 Emission

The emission indicators address on the one hand the political targets on an basis of absolute emission while on the other hand making systems comparable with respect to their environmental impact. Having a higher emission allowance per energy demand typically correlates to lower system cost if no carbon costs and constraints are taken into account. As a direct result this reduces the electricity generation from renewable capacities and having more reserve capacities in the form of conventional power plants. This however causes the systems to be not directly comparable anymore under the premise of political targets towards carbon emissions.

Carbon Dioxide Emissions per Unit of Energy Consumption This indicator, for which a lower value is deemed to be better, gives an indication of the climate impact of the power system as well as its adaption of renewables.(Du et al., 2020)

Total Carbon Dioxide Emissions The total amount of carbon emission generated by the power system. A lower value is better. This value also depends on the degree of electrification in the context of REMix, since REMix is not a technology complete IAM that also includes all production and transportation sectors. For the analysis, the total carbon emissions were divided by the total power demand of the highest scenario, to ensure a better performance of the machine learning.

4.3.4 Security of Supply and Flexibility Indicators

In future energy systems with high shares of renewable energies, the security of supply will strongly depend on the available flexibilities. For this reason, a great deal of attention is paid to this when selecting suitable security of supply indicators.

Insufficient Ramping Resource Expectation (IRRE) The IRRE indicator indicates how high the probability is that too low ramping capacity is available in the energy system to absorb the ramps that occur in the residual load (Lannoye, Flynn, & O'Malley, 2012). For this purpose, available flexibility distributions are formed from the time series of the flexible power plants and storage facilities and the ramps that occur are classified in them. A distinction is made between the lack of ramping down and ramping up capacities.

Periods of Flexibility Deficiency (PFD) Number of times when the flexibility in the energy system is not sufficient to cover ramps in the residual load. Based on the time series for available flexibility and the residual load. This indicator can also be divided into up and down. (Papalexopoulos, Hansen, Frowd, Tuohy, & Lannoye, 2016)

Expected Flexibility Shortfall (EFS) Level of lack of flexibility in the system that results in a certain probability of occurrence. Since the IRRE only indicates how high the probability of unfulfilled ramps is, the EFS was developed by Abdin and Zio (Abdin & Zio, 2018) to also quantify the extent of these unfulfilled ramps in terms of energy quantities. For this indicator also available flexibility distributions are necessary like for the IRRE. The EFS is also differentiated into up and down flexibility.

Reserve Inadequacy Times at which the reserve power for the energy system falls below a specified level. Without sufficient reserve power, the energy system cannot react to unforeseen events such as power plant outages or similar. This indicator was also used by IRENA in (Taibi et al., 2018).

Grid-related Curtailment of Renewables Curtailment of renewables due to grid congestion. Indicates the ratio of curtailment to potential generation of renewables due to grid congestion. Is intended to make a statement about the available grid and how adequately it is developed in relation to the spatial distribution of renewables.

Energy Not Served (ENS) Energy that was demanded but could not be served. REMix outputs this indicator via the slack variable, which supplies energy at a very high cost when there is no other cheaper way to generate electricity at these times. This indicator shows when and to what extent such an extreme threat to the energy system is imminent.

Share of Time when Transmission is Not Congested Proportion of times when no congestion occurs in the grid of the energy system under consideration (Taibi et al., 2018). This indicator provides information on how adequately the transmission capacities are dimensioned.

Normalized Flexibility Index (NFI) This indicator shows the flexibility for the energy system through the installed flexible power plants (Ma, Silva, Belhomme, Kirschen, & Ochoa, 2012). The flexibility indicator for each type of power plant is calculated primarily via the ramping speed in relation to the maximum flexible capacity that can be used by these power plants.

Loss of Wind Estimation (LOWE) In its original form, the LOWE index gives an estimated probability of wind energy being curtailed due to a lack of flexibility (Ma et al., 2012). For our purposes, we have extended it to all renewable energy variables used in the scenarios. This also allows us to estimate whether the power system would be able to cope with even more installed renewables without having to curtail large amounts of it.

Non-Synchronous Penetration Ratio (SNSP) The SNSP is an indicator of the network stability. A high value shows a lower inertness of the system, which can lead to higher fluctuation of the voltage in short time frames. For the calculation, the PV and wind production of every hour and the import and export according to Poncela et al.(Poncela et al., 2018)

$$\mathrm{SNSP}(t) = \frac{R_{\mathrm{wind}}(t) + R_{\mathrm{PV}}(t) + P_{\mathrm{import}}}{L(t) + P_{\mathrm{export}}} \tag{4.13}$$

where R is the renewable energy production, P is the import/exported power, and L is the demand, each at time t. This indicator is restricted by technical constraints, i.e. it should not get too large.

Power Capacity Reserve Margin $P_{\mathbf{CRM}}$ The power capacity reserve margin as described in (Kisel, Hamburg, Härm, Leppiman, & Ots, 2016) aims at quantifying the flexible reserve capacities in relationship to the peak load and is defined as follows:

$$P_{\text{CRM}} = \frac{P_{\text{inst}} + P_{\text{imp}} - P_{\text{restricted}} - P_{\text{main}} - P_{\text{out}} - P_{\text{SR}} - P_{\text{export}}}{\max_t L(t)} \tag{4.14}$$

where:

P_{inst} = installed power capacity of the system
P_{imp} = power capacity that can be imported
$P_{\text{restricted}}$ = power capacity with restricted availability
P_{main} = power capacities under scheduled maintenance
P_{out} = power capacities under unplanned outage or maintenance
P_{SR} = system reserve of the TSO
P_{export} = capacities pre-booked with binding export agreements
$L(t)$ = Load at time t

P_{CRM} is an indicator of the systems ability to cover peak load even in non-optimal conditions and is the main indicator of Power Generation Adequacy (Kisel et al., 2016). Power capacities with restricted availability include among others intermittent power, permanently not available power, and capacities down to environmental restrictions.

4.3.5 Definition of the Methodology and Delimitation of the Literature Review

The goal of this review is to identify appropriate indicators that can be used to assess the security of energy supply. Specifically, we seek to determine common indicators used to evaluate energy supply security and the models and algorithms used to calculate these indicators. We also seek to identify the scenarios where these indicators have been applied and if it is possible to calculate them using REMix.

To achieve our goal, we conducted a systematic search for appropriate literature. We searched for publications from as early as 1990, but with emphasis on those from 2012 to 2020. We focused on publications that use real data from the past, energy system models using electricity grids, and models without power grids at all or with simplified artificial ones.

Our search was based on key words including power system modeling, energy system modeling, security of supply, flexibility, power system, energy system, energy security, and indicators. We used search engines such as Google Scholar as well as referenced literature to locate relevant studies. We searched for research written in English.

Our results demonstrate that there are common indicators used to evaluate energy supply security: demand, diversity, flexibility, market, and production. Models and algorithms vary greatly in complexity from simplified models to models using real networks. These indicators are used as input data and must be feasible to integrate into REMix to calculate supply security.

4.3.5.1 Overview of all indicators

In table 4.10 all calculated indicators are listed and the ones used for the POI analysis are emphasized.

Table 4.10: Overview of the indicators. The key indicators for the assessment of the energy systems are in italics.

Power market	**Overall System**	**Sustainability**	**Energy Security**
Market-driven total Carbon Emissions	Shannon-Wiener-Index (SWI)	Carbon Dioxide Emissions per Unit of Energy Consumption	Insufficient Ramping Resource Expectation (IRRE)
Supply-based Indicators	Herfindahl Index (HHI)	*Freshwater and terrestrial acidification (fw_terr_acid)*	Periods of Flexibility Deficiency (PFD)
Missing Peak Production	*Share of Renewable Energy in Energy Mix*	Freshwater eutrophication (fw_eutr)	*Expected Flexibility Shortfall (EFS)*
Average Electricity Price	Renewable Energy Generation Curtailment Risk (RCR)	Marine eutrophication (mar_eutr)	*Reserve Inadequacy*
Hours with maximum Prices	Renewable Energy Curtailment	*Terrestrial eutrophication (ter_eutr)*	Energy Not Served (ENS)
Profitability by technology	*Geographical Dispersion of VRES Generation and Demand (GDVRES)*	Non-carcinogenic effects (HH_nc)	*Share of Time when Transmission is Not Congested*
Electricity Price Volatilities	RES Load Penetration Index (RLPI)	*Carcinogenic effects (HH_c)*	Normalized Flexibility Index (NFI)
Hours with Negative Prices	System costs	Ionizing radiation (HH_ion)	Loss of Wind Estimation (LOWE)
Policy instrument costs	Annual demand	*Ozone layer depletion (HH_od)*	*Grid-related Curtailment of Renewables*
Share of economic curtailment	Renewable Energy Penetration Index (REPI)	Photochemical ozone creation (HH_poc)	Non-Synchronous Penetration Ratio (SNSP)
	Energy Self Sufficiency Rate (ESSR)	Respiratory effects, inorganics (HH_resp)	Power Capacity Reserve Margin P_{CRM}
Market-driven energy not served		Fossils including uranium (RES_fossils)	
		Land use (RES_land_use)	
Dispatch-related system costs		Dissipated water (RES_water)	
		Global Warming Potential	
		Total Carbon Dioxide Emissions	
		Minerals and metals (RES_minerals_metals)	
		Freshwater ecotoxicity (fw_ecotox)	

4.4 Implementation of the models

4.4.1 Implementation of the Indicator Models

4.4.1.1 Workflow of the Indicator Models

One goal of the workflow is the automatic calculation of different indicators to assess the solved scenarios. On the basis of these indicators a POI analysis takes place, which is used to investigate some of the scenarios in more depth. Figure 4.1 shows how this process looks like in a schematic way.

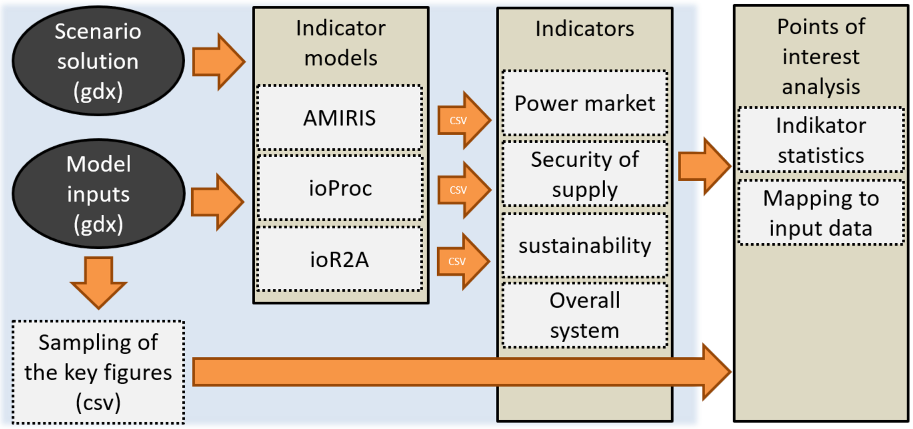


Figure 4.1: Workflow of the Indicator Models

First of all, the solved scenarios results and if also necessary data scenarios enter the indicator models as GDX-files. For example, to calculate the power market indicators, the result data must first be pre-processed. They are then used as input data in the AMIRIS model. This model simulates the behavior of the various electricity market players and uses the results to calculate the power market indicators described above.

The indicator models for calculating the security of supply indicators, the sustainability and the overall system indicators are also fed with the REMix results. The Python scripts, which are based on the workflow manager ioproc, then calculate the indicators for supply security, flexibility and diversity of the scenarios.

All indicator models write all indicators used in CSV-files. These are statistically processed for the POI analysis and mapped to the scenario input data. The goal is to find the scenarios which are of high interest for further manual analysis by an automated process.

4.4.1.2 Documentation of the Calculation of the Security of Supply Indicators

Here, we will provide an illustrative explanation of the computation of the “Reserve inadequacy” indicator. We will cover its implementation in Python code and also list the necessary REMix result parameters.

The code calculates the Reserve Inadequacy Indicator, which assesses the insufficiency of reserve capacity in a power system. It takes into account various data such as system load, VRES produc-

required data	REMix symbol	description
flexibility production	commodity_balance	hourly profile of flexible power plant generation
system load	commodity_balance	hourly profile of demand
VRES production	commodity_balance	hourly profile of VRES generation
storage level	storage_level_out	hourly profile of storage level
storage output	commodity_balance	hourly profile of storage output
converter capacities	converter_caps	installed capacities of converters (power plants and storage output)

Table 4.12: Required REMix result symbols for reserve inadequacy indicator

tion, flexibility production, storage level, storage load, storage output and converter capacities. In table 4.12 the required REMix symbols are associated with the previously mentioned data.

The code first preprocesses the input data and calculates the maximum load and minimum reserve capacities for upward and downward flexibility. It then calculates net load ramps, upwards load ramps, and downwards load ramps based on the system load and VRE production.

Next, the code calculates the available flexibility for both storage and converter technologies. It considers factors such as storage capacity, storage level, production, and load to determine the available flexibility in both upward and downward directions.

Afterwards, the code calculates the overall flexibility by combining the flexibility values from storage and converter technologies. It aggregates the available flexibility values for each time interval.

Finally, the reserve inadequacy is computed by comparing the available flexibility with the minimum reserve capacities. The code calculates the inadequacy in both upward and downward directions, and then normalizes the values by dividing them by the product of the minimum reserve capacity and the total number of time intervals (8760). Additionally, the code determines the maximum reserve inadequacy values for both upward and downward directions.

The reserve inadequacy indicator provides insights into the capacity shortfall of reserves in the power system and helps identify potential issues with meeting the system's demand and maintaining grid stability.

4.4.2 Design and Implementation of Model Interfaces

All three models have been designed to be used in a fully automated pipeline on High-Performance-Clusters. The interfaces are therefore the inputs received by the step before, and the outputs that are then used by the next model in the pipeline. For the outputs, we chose a simple format, CSV, that can be recognized by all other models.

4.4.3 Conclusion and Proposals for Further Development

The wide scope and the large number of the indicators analyzed have been positively received at several conferences, since they allow to get a rather complete picture of a future energy system.

4.5 Results of the Models and Analysis of the POI

Indicators Each model produces a number of indicators as described above. During iterations, new indicators, e.g. system cost or generation capacities of each technology (REMix) have been added. Furthermore, life-cycle-assessment indicators have been added to get an even more comprehensive picture. These indicators are coupled with input parameters for the analyses, in total 55. For later analyses, k-means and POI-analysis, a subset of 27 of the most important POI is selected. For some indicators, the interpretation is clear, e.g., system costs or CO_2 emissions - the lower the better. However, for some indicators this is less clear. Therefore, the indicators are evaluated in relation to the overall mean value of all scenarios. If it is above or below one standard deviation in the desired direction (if possible), it is considered for further investigation. Scenarios that score high on many of these indicators are selected and designated as POI, standing for points of interest. These data (indicators and inputs) are extracted from the runs via R scripts. First, a script collects all path names via a database on the HPC-environment. Second, all GDX-parameters are collected. Third, all indicators are collected. Fourth, GDX-parameters and indicators are combined. Fifth, subsets are produced based on the thresholds defined for POI. A POI scenario is one that accumulates more than 15 of POI below or above average of 18 indicators that are analyzed in more detail. They are then combined to a large data frame with information added which run and which model they originate from. This data frame is then used to determine the POI. For that, various thresholds are used in order to find POI. Another statistical way employed was to calculate k-means and k-medoids, i.e. clustering techniques to determine related parameters.

Clustering First, both indicators and inputs are clustered. For later runs, only indicators were considered, resulting in a smaller cluster number across multiple runs. For robustness, both k-means and k-medoids are used. K-medoids is better suited to deal with outliers, since it is more robust. However, both methods almost always produced the same number of clusters. To determine the optimal number of clusters, manual ways were impractical due to the nature of the automated workflow. Hence, two methods were implemented that can programmatically determine the best number. These were the BIC (Bayesian information criterion) method and gap statistics, respectively. Second, we used different settings for running the HPC-workflow and to observe a statistically meaningful number of indicator and input sets. Therefore, the used probability distributions for the parameter sampling with the scenario generator (see section 3.2.1) were varied and different REMix instances and thus, different spatial resolutions of ESOM instances have been examined. Across these several runs the number of clusters varied between 1-9. For example, 1000 scenarios with pre-existing power plant capacities and low spatial resolution, this does not result in detectable clusters. Therefore, we evaluate a smaller number of high resolution and green field ESOMs. In both the latter cases, we observe three clearly delineated clusters (nine indicators are collapsed to two dimensions using principal component analysis (PCA), i.e., main correlations. With no inputs considered, cluster numbers center around 3-5. The correlations between the indicators are as expected, e.g. a high share of RE corresponds to low CO_2 emissions, etc. This gives credibility to the scenario analyses.

POI results All figures are for medium-sized model instances in terms of spatial resolution. Clustering the 3600 uniformly distributed parameters results in three clusters (see figure 4.2).

Clustering the runs with 24 instead of 7 weather years results in 7 clusters, which is the maximum number of clusters across runs (see figure 4.3)

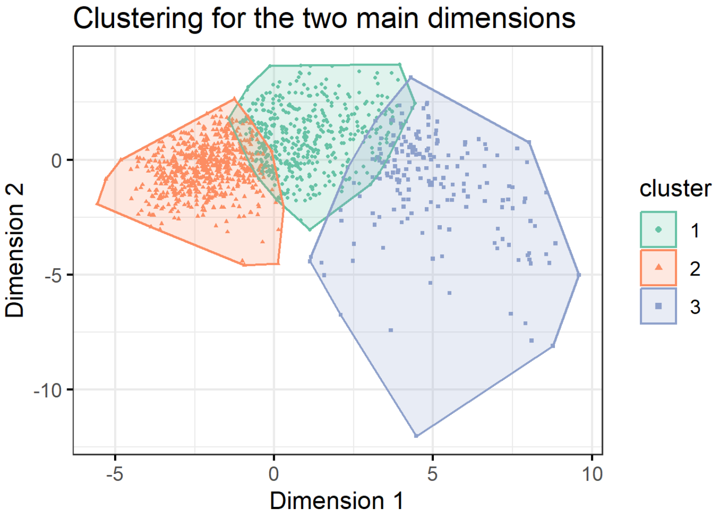


Figure 4.2: Two main dimensions of clusters in 3600 runs, uniform distribution

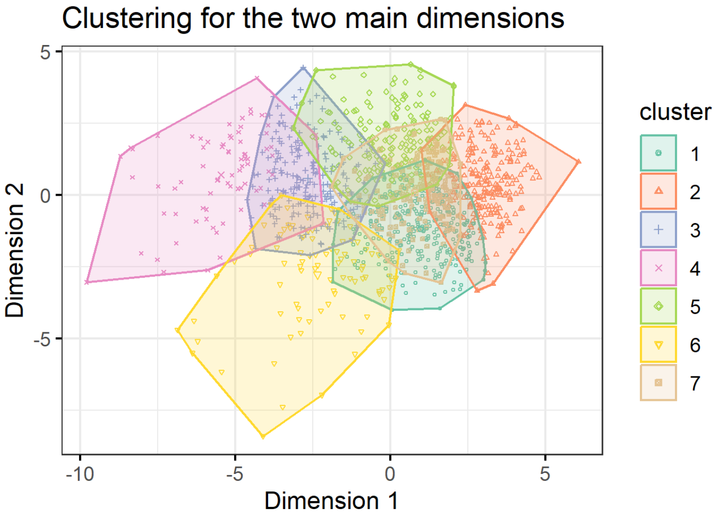


Figure 4.3: Two main dimensions of clusters for 1300 runs with 24 weather years

Perhaps the most robust number of clusters is for the truncated normal distribution for 2000 LP runs (see figure 4.4)

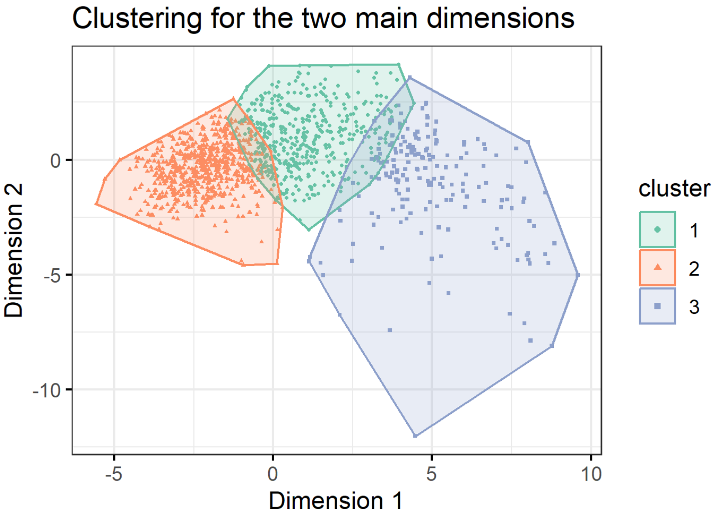


Figure 4.4: Two main dimensions of clusters for 2000 LP runs, truncated normal distribution

4.5.1 Summary and Analysis of the Results: Evaluation of the Indicators and Scenarios

An overall evaluation of the scenarios must begin with the percentages of technologies that are built by REMix. The following three figures show the results for the 2000 runs with truncated normal distribution, 3600 runs with uniform distribution, and 1300 runs with an extended option to randomly select VRES times-series based on a historical weather year[2] for the clusters above. All share the same spatial medium-size resolution (Edelpraut instances, see Table 4.1). Figure 4.5 (2000 runs with truncated normal distribution) shows a relatively high proportion of PV.

These runs (3600 runs with uniform distribution) show the highest share of PV percentage across scenarios (see figure 4.6).

More weather years (1300 runs) seem to have a dampening effect on the PV-share (figure 4.7)

Evaluation Overall, POI scenarios share a number of attributes that merit a more detailed breakdown. For example, they have a higher RE-share (84 % vs. 64 %). Consequently, they have more PV (97 % more), wind offshore (23 % more), and 6 % more wind onshore. In line with these results, such systems emit 44 % less CO_2-emissions, but are 17 % more expensive (total system cost). The mean electricity price is higher (296 Euro compared to 194 Euro). These differences are all statistically significant (t-tests, two-tailed, $p < 0.05$).

4.6 Enhancements of REMix

While the consideration of the complete analysis chain of the previous chapters requires significant efforts in computational resources and access to adequate hardware in the form of high performance

[2] one out of 24 instead of one out of six for the other evaluations

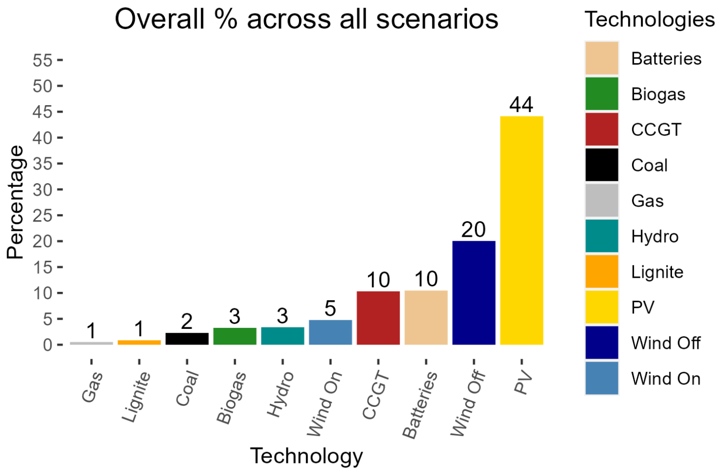


Figure 4.5: Technology distribution for 2000 LP runs, truncated normal distribution

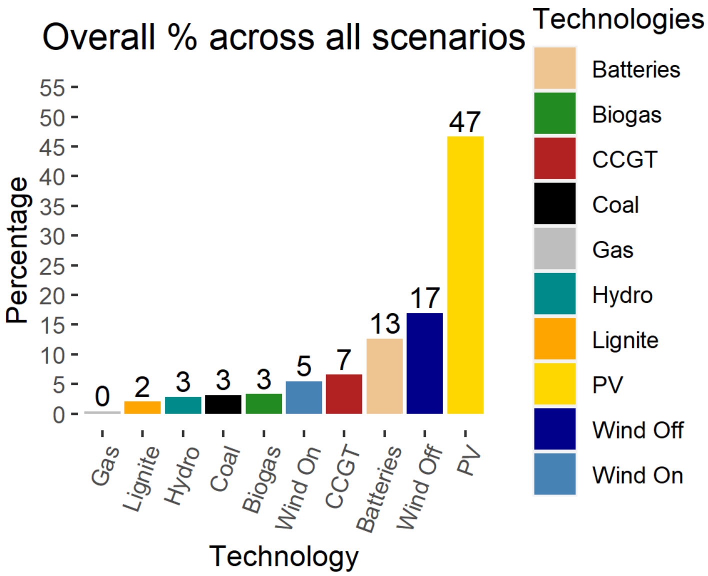


Figure 4.6: Technology distribution for 3600 LP runs, uniform distribution

computers, similar methods for finding a wide range of model results can be directly integrated into the REMix modeling framework. To this end, five distinct aspects are addressed in the methodological enhancement of REMix: Discrete investment decisions, unit commitment constraints, stochastic programming, near-optimal solutions needs, and visualization for comparing the obtained solutions to support making informed decisions. Therefore, the following sections describe the efforts of the UNSEEN project to improve the REMix modeling framework in the mentioned directions and discuss exemplary findings based on exemplary test cases.

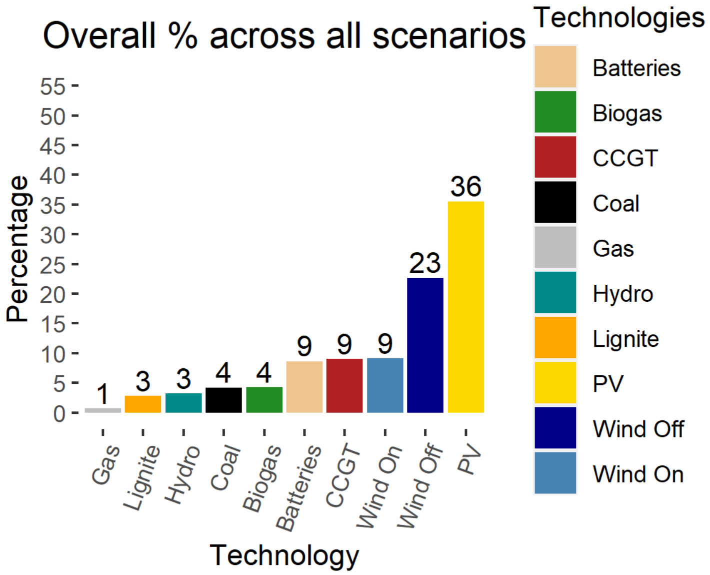


Figure 4.7: Technology distribution for 1300 LP runs, 24 weather years

4.6.1 Implementation of discrete investment decisions

For a subset of indicators (mainly relevant if spatially highly resolved regions are examined) it is advisable to provide results from solved scenarios that are based on discrete investment decisions. Therefore, in a first stage of MIP implementation, the linear programming problem in REMix is extended by a more realistic representation of actual investment decisions. To receive meaningful results with the indicator calculation ESOMs with high technological detail and a high spatial resolution are necessary. However, for testing purposes on local machines the calculations during development were carried out for model instances with low spatial resolution (e.g., Benemipt instances, see table 4.1). That means we aggregated the basic REMix model instance to work with ESOMs that consider only a few regions.

Discrete investment decisions can be activated in REMix by setting the parameter "mipUnits^{i,y}" to one for the according technologies i and vintage years y. In this way, a constraint is activated that fixes the continuous variable of the total capacity $\text{Units}_{\text{Total}}^{n,i,y}$ to an integer variable $\text{Units}_{\text{Total,MIP}}^{n,i,y}$ for all modeled regions n:

$$\text{Units}_{\text{Total}}^{n,i,y} = \text{Units}_{\text{Total,MIP}}^{n,i,y} \tag{4.15}$$

Furthermore, the continuous variables for the number of online units in each time step t

$$\text{Units}_{\text{Online}}^{t,n,i,y} = \text{Units}_{\text{Total}}^{n,i,y} \tag{4.16}$$

is also set equal to an integer variable:

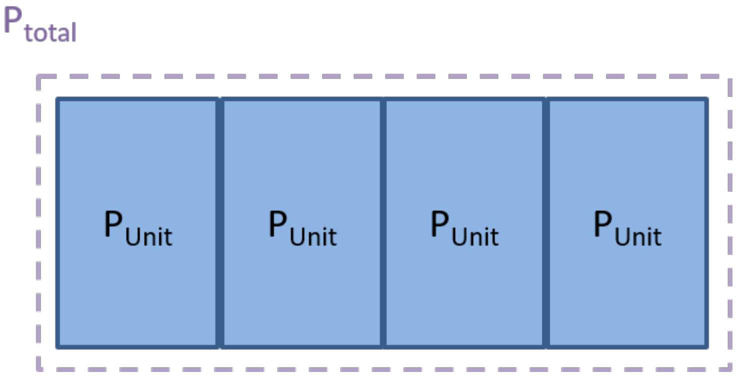


Figure 4.8: Discrete investment decisions. The total power of a technology consists of a discrete number of units, each with a technology-specific block size.

$$\text{Units}_{\text{Online}}^{t,n,i,y} = \text{Units}_{\text{Online,MIP}}^{t,n,i,y} \tag{4.17}$$

The block size of each technology can be adjusted with the coefficient parameter which indicates how much of each commodity is consumed and produced per unit and activity mode.

4.6.2 Implementation of unit commitment constraints

In a second stage of MIP implementation, REMix has been extended by modeling unit commitment decisions. The goal is to achieve a more detailed and technically precise representation of power plant operation in the energy system. Unit commitment features can be activated by setting the parameter "mipDispatch^{i,y}" to one. This allows the start-up and shut-down of units since the number of online units can be less than or equal to the number of totally available units:

$$\text{Units}_{\text{Online}}^{t,n,i,y} \leq \text{Units}_{\text{Total}}^{n,i,y} \tag{4.18}$$

Partial load The partial load feature allows us to model a minimum load and a maximum load per activity mode m. For all the units using a specific activity mode $\text{Units}_{\text{UsingActivityMIP}}^{t,n,i,y,m}$ the activity has to be less than or equal to the upper limit

$$Activity^{t,n,i,y,m} \leq maxLoad^{t,n,i,y,m} Units_{\text{UsingActivityMIP}}^{t,n,i,y,m} \tag{4.19}$$

and greater than or equal to the lower limit

$$Activity^{t,n,i,y,m} \geq minLoad^{t,n,i,y,m} Units_{\text{UsingActivityMIP}}^{t,n,i,y,m}. \tag{4.20}$$

An additional constraint

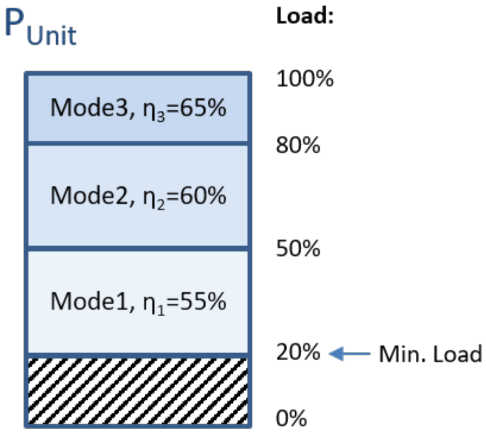


Figure 4.9: Partial load behavior of unit commitment. The units of a technology can be shut down or started up while a minimum load has to be reached by each online unit. Depending on the partial load level different operational modes can be activated with varying efficiencies.

$$Units^{t,n,i,y}_{\text{Online}} = \sum_m Units^{t,n,i,y,m}_{\text{UsingActivityMIP}} \tag{4.21}$$

ensures that each online unit uses exactly one activity mode.

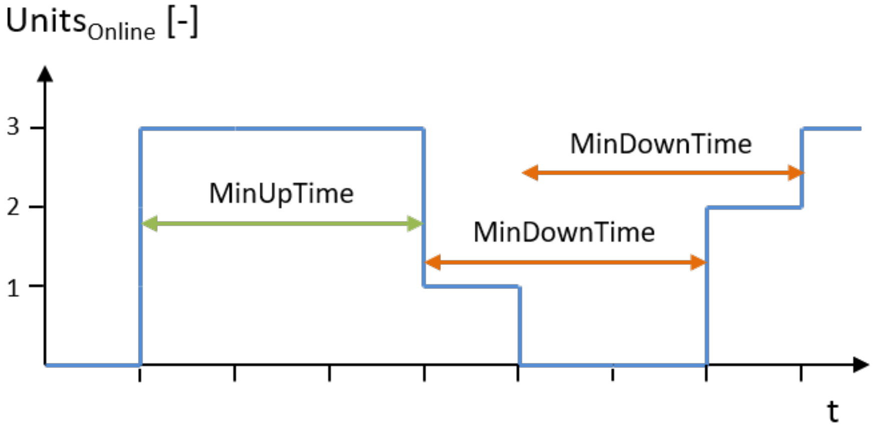


Figure 4.10: Minimum up- and downtime behavior of unit commitment. The units can only be shut down and started up in a way that complies with the minimum up-time and downtime restrictions of each technology.

Minimum up- and downtime Another unit commitment feature that is implemented in REMix is the minimum up- and downtime. Units can only be started up if they have been offline for a sufficient time and shut down if they meet the minimum up-time for being online.

Therefore, the number of startups in each time step

$$Unit^{t,n,i,y}_{\text{Startups}} \geq Units^{t,n,i,y}_{\text{Online}} - Units^{t-1,n,i,y}_{\text{Online}} \tag{4.22}$$

as well as the number of shutdowns in each time step

$$Unit^{t,n,i,y}_{\text{Shutdowns}} \geq Units^{t-1,n,i,y}_{\text{Online}} - Units^{t,n,i,y}_{\text{Online}} \tag{4.23}$$

have to be tracked. By indicating that $Unit^{t,n,i,y}_{\text{Startups}}$ and $Unit^{t,n,i,y}_{\text{Shutdowns}}$ are positive integer variables the number of startups and shutdowns is always greater than or equal to zero.

The parameters "$MinUpTime^{i,y}$" and "$MinDownTime^{i,y}$" can be set per technology and vintage. They activate the minimum uptime constraint

$$\sum_{x=t-MinUpTime^{i,y}+1}^{t} Unit^{x,n,i,y}_{\text{Startups}} \leq Units^{t,n,i,y}_{\text{Online}} \tag{4.24}$$

and the minimum downtime constraint

$$\sum_{x=t-MinDownTime^{i,y}+1}^{t} Unit^{x,n,i,y}_{\text{Shutdowns}} \leq Units^{n,i,y}_{\text{Total}} - Units^{t,n,i,y}_{\text{Online}}. \tag{4.25}$$

A list of all the used symbols can be found in Table 4.13.

4.6.3 Addressing parameter uncertainty via stochastic programming

4.6.3.1 From deterministic optimization to stochastic optimization

Stochastic optimization enables the analysis of uncertainties within a single model run by considering different data scenarios with their respective probabilities. Thus, a hedging strategy against the possible risks can be realized. The uncertainties that we consider for our analysis encompass weather, demand, and parameter uncertainties. The demand is divided into the normalized demand profile and the annual demand, enabling independent adjustments for each. For the techno-economic parameter uncertainties fuel and investment costs are considered. For the parameter values the parameter space for the Scenario Generator can be used here as well (see Section 3.1.2). REMix considers input parameters in a deterministic manner. In order to consider parameter uncertainties the Extended Mathematical Programming (EMP) feature of GAMS is used which works as an add-on to the deterministic model and which transforms it into a stochastic model (GAMS, n.d.).

4.6.3.2 Extending the GAMS EMP framework for PIPS-IPM++

During the project two different phases for the extension of the EMP framework were conceptualized and conducted. This enabled us to see the practical viability of the EMP framework in conjunction with PIPS-IPM++ and, in a second stage, to improve usability and reporting of the results.

Sets	
i	index for the technology
m	index for the activity mode
n	index for the region
t	index for the time
x	alias for the time index
y	index for the year
Variables	
$Activity^{t,n,i,y,m}$	Power generation per time step and activity mode (continuous)
$Units^{t,n,i,y}_{\text{Online}}$	Online units (continuous)
$Units^{t,n,i,y}_{\text{Online,MIP}}$	Online units (integer)
$Units^{n,i,y}_{\text{Total}}$	Total units (continuous)
$Units^{n,i,y}_{\text{Total,MIP}}$	Total units (integer)
$Units^{t,n,i,y,m}_{\text{UsingActivityMIP}}$	Number of units using activity mode (integer)
Parameters	
$maxLoad^{t,n,i,y,m}$	Maximum load per activity mode
$MinDownTime^{i,y}$	Minimum downtime
$minLoad^{t,n,i,y,m}$	Minimum load per activity mode
$MinUpTime^{i,y}$	Minimum up-time
$mipDispatch^{i,y}$	Activation of unit commitment
$mipUnits^{i,y}$	Activation of discrete investment decisions
$Unit^{t,n,i,y}_{\text{Shutdowns}}$	Number of shutdowns per time step
$Unit^{t,n,i,y}_{\text{Startups}}$	Number of startups per time step

Table 4.13: Nomenclature

Phase 1 - proof of concept As of GAMS Release 39 (released on May 03, 2022), the Deterministic Equivalent (DE) solver comes with a new hidden option *pipsstages* (GAMS, 2022). *pipsstages* can take 3 values

- 0 (default): DE works as usual and does not set the .scale variable/equation attribute
- 1: DE ignores existing .stage variable/equation attributes. For 2-stage Stochastic Programs that optimize the expected value, DE creates .stage attributes for variables and equations when generating the deterministic equivalent. First stage variables (and equations containing only first stage variables) get assigned $.stage = 1$, 2nd stage variables that belong to scenario s (and equations containing only first stage variables and variables of scenario s) get assigned $.stage = 1 + s$. The equation that defines the objective variable gets assigned $.stage = S + 2$ where S is the number of scenarios.
- 2: DE uses existing .stage variable/equation attributes. Assuming that the current .stage annotation defines a PIPS-suitable block structure it is combined with the scenario induced block structure such that the deterministic equivalent of a 2-stage SP where the underlying model has N PIPS Stages and the 2nd stage has S scenarios results in up to $1+N*S$ (variable) stages in total. DE ensures that we get consecutive stages and have no “holes” in the stage annotation. While in most cases we will end up with $1+N*S$ variable stages, it is theoretically

possible to get less variable stages. The final number of stages has to be retrieved from the GDX File (or we may add some additional DE output for it in the future). The resulting deterministic equivalent is hence generated with a PIPS-suitable .stage annotation.

Assuming all blocks of the annotation are numbered consecutively, the following rules (1 and 2) apply for variables, where

- de_j describes the variable index in the deterministic equivalent
- j describes the variable index in the underlying model
- s describes the emp stage of variable j
- n describes the scenario index $s = 1, 2, ..., S$ of de_j
- $pips_{\text{stage}}$ describes the .stage attribute of j in the underlying model
- de_{stage} describes the .stage attribute of variable de_j in the resulting deterministic equivalent

if $s = 1$ or $pips_{\text{stage}} = 1 \rightarrow de_{\text{stage}} = 1$
if $s > 1$ and $pips_{\text{stage}}{>}1 \rightarrow de_{\text{stage}} = 1 + s * pips_{\text{stage}}$

For equations, the following rules (1 to 3) apply, where

- de_i describes the equation index in the deterministic equivalent
- i describes the equation index in the underlying model
- s describes the emp stage of equation i
- n describes the scenario index $s = 1, 2, ..., S$ of de_i
- $pips_{\text{stage}}$ describes the .stage attribute of i in the underlying model
- de_{stage} describes the .stage attribute of equation de_i in the resulting deterministic equivalent

if $pips_{\text{stage}} = N + 2$ (linking constraint in underlying model) $\rightarrow de_{\text{stage}} = N * S + 2$ if $s = 1$ or $pips_{\text{stage}} = 1 \rightarrow de_{\text{stage}} = 1$ if $s > 1$ and $pips_{\text{stage}} > 1 \rightarrow de_{\text{stage}} = 1 + s * pips_{\text{stage}}$

In summary, the required steps to create a GDX file containing the deterministic equivalent of a 2-stage SP in a format suitable for gmspips are the following:

1. define the emp info file as described in the documentation.
2. specify DE as solver for model type EMP.
3. provide a DE solver option file containing options *pipsstages* 1/2 subsolver convert subsolveropt <number>
4. provide a convert solver option file containing options dumpgdx <filename> gdxnames 0

The resulting GDX file with name <filename> will contain the deterministic equivalent with the PIPS-suitable .stage annotation. The usual workflow to split the gdx via gmschk and solve the model via gmspips can be applied (except for the solution reporting).

Phase 2 - improving performance and usability While the first development phase provided a technical basis for demonstrating the extension of EMP towards propagating the annotation of the underlying model, there were still some shortcomings, which have been addressed in a second phase. The following outlines corresponding problems and approaches implemented:

DE was designed in a way that random variables (defined in the emp.info file) are translated into fixed variables when generating the deterministic equivalent. Those fixed variables are irrelevant for the deterministic equivalent of linear models. However, they still appear in the GDX file with .stage=1 and jacobian records in matrix A. With that said, if the number of random variables is large, this can result in large number of fixed variables which makes the splitting (via gmschk) and also the file interaction and presolving when running gmspips expensive. As a workaround, those fixed variables (and any other fixed variables) can be squeezed out of the model by running squeezeJacobian.gms with *–jacfile=myModel.gdx* which will transform *myModel.gdx* into *myModel_squeezed.gdx*

The whole concept of EMP-SP relies on synchronous solves and a user defined dictionary with output-handling information that allows to map the solution of a deterministic equivalent to GAMS data structures. However, bringing this EMP concept in line with the intentionally asynchronous GAMS/PIPS workflow is not trivial because:

- The dictionary required by the gmspips option printsol to generate **_sol.gdx* solution file does not exist for the deterministic equivalent. This is also the reason why convert has to be used with option gdxnames=0.
- The workaround of squeezing fixed variables out of the model via squeezeJacobian.gms adds an additional layer of complexity.

4.6.3.3 Computational performance of PIPS-IPM++ for stochastic optimization

Solver	Decomposition by	PV investment cost	Demand profile
CPLEX	-	0:47 h	1:20 h
PIPS (0)	time	0:18 h	0:28 h
PIPS (1)	scenarios	2:17 h	3:57 h
PIPS (2)	time and scenarios	0:30 h	0:43 h

Table 4.14: Performance tests for PV investment cost and demand profile uncertainties. A decomposition by time in PIPS leads to time savings of up to 65%.

Two test cases are analyzed to compare the performance of PIPS-IPM++with different decomposition strategies. In the first test case PV investment cost uncertainties and in the second test case demand profile uncertainties are considered. In both cases a model instance with 19 regions and 10 stochastic scenarios is analyzed. The results of the performance tests are shown in Table 4.14. When using the CPLEX solver for the deterministic equivalent created by EMP the solving times are 0:47 h and 1:20 h, respectively. If PIPS-IPM++is combined with EMP for solving the stochastic problems, a decomposition by time, stochastic scenario or both can be chosen by setting the option *pipsstages* accordingly. While a decomposition solely by scenarios leads to an increase in calculation times, a decomposition by time reduces the calculation time. However, a variation in the size of the model instance and the number of considered stochastic scenarios can lead to other preferable *pipsstages* settings. Furthermore, other settings to PIPS-IPM++can be subject to further fine-tuning. This will be analyzed in a prospective paper.

4.6.3.4 Case study on stochastic optimization

The implementation of stochastic optimization with EMP in REMix was used to analyze capacity adequacy in energy system modeling (Sasanpour & Cao, 2022). Different uncertainties are considered and their impact on the structure of a decarbonized German power system is examined.

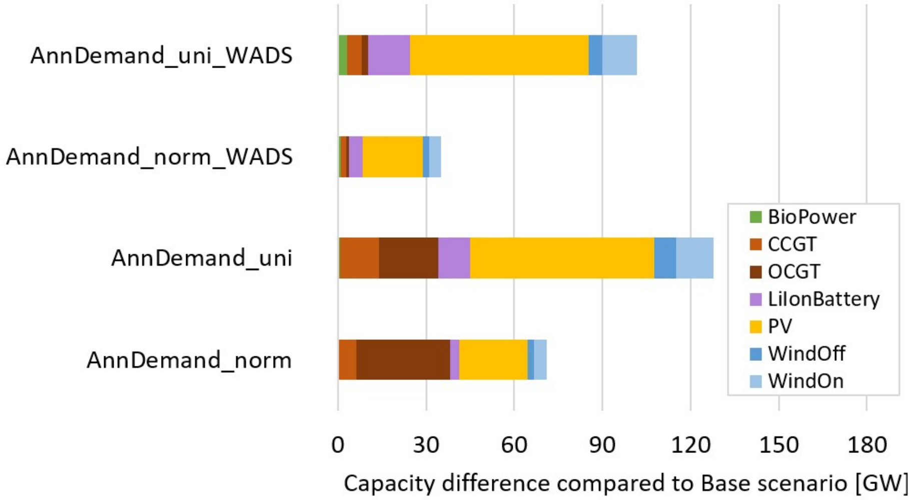


Figure 4.11: Capacity difference of stochastic optimization with annual demand uncertainty compared to deterministic Base case.

Figure 4.11 shows some exemplary results of the capacity difference if annual demand uncertainties are considered. With a truncated normal distribution (AnnDemand_norm) about 70 GW of additional capacities are expanded compared to the deterministic Base case. If a uniform probability distribution is used (AnnDemand_uni) significantly more capacities are exploited to achieve capacity adequacy. Additonally, more CCGT (combined-cyle gas turbine) instead of OCGT (open-cyle gas turbine) are expanded which, on the one hand, have higher capital expenditures. On the other hand, their operational expenditures are lower, which is advantageous due to the higher probability of extremely high annual demands with a uniform distribution. In the cases where the weighted average of the deterministic scenarios (WADS) is used, similar amounts of additional renewable power plants are expanded. However, significantly less gas power plants are available. Therefore, the stochastic optimization results in the availability of more secured capacities and the hedging against risk is higher.

Further information on the model setup, the uncertainties and the detailed results of the analysis can be found in (Sasanpour & Cao, 2022).

4.6.4 Finding near-optimal solutions and quantification of trade-offs

In contrast to dealing with parameter uncertainty via stochastic programming, there are uncertainties associated with the solution space of deterministic models. In general, ESOMs are characterized by a very flat optimum, which allows for quite different system designs when different optimization criteria are considered (Trutnevyte, 2016). These near-optimal solutions can provide a more detailed view of possible trade-offs that can be made during the overall energy system design. During the project, two different methods for generating trade-offs have been implemented into the REMix framework: First, two-dimensional Pareto-optimal frontiers and second, linear approaches from the field of MGA. For both approaches, the relevant literature was reviewed and an appropriate approach was implemented in the REMix framework to generate example data based on selected instances from the analysis presented in section 4.1.

4.6.4.1 Pareto-optimal frontiers

Pareto-optimal solutions are defined as multi-criteria solutions where one indicator can not be improved without reducing the performance of another indicator. Sampling a wide range of pareto-optimal points in an n-dimensional solution space allows the generation of pareto-optimal frontiers. There are various methods for generating these frontiers, with the ϵ-constraint method being one of the more widely used options. The method has been further improved by Mavrotas (Mavrotas, 2009) for efficient sampling in n-dimensional spaces.

For the implementation in REMix, we follow the standard implementation of a two-stage approach. First, the optimum in the first dimension to be optimized is evaluated (e.g. least cost solution) and the second dimension (e.g. carbon emissions) is kept as a reference value. For the second optimization stage, the first indicator is added as an additional constraint with a predefined additional tolerance of x

Mathematically, the generation of the two-dimensional pareto-optimal frontier can be formulated as follows:

Stage 1: cost minimization

$$
\begin{array}{lll}
\text{minimize} & I_{\text{syscost}} & \\
\text{subject to} & I_{\text{syscost}} & = \sum\limits_{y,r,p} c_{\text{invest},y,r,p} \cdot n_{\text{build},y,r,p} \\
 & & + \sum\limits_{y,r,p,t,a} c_{\text{var},p,a} \cdot act_{y,r,p,a,t} \\
 & & + \sum\limits_{y,r,r',p,t} c_{\text{flow},r,r',p} \cdot flow_{y,r,r',p,c,t} \\
 & & + \sum\limits_{y,r,p,t} c_{\text{import},p,c} \cdot import_{y,r,p,c,t} \\
 & P_{y,r,p,c} & = p_{\text{rated},c} \cdot n_{\text{total},y,r,p} \\
 & n_{\text{total},y,r,p} & = n_{\text{total},y-1,r,p} + n_{\text{build},y,r,p} - n_{\text{decom},y,r,p} \\
 & act_{y,r,p,a,t} & \leq n_{\text{total},y,r,p} \\
 & level_{y,r,p,c,t} & \leq P_{y,r,p,c} \\
 & flow_{y,r,r',p,c,t} & \leq P_{y,r,p,c} \\
 & 0 & = act_{y,r,p,a,t} \cdot coef_{p,a,c} \\
 & & + level_{y,r,p,c,t-1} - level_{y,r,p,c,t} \\
 & & + flow_{y,r',r,p,c,t} - flow_{y,r,r',p,c,t} \\
 & & + import_{y,r,p,c,t} - demand_{y,r,p,c,t}
\end{array}
$$

Stage 2: CO_2 minimization

$$\begin{array}{ll} \text{minimize} & I_{\mathrm{CO_2,direct}} \\ \text{subject to} & I_{\mathrm{CO_2,direct}} = \sum\limits_{y,r,p,a,t} i_{\mathrm{CO_2,direct},p,a} \cdot act_{y,r,p,a,t} \\ & I_{\mathrm{syscost}} \quad \leq I'_{\mathrm{syscost,ref}} \cdot \{1.01 \ldots 1.10\} \end{array}$$

Figure 4.12 shows exemplary results generated with a 120 region model instance comparable to the Edelpraut model configuration. By minimizing carbon emissions under different levels of allowed incremental system cost (shown in the top figure), a wide range of different capacity expansion decisions can be observed (shown in the bottom figure).

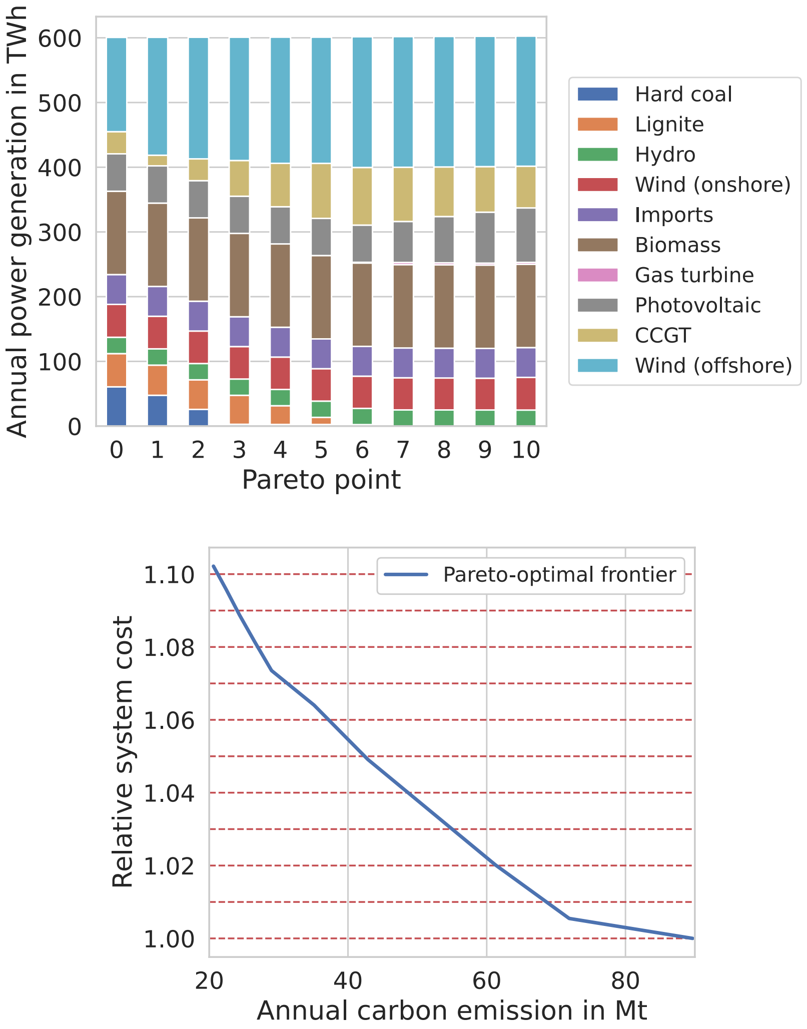


Figure 4.12: Overview of different results generated with a pareto-optimal frontier

4.6.4.2 Modeling-to-generate-alternatives (MGA)

MGA takes a similar approach to exploring the near-optimal solution space as pareto-optimal frontiers. However, the aim is to explore a wide range of different system designs. One of the first applications in the field of energy system modeling was presented by DeCarolis (DeCarolis, 2011). Several other studies have used similar approaches from the field of MGA to derive a wide range of possible systems, such as a follow-up study by DeCarolis et al. (DeCarolis, Babaee, Li, & Kanungo, 2016) and studies by Berntsen and Trutnevyte (Berntsen & Trutnevyte, 2017) and Price and Keppo (Price & Keppo, 2017). Similarly, the method has also allowed the robustness of results to be increased by systematically exploring the sensitivities of individual technologies, as presented by Neumann and Brown (Neumann & Brown, 2021).

While MGA describes a more general methodological approach, all individual studies use individual formulations to achieve this diversification effect. For the implementation in REMix, two key factors in the design of the MGA approach have been considered i) keeping the model formulation linear in order to allow the solution of large-scale models and ii) using a deterministic method that allows the simultaneous computation of multiple MGA points. Analogous to pareto-optimal frontiers, the approach follows a two-step process. In the first stage, the least-cost system is solved to provide a reference for the minimum system cost, and in the second stage, the distance metric of different combinations of indicator vectors is maximized while allowing a predefined increase in the total system cost. Mathematically, the approach can be formulated as follows:

Stage 1: cost minimization

$$
\begin{array}{lll}
\text{minimize} & I_{\text{syscost}} & \\
\text{subject to} & I_{\text{syscost}} & = \sum\limits_{y,r,p} c_{\text{invest},y,r,p} \cdot n_{\text{build},y,r,p} \\
 & & + \sum\limits_{y,r,p,t,a} c_{\text{var},p,a} \cdot act_{y,r,p,a,t} \\
 & & + \sum\limits_{y,r,r',p,t} c_{\text{flow},r,r',p} \cdot flow_{y,r,r',p,c,t} \\
 & & + \sum\limits_{y,r,p,t} c_{\text{import},p,c} \cdot import_{y,r,p,c,t} \\
 & P_{y,r,p,c} & = p_{\text{rated},c} \cdot n_{\text{total},y,r,p} \\
 & n_{\text{total},y,r,p} & = n_{\text{total},y-1,r,p} + n_{\text{build},y,r,p} - n_{\text{decom},y,r,p} \\
 & act_{y,r,p,a,t} & \leq n_{\text{total},y,r,p} \\
 & level_{y,r,p,c,t} & \leq P_{y,r,p,c} \\
 & flow_{y,r,r',p,c,t} & \leq P_{y,r,p,c} \\
 & 0 & = act_{y,r,p,a,t} \cdot coef_{p,a,c} \\
 & & + level_{y,r,p,c,t-1} - level_{y,r,p,c,t} \\
 & & + flow_{y,r',r,p,c,t} - flow_{y,r,r',p,c,t} \\
 & & + import_{y,r,p,c,t} - demand_{y,r,p,c,t}
\end{array}
$$

Stage 2: VRE diversification

$$
\begin{array}{lll}
\text{maximize} & dist_{\text{mga}} & \\
\text{subject to} & dist_{\text{mga}} = (I_{\text{wind}} - I'_{\text{wind,ref}}) \cdot v_{\text{wind},i}) & \\
& \qquad +(I_{\text{PV}} - I'_{\text{PV,ref}}) \cdot v_{\text{PV},i}) & i = 1, \dots, n \\
& & \\
& I_{\text{wind}} \;\; = \sum\limits_{y,r,p,a,t} i_{\text{wind},p,a} \cdot act_{y,r,p,a,t} & \\
& I_{\text{PV}} \quad = \sum\limits_{y,r,p,a,t} i_{\text{PV},p,a} \cdot act_{y,r,p,a,t} & \\
& I_{\text{syscost}} \leq I'_{\text{syscost,ref}} \cdot 1.05 &
\end{array}
$$

The target vector $v_{\text{indicator},i}$ is computed based on an n-dimensional hypersphere, so that all combinations of minimizations and maximizations are taken into account. For the two-dimensional example presented, this gives a set of vectors equally distributed radially from the reference point, based on the cost-optimal solution from the first stage.

Figure 4.13 shows the results of an example case study. Diversification was based on minimizing and maximizing system-wide solar PV and onshore wind capacity (see top figure). The resulting ranges in terms of annual energy production are shown in the lower figure. All systems were within 5% of the minimum achievable total system cost.

A list of the symbols used for the description of pareto-optimal frontiers and MGA can be found in Table 4.16.

4.6.5 Visualization of REMix results using the GAMS MIRO WebUI

GAMS MIRO is a deployment environment which allows to turn GAMS models into fully interactive applications. An intuitive, user-friendly interface allows to interact with the underlying GAMS model, create scenarios, compare results and more. MIRO's extensive data visualization capabilities allow to create charts, time series, maps, widgets, etc. A gallery of example visualizations created using the GAMS MIRO engine for a highly simplified energy system model created during the UNSEEN project can be found at `https://miro.gams.com/gallery/app_direct/simple_mip/`.

In order to offer similar capabilities for large-scale energy system models with an improved representation of different technologies a GAMS MIRO prototype has been developed for the REMix model. The prototype allows to interact with the REMix model through a GAMS MIRO application, for example, modify selected input parameter and view the results of a run in different charts and tables. This prototype also included both methods for the generation of pareto-optimal frontiers and computation of a wide range of different solutions using MGA. Figure 4.14 shows an example visualization for different MGA solutions using a stacked bar plot for annual energy generation per technology and a radar plot for the ratio between annual energy prodcuced via onshore and offshore wind energy and solar PV.

The prototype will be made available as open source alongside the publication of the REMix framework at `https://gitlab.com/dlr-ve/esy/remix/framework/-/tags/gams-miro` as soon as the necessary internal steps for release to the public are done. While it will allow others to get insights on how to provide a GAMS MIRO engine for similar ESOMs, long-term compatibility with the main version of the REMix framework cannot be guaranteed.

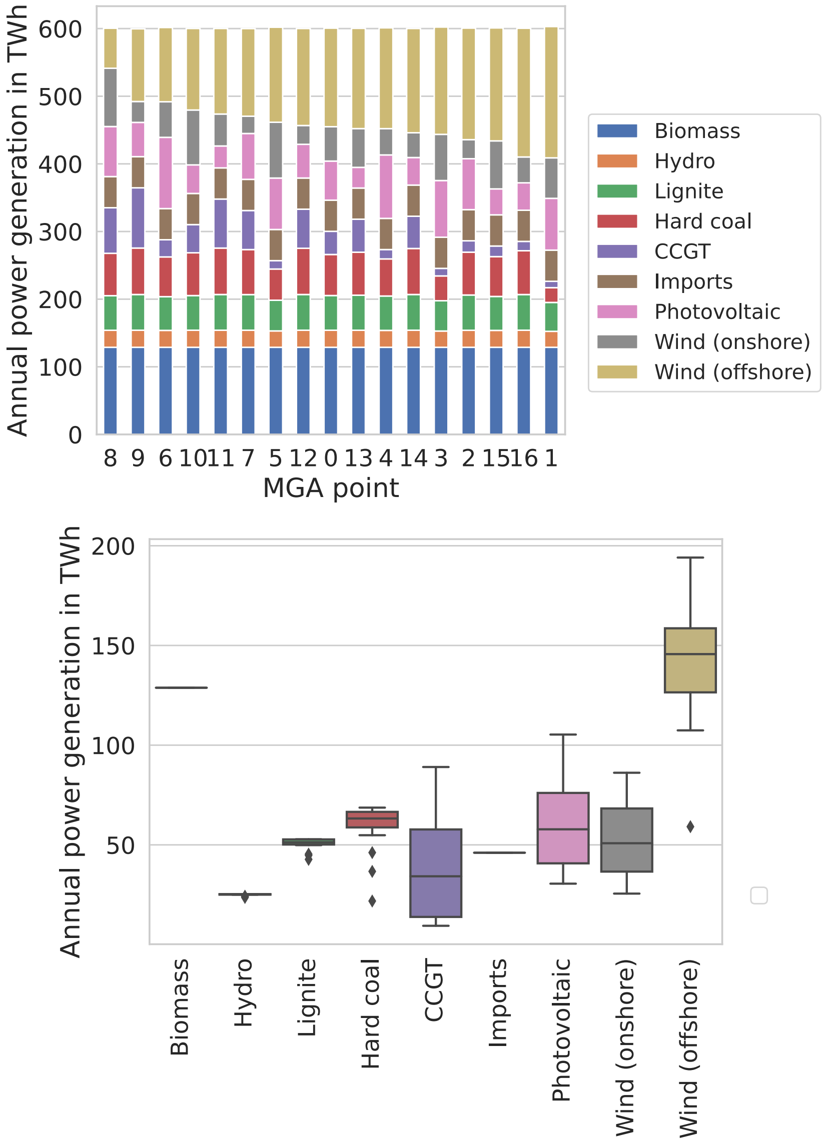


Figure 4.13: Overview of differ555nt results generated with a MGA methodology based on wind and photovoltaics to diversify the results

Sets	
a	Index for the activity mode
c	Index for commodity
p	Index for the technology
r	Index for the region
r'	Alias for the region index
t	Index for the time step
y	Index for the year
Variables	
$act_{y,r,p,a,t}$	Activity level of activity mode a
$flow_{y,r,r',p,c,t}$	Commodity flow from region r to region r'
$I_{\mathrm{CO_2,direct}}$	Indicator for amount of CO_2 emission
I_{PV}	Indicator for annual PV electricity generation
$I'_{\mathrm{PV,ref}}$	Indicator for annual PV electricity generation from least-cost optimization which is used as reference for current optimization
I_{syscost}	Indicator for the total system cost
$I'_{\mathrm{syscost,ref}}$	Indicator for the total system cost from least-cost optimization which is used as reference for current optimization
I_{wind}	Indicator for annual wind electricity generation
$I'_{\mathrm{wind,ref}}$	Indicator for annual wind electricity generation from least-cost optimization which is used as reference for current optimization
$import_{y,r,p,c,t}$	Commodity import
$level_{y,r,p,c,t}$	Storage level
$n_{\mathrm{build},y,r,p}$	Technology units build in year y
$n_{\mathrm{decom},y,r,p}$	Technology units decomissioned in year y
$n_{\mathrm{total},y,r,p}$	Total technology units available in year y
$n_{\mathrm{total},y-1,r,p}$	Total technology units available in previous year $y-1$
$P_{y,r,p,c}$	Technology capacity
Parameters	
$c_{\mathrm{flow},r,r',p}$	Specific cost for commodity transfer from region r to region r'
$c_{\mathrm{import},p,c}$	Specific importing cost
$c_{\mathrm{invest},y,r,p}$	Specific investment cost per technology unit with considered annuity
$c_{\mathrm{var},p,a}$	Specific variable cost per technology activity
$coef_{p,a,c}$	Coefficient indicating the generated or consumed commodity per activity mode
$demand_{y,r,p,c,t}$	Commodity demand
$dist_{\mathrm{mga}}$	Distance metric
$i_{\mathrm{CO_2,direct},p,a}$	Specific CO_2 emission per technology p and activity mode a
$i_{\mathrm{PV},p,a}$	Specific indicator for annual PV electricity generation per activity mode a
$i_{\mathrm{wind},p,a}$	Specific indicator for annual wind electricity generation per activity mode a
$p_{\mathrm{rated},c}$	Specific capacity per technology unit
$v_{\mathrm{PV},i}$	Target vector for annual PV electricity generation
$v_{\mathrm{wind},i}$	Target vector for annual wind electricity generation

Table 4.16: Nomenclature

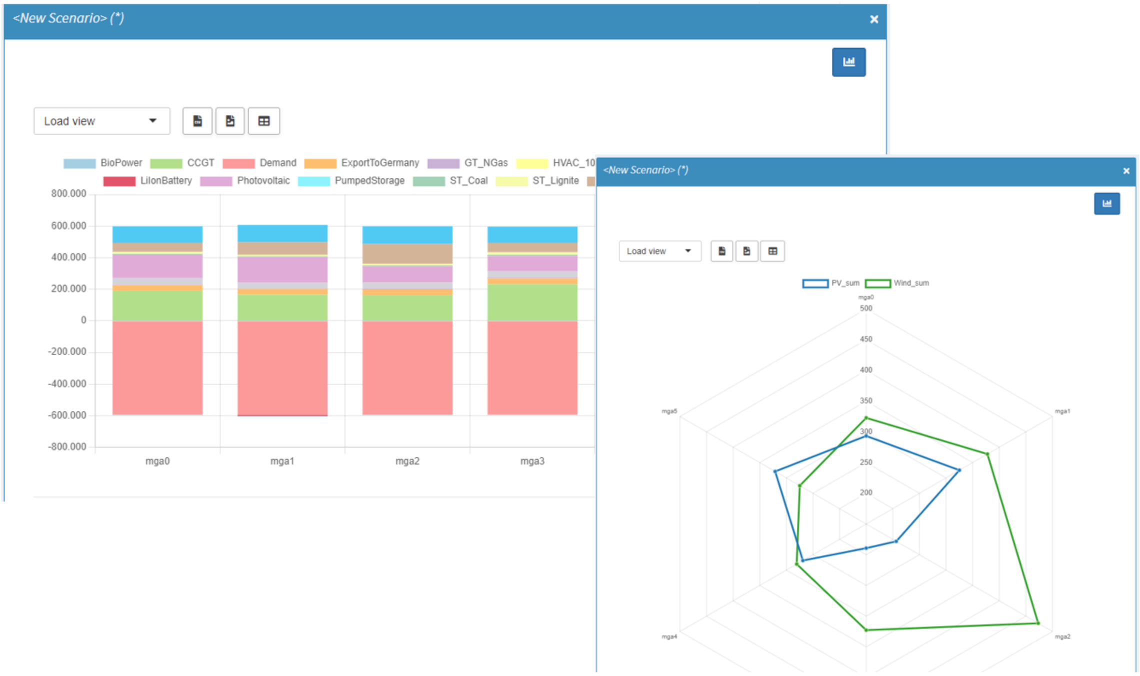


Figure 4.14: Example visualization using the MIRO REMix prototype for the MGA methodology based on the diversification of annual energy feed-in from wind and photovoltaics

Chapter 5

PIPS-IPM++: a distributed interior-point method for solving block-structured LPs

Within the project, we implemented and designed PIPS-IPM++, a parallel interior-point solver for large-scale block-structured LPs. The solver is freely available on GitHub.

5.1 PIPS-IPM++: a distributed, massively parallel, structure-preserving and highly scalable LP solver

The solver PIPS-IPM++, developed and released during the project, implements a massively parallel interior-point method designed for the parallel solution of block-structured linear programs. Mathematically, a salient characteristic of these LPs is (after suitable permutation) their arrowhead structure:

$$
\begin{array}{lllllll}
\min & \sum_{i=0}^{N} c_i^T x_i & & & & & \\
\text{s.t.} & T_0 x_0 & & & & & = h_0 \\
 & T_1 x_0 + W_1 x_1 & & & & & = h_1 \\
 & T_2 x_0 + & W_2 x_2 & & & & = h_2 \\
 & \vdots & & \ddots & & & \vdots \\
 & T_N x_0 + & & & W_N x_N & & = h_N \\
 & U_0 x_0 + U_1 x_1 & + \; U_2 x_2 & \cdots & U_N x_N & & = h_{N+1} \\
 & x_0, \quad x_1, & x_2, & \cdots & x_N & & \geq 0
\end{array}
$$

The *linking variables* are represented by the vector x_0, and the *linking constraints* by the matrices $U_0, ..., U_N$ and the vector h_{N+1}. PIPS-IPM++ exploits this block structure and applies a parallel

Schur complement decomposition to efficiently solve the LP in parallel. The main algorithm of PIPS-IPM++ is described in (Rehfeldt, 2021). In the following, we describe two major extensions that were implemented and developed during the project. First, we will report on the implementation of the so-called hierarchical solution approach in PIPS-IPM++ – a nested, fully parallel and distributed Schur complement decomposition strategy. Second, we describe the current extensions in PIPS-IPM++ specialized and parallel presolve routines.

One of the main ingredients in state-of-the-art mathematical optimization solvers is presolving. Presolve aims at reducing the size and complexity of the mathematical model, while improving its numerical properties. In practice, solver benefit massively from presolve routines. It is usually applied before the actual solution algorithm and can as such be treated as a separate part of the solution pipeline. Presolve improves the model by applying a set of predefined invertible mathematical reformulations to it; this should improve the overall performance of the solver, while not changing the solution of the problem. The result of the presolving step is the *reduced model* which is then solved more easily by the solver. To recover an original solution from the solution of the reduced model, a step known as postsolve is performed after solving the reduced model. Postsolve reverts the changes made to the original model during postsolve, while also transforming the computed solution of the reduced model. This yields a primal-dual solution of the original problem.

Since presolve is usually orders of magnitude cheaper to perform than the actual solution algorithm, much energy has been invested into finding a complete and efficient set of presolve routines that should be applied to a given model. While (Achterberg, Bixby, Gu, Rothberg, & Weninger, 2019) describes the impact and importance of presolving in a state-of-the art commercial optimization solver, further analysis and an overview over most LP presolve can be found in (Andersen & Andersen, 1995; Gamrath, Koch, Martin, Miltenberger, & Weninger, 2015; Gemander et al., 2020; Gondzio, 1997).

The difficulty when applying these presolve routines within PIPS-IPM++ lies in the solver's strong dependency on the problem structure. Possible presolve transformations must not destroy the carefully annotated block-structure given with the model, so that the PIPS-IPM++ efficiency is preserved. For this reason, we implemented a set of presolve routines within PIPS-IPM++ that are both structure preserving and distributed parallel (therefore highly efficient). Equally important and intricate to implement are postsolving routines that recover original solutions. For every implemented presolve routine, PIPS-IPM++ features an "inverse" postsolve routine also implemented in distributed parallel. Generally, the postsolving in PIPS-IPM++ is extremely cheap compared to presolve and solve. Currently, the PIPS-IPM++ presolve (and postsolve) suite consists of the following reduction algorithms:

- Singleton row elimination
- Singleton column elimination
- Bound tightening
- Parallel and nearly parallel row detection
- Removal of empty rows
- Dual fixing
- Redundant row detection

- Detection of fixed variables

In practice, presolve is often the deciding factor within PIPS-IPM++ whether or not an optimization problem can be solved. While presolve significantly speeds up the optimization process, the interior-point method also benefits from the increased numerical stability of the reduced model. For a more detailed explanation of our presolve routines, we refer the reader to (Gleixner, Kempke, Koch, Rehfeldt, & Uslu, 2020).

We implemented another transformation within PIPS-IPM++ that is not meant to reduce the problem size, but should improve numerical aspects of the problem: the distributed parallel transformation of inequalities into equalities. A linear inequality constraint is given by $a^T x \leq b$, with $a \in \mathbb{R}^n, b \in \mathbb{R}$ and $x \in \mathbb{R}^n$. The modeler may wish to model a real world problem as an LP with numerous (possibly redundant) inequalites, which leads to a rank deficient constraint matrix and numerical difficulties in the interior-point method. To partially address this problem, we transform all inequalites into equalities by adding so-called slack variables: $a^T x + s = b$, where $s \geq 0$ is an additional slack variable. While this increases the problem size, it also guarantees a full rank inequality constraint matrix, which often outweighs the inconvenience of having a slightly larger problem. The above transformation is non-standard within PIPS-IPM++ and can be activated in the `PIPSIPMpp.opt` file (see section 5.4) with the option:

```
PRESOLVE_TRANSFROM_INEQUALITIES_INTO_EQUALITIES true bool
```

5.2 The Hierarchical solution approach

The main idea of the hierarchical solution approach is to overcome current difficulties in the size of the Schur complements as described in (Rehfeldt, 2021). The approach mitigates this bottleneck by further splitting and distributing the Schur complement among the processes – as opposed to the *classical* PIPS-IPM++ approach where one dedicated process processes the whole Schur complement.

5.3 Other improvements and changes in PIPS-IPM++

Small correctors algorithm Factorizing linear systems (an $O(n^3)$ operation) is the bottleneck in PIPS-IPM++; for large problems, it is much more expensive than solving the system once the factorization is available (an $O(n^2)$ operation). Moreover, the MPI communication necessary to successfully complete the factorization is costly. Thus, higher-order path following methods or higher-order Gondzio correctors are well worthwhile. However in practice, these higher-order correctors often cannot guarantee enough progress, therefore we choose not to compute them. Blocking directions at the current iterate may be converged too far along the central path, which prevents large Newon steps to be taken.

While the original PIPS-IPM++ employs Gondzio's multiple corrector algorithm (Colombo & Gondzio, 2008), the current version of PIPS-IPM++ improved upon this by adding a so-called small corrector step to the problem. We thus add additional Gondzio correctors to the step computation; they specifically focus on lifting small slack-dual pairs on the central path back to the current target on the central path. This is simply the application of further Gondzio steps where the Gondzio projection of the current point does not target a box around the current target but a halfspace: projecting all slack-dual pairs to zero if they have not yet reached the target and only focusing on pairs that are too far converged, thus too small. In practice, this seems helpful, in particular when applying primal-dual steps, and usually results in a 5% decrease of iterations compared to the normal corrector scheme.

Interfaces to linear solvers Currently, PIPS-IPM++ has working interfaces to WSMP, MA27, MA57, MUMPS, PARDISO, and MKL Pardiso. For normal application of PIPS-IPM++, we currently recommend PARDISO from Pardiso-Project as the fastest option. For the application of the hierarchical solution approach, we urge the user to download and get a license for MA57 as it offers the best parallelization potential within our solver.

5.4 Configuring the solver

With the GitHub release of PIPS-IPM++, we implemented a mechanism to pass runtime parameters via an option file. If the file **PIPSIPMpp.opt** lies in the same directory as the PIPS-IPM++ executable, it will be read at runtime and will be used to adjust certain aspects of our solver. The general appearence of a line within this file is

```
NAME VALUE TYPE
```

where `NAME` is the name of the setting, `VALUE` its (allowed) value, and `TYPE` one of `bool`, `int`, `double`.

Chapter 6

Performance analysis

In the field of high-performance computing, the performance of applications running in parallel plays a major role, as the aim is to ensure that the applications make optimal use of the hardware. Thus, a comparison was made between four commercial state-of-the-art solvers and the solver PIPS-IPM++ developed in the project. In addition, PIPS-IPM++ was tested in various configurations as part of a sweet spot analysis in order to develop a guideline as to which division of the data is recommended and how the ratio of computing time requirement to program runtime behaves when various parameters are taken into account.

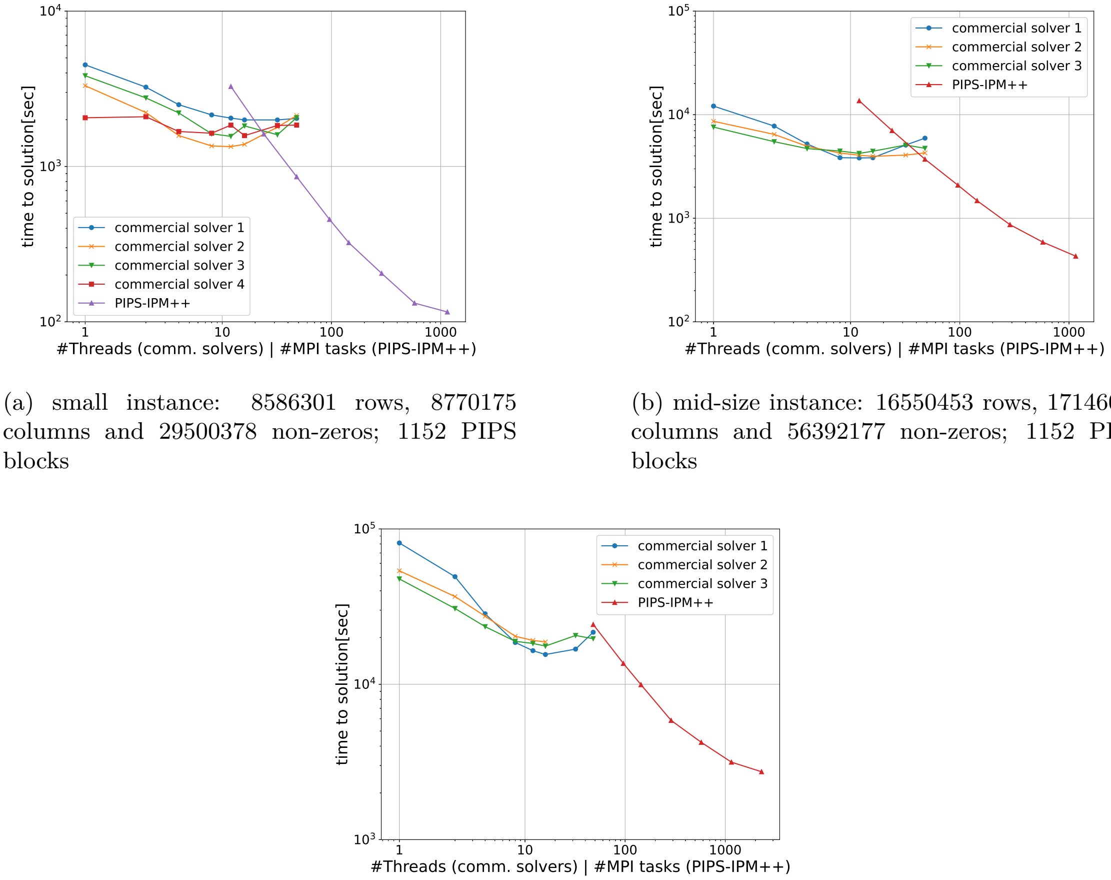
6.1 PIPS-IPM++ vs commercial solvers

The performance comparison of PIPS-IPM++ with four commercial solvers was carried out on the JUWELS (Jülich Supercomputing Centre, 2021) supercomputer at the Jülich Supercomputing Centre. A small, a medium and a large scenario were considered in order to exclude the possibility that the results only apply to a specific problem size. In addition, PIPS-IPM++ was started in hierarchical mode (see chapter 5.2) and up to a maximum of 2304 tasks with 4 threads per task were distributed over 192 computing nodes. Since the commercial solvers are not implemented for distributed memory systems, they can only be executed on a single shared memory compute node and are thus limited in execution to the 48 cores of a JUWELS node.

In the figures 6.1 the number of processes is varied to compare the scaling behaviour of PIPS-IPM++ with commercial solvers based on the required program runtime. **For all problem sizes tested, both the scaling, i.e. the performance gain when doubling the number of processes used, and the runtime are significantly better for PIPS-IPM++ than for all commercial solvers tested.**

6.2 Sweetspot analysis

When running PIPS-IPM++ on a supercomputer, there are several configuration options. On the hardware side, the number of computing nodes, Message Passing Interface (MPI)-tasks and threads per task can be varied, and on the software side, PIPS-IPM++ can be started in different modes



(a) small instance: 8586301 rows, 8770175 columns and 29500378 non-zeros; 1152 PIPS blocks

(b) mid-size instance: 16550453 rows, 17146016 columns and 56392177 non-zeros; 1152 PIPS blocks

(c) large instance: 38857052 rows, 40635185 columns and 130896035 non-zeros; 2304 PIPS blocks

Figure 6.1: Performance comparison: PIPS-IPM++ (hierarchical mode) vs. commercial solvers on JUWELS hosted at the Juelich Supercomputing Centre

nodes	tasks	tpt	pips_mode	runtime[sec]	core_hours
1	12	4	hierarchical	7234.11	96.45
1	12	4	normal	2447.93	32.64
6	144	2	hierarchical	1176.29	94.10
6	144	2	normal	580.91	46.47
12	144	4	hierarchical	856.23	137.00
12	144	4	normal	293.83	47.01

Table 6.1: Runtime and core hour usage of the small instance with 144 time blocks executed on JUWELS for different hard- and software configurations

nodes	tasks	tpt	pips_mode	runtime[sec]	core_hours
2	48	2	hierarchical	1818.78	48.50
2	48	2	normal	2658.22	70.89
4	48	4	hierarchical	1479.39	78.90
4	48	4	normal	1995.91	106.45
192	2304	4	hierarchical	166.07	425.14
192	2304	4	normal	1049.26	2686.11

Table 6.2: Runtime and core hour usage of the small instance with 2304 timeblocks executed on JUWELS for different hard- and software configurations

and the distribution of the input data can be adjusted. Each of these parameters influences the runtime for calculating the solution and thus also the consumption of computing time, which in turn is only available to a limited extent. Therefore, an analysis was carried out to find the best compromise between runtime and computing time consumption in order to derive general rules and identify a sweet spot that can serve as a basis for further studies.

The basis for this analysis is the small instance with 8586301 rows, 8770175 columns and 29500378 non-zeros already used in the chapter 6.1, which was compared with a smaller number of time blocks (144) and a larger number (2304). In further tests, however, it could be shown that the results described below also apply to larger instances.

- With a small number of blocks, PIPS-IPM++ runs faster in normal mode than in hierarchical mode (see table 6.1).
- With a large number of blocks, PIPS-IPM++ runs faster in hierarchical mode than in normal mode (see table 6.2).
- The best runtime is achieved with a large number of blocks, a number of MPI tasks corresponding to the number of blocks and PIPS-IPM++ in hierarchical mode (see penultimate row in table 6.2).
- The lowest computing time consumption is achieved with a smaller number of blocks, a number of MPI tasks according to the formula $floor(sqrt(\#blocks))$ and PIPS-IPM++ in normal mode (see table 6.2).

Chapter 7

MIP heuristics

We consider a mixed-integer program given by:

$$\begin{aligned} \min \quad & c_1^T x + c_2^T z \\ \text{s.t} \quad & A \begin{bmatrix} x \\ z \end{bmatrix} \leq b \\ & x \in \mathbb{R}^{n_1}, z \in \mathbb{Z}^{n_2}. \end{aligned} \tag{7.1}$$

The goal is to find integer values for z that induce a low objective value while not violating the constraints. This is a hard problem: linear programming lies in the complexity class $\mathcal{P}$, while MIP is $\mathcal{NP}$-hard.

7.1 An approach based on reinforcement learning

Predicting integer fixings in the context of mixed-integer programming is a rather new, but active research topic. Bertsimas et al. (Bertsimas & Stellato, 2020) did not only predict the integer values, but the active set as well. This reduces the runtime of the (small) problems they tested their approach on to milliseconds and is most useful for online control applications. Another recent work is given by (Nair et al., 2020), where integers are digit-wise predicted and the prediction is combined with a deep learning branching oracle. A successful application specific application of deep learning can be found in (Anderson, Turner, & Koch, 2021). This work is application specific, as the neural network uses a handcrafted set of input features instead of the whole mixed-integer program. This seems applicable to energy system models as well, but requires a static formulation of the problem.

7.1.1 Reinforcement learning

In reinforcement learning, an agent interacts with an environment, observes the current state s_k of the environment and takes an action a_k with which the environment transitions into a new state s_{k+1}. Depending on a_k and s_{k+1}, it collects a reward r_{k+1}. The agent continues interacting with the environment until a terminal state is reached at time T. The corresponding sequence can be written as $s_1, a_1, r_2, s_2, a_2, \ldots, r_T, s_T$. The value $V(s)$ of a state s is given as the expected cumulative reward. There are different ways to train an agent to take good actions. A basic distinction is, if the agent predicts the action directly or the Q-value (the expected cumulative reward) of a state-action pair. If the agent predicts the action directly, it is usually updated with some sort of policy gradient.

In the context of UNSEEN, the environment corresponds to an LP solver, a state to an LP solution, an action to fixing the integers and a reward to the objective value of the LP solution (Figure 7.1).

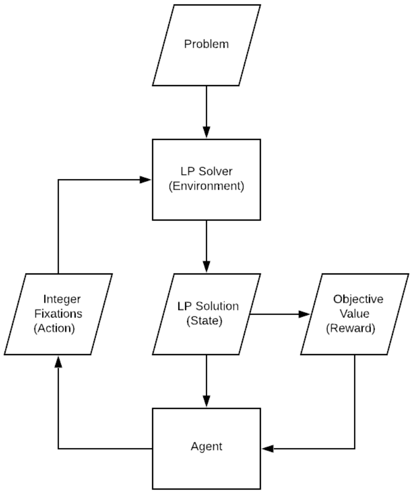


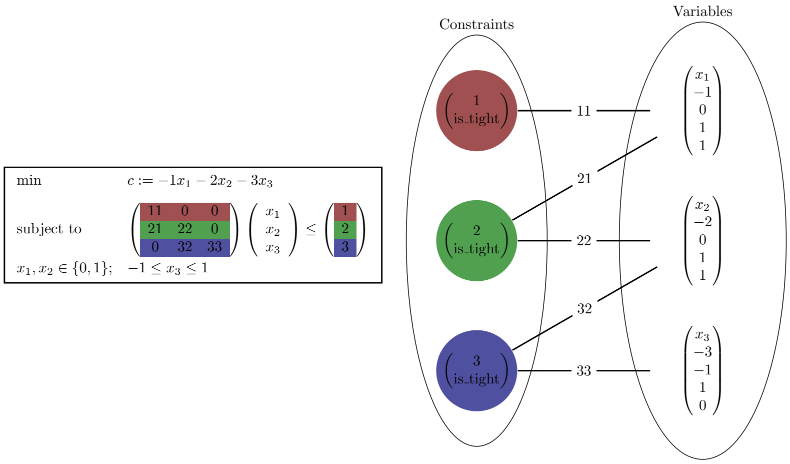
Figure 7.1: Reinforcement learning flow diagram in the context of UNSEEN.

In general, reinforcement learning is hard. It requires not only a lot of computational power, but also manual tuning as well as the correct choice of training method, architecture and reward. For generating integer values, the action space is large. In particular, it is too large for Q-Learning as training method, see (Watkins & Dayan, 1992). We deem actor-critic methods as promising, especially distributed variants as A3C (Mnih et al., 2016).

7.1.2 Graph Convolutional Neural Networks

Graph Convolutional Neural Networks (GCNNs) are a rather new neural network architecture, see e.g. (Kipf & Welling, 2017; Zhang, Tong, Xu, & Maciejewski, 2019). GCNN process graph structured input data: in each layer, data is processed and aggregated locally, i.e. inside neighborhoods in the graph. A well documented python library for GCNN is *torch-geometric*, see (Fey & Lenssen, 2019). GCNN have been sucessfully applied to solving MIPs represented as bipartite graphs (Gasse, Chételat, Ferroni, Charlin, & Lodi, 2019), see Figure 7.2.

We followed the neural network design of (Gasse et al., 2019), adapting it to a actor-critic design and including *LayerNorm* as mentioned in (Nair et al., 2020). For multi-step trajectories, we think LSTM cells could be promising in order to give the neural network a memory and make it more capable to develop strategies for exploring the search space that span the whole trajectory. This could be realized with an attention module (see (Vaswani et al., 2017)), whereby the attention coefficients are determined by an lstm module.



min $c := -1x_1 - 2x_2 - 3x_3$
subject to $\begin{pmatrix} 11 & 0 & 0 \\ 21 & 22 & 0 \\ 0 & 32 & 33 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \leq \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$
$x_1, x_2 \in \{0, 1\}$; $-1 \leq x_3 \leq 1$
Constraints
Variables
$\begin{pmatrix} 1 \\ \text{is\_tight} \end{pmatrix}$
$\begin{pmatrix} 2 \\ \text{is\_tight} \end{pmatrix}$
$\begin{pmatrix} 3 \\ \text{is\_tight} \end{pmatrix}$
11
21
22
32
33
$\begin{pmatrix} x_1 \\ -1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$
$\begin{pmatrix} x_2 \\ -2 \\ 0 \\ 1 \\ 1 \end{pmatrix}$
$\begin{pmatrix} x_3 \\ -3 \\ -1 \\ 1 \\ 0 \end{pmatrix}$

Figure 7.2: Left: a small MIP. Right: the corresponding bipartite graph. Each node has a feature vector attached. The feature dimension of the edge is 1.

7.1.3 Difficulties

Unfortunately, the prediction of integers by neural networks is not possible out of the box, therefore we were not able to solve UNSEEN instances with general integers. However, we considered creating equivalent models with only binary variables, which turned out to be impractical on account of the exploding size of the instances. An approach to output general integers is given by (Nair et al., 2020), where the neural network is used k times to predict the k most significant digits in the binary representation of the integer. This approach could be seen as a form of branching inside a prediction loop, since the bounds on the integers are updated with each iteration.

7.2 A simulated annealing approach

Simulated annealing (Kirkpatrick, Gelatt Jr, & Vecchi, 1983) is a metaheuristic inspired by the cooling process of a heated material, which tends to a level of low energy of the physical system. An iterative method, simulated annealing attemps to find the solution of an optimization problem by performing random perturbations on the current iterate and evaluating its objective value. The trial iterate is accepted if it improves upon the current iterate (for some criterion) or, if it degrades it, with a probability that decreases to 0 over the iterations (it is a function of the decreasing "temperature" of the physical system). The algorithm is summed up in Algorithm 1.

7.2.1 An improvement criterion for constrained optimization

Historically, the algorithm has been applied to combinatorial problems (such as the Traveling Salesman Problem) in which the improvement criterion is a decrease of the objective function. However, this criterion is irrelevant for constrained optimization, as feasibility should take precedence over the objective value. We therefore investigated with the following improvement criterion that balances

Algorithm 1: Generic simulated annealing.

```
Input: initial iterate $z^{(0)}$, probability $p^{(0)}$
$k \leftarrow 0$
while termination criteria not satisfied do
    trial iterate: randomly generate $\hat{z}^{(k+1)}$ in the neighborhood of $z^{(k)}$
    acceptance test: if $\hat{z}^{(k+1)}$ "better" than $z^{(k)}$ then
        accept $\hat{z}^{(k+1)}$: $z^{(k+1)} \leftarrow \hat{z}^{(k+1)}$
    else if $rand(0, 1) < p^{(k)}$ then
        accept $\hat{z}^{(k+1)}$: $z^{(k+1)} \leftarrow \hat{z}^{(k+1)}$
    else
        keep $z^{(k)}$
    $k \leftarrow k + 1$
    Decrease $p^{(k)}$
return $z^{(k)}$
```

objective and constraint violation:

- if the current iterate is feasible:
  - if the trial iterate is feasible, compare the objective values;
  - otherwise, keep the current iterate.
- otherwise:
  - if the trial iterate is feasible, accept the trial iterate;
  - otherwise, compare the infeasibility values.

7.2.2 Difficulties

Although simulated annealing has proven efficient for numerous black-box problems, it requires many evaluations of the problem's functions. In the context of MIP solving, the objective value and the constraint violation of a trial iterate are obtained after fixing the integers to the trial values and solving the resulting LP. Since solving LPs is the computational bottleneck, simulated annealing (and any metaheuristic) requires tremendous computational effort. Moreover, integer fixings may result in infeasible LPs, which our LP solver PIPS-IPM++ cannot reliably detect. Consequently, simulated annealing does not seem to be a suitable primal heuristic to find satisfactory integer fixings.

7.3 Fix-propagate-repair heuristic

Presolving UNSEEN instances reduces the number of columns (variables) and rows (constraints) by more than a half, which in turn accelerates convergence of LP and MIP solvers. It is therefore crucial that any strategy handles the presolved problem instead of the original problem. The reduced problem (the LP that results from fixing the integers in the MIP) can again be presolved, which further reduces the size of the problem.

A heuristic that leans heavily on presolving and propagation would allow us to fix as few integers as possible, thereby reducing the chances of a mistake being made. This lead us to experiment with the fix-propagate-repair (Fix-propagate-repair (FPR)) heuristic (Salvagnin, Roberti, & Fischetti, 2022).

7.3.1 Description

FPR is a modular strategy that relies on generic presolve, propagation, repair and solve routines. It is interfaced with commercial solvers CPLEX, GUROBI, COPT and XPRESS. FPR stands out from other MIP heuristics as it requires few LP solves. Since the repair step is not used for solving UNSEEN instances, we describe the simpler fix-propagate (FP) strategy in Algorithm 2. The MIP is reduced by the presolving phase of the solver, then each discrete (binary, then integer) variable is fixed to an integer value according to one of few simple heuristics:

- "good_obj": direction of decreasing objective;
- "bad_obj": direction of increasing objective;
- random;
- LP-based: round the root LP solution to nearest;

The fixing is then propagated through the constraints. When no variable remains to be fixed, the resulting reduced LP is solved. FPR implements a depth-first search and triggers a backtracking phase when an integer fixing was proven to cause infeasibility.

Algorithm 2: Fix-and-propagate heuristic

```
Data: MIP with integer variables $z$
Result: MIP solution
Presolve MIP
for $z_i$ in $z$ do
    Fix $z_i$ to an integer value, depending on one of few simple heuristics
    infeasible integer fixing ← propagate(MIP, $z$)
    if infeasible integer fixing then
        backtrack on fixing
    end
    if all integer variables fixed then
        break
    end
end
Solve resulting LP
return MIP solution
```

7.3.2 Numerical results

We tested the FPR heuristic (interfaced with LP solvers COPT, CPLEX, GUROBI and XPRESS) and the commercial MIP solvers GUROBI and COPT on REMix instances of all sizes. All LP solvers run a barrier method without crossover (note that the GUROBI crossover could not be disabled, which prevented relevant results on large instances). For the FPR(COPT), FPR(GUROBI) and

FPR(XPRESS) combinations, CPLEX was used for presolving the problem. All algorithms were run on a machine with 176 Intel(R) Xeon(R) CPU E7-8880 v4 @ 2.20GHz and 2 TB of RAM. COPT and GUROBI were run with a MIP relative gap of 1% and a time limit of 24 hours. **Because a single large instance requires 8 to 15 hours of computation, only a limited number of large instances could be run with particular solvers when convergence was slow.** All tests were performed with 32 threads, which seems to be the sweet spot for FPR. The FPR integer fixing strategy (either bad_obj or good_obj) is indicated in parentheses.

In Figure 7.3, the distribution of CPU times (wall clock) is given in a boxplot with logarithmic scale. Only the solved instances are shown. Tables 7.1, 7.2 and 7.3 indicate the percentage of solved instances and the average CPU time. As mentioned in the previous paragraphs, the amount of instances had to be reduced for several scenarios (2 instances for FPR (XPRESS, bad_obj) on the REMix 57M and 10 instances for the REMix 105M) on account of slow convergence.

FPR scales remarkably well and better than GUROBI and COPT; both commercial solvers time out for larger instances. Although the root LP is solved relatively quickly with a barrier method, the crossover and the branch and cut method cannot converge within reasonable time. For large instances however, the number of barrier iterations consistently reaches 400-500 iterations, which indicates that the instances are numerically challenging. The combination FPR(COPT) is the best heuristic option for solving large-scale REMix MIP instances. This suggests that COPT implements the most robust LP solvers of all the commercial solvers. On the other hand, GUROBI performs the best on the MIP instances.

The next step would be to interface FPR and PIPS-IPM++ to exploit the structure of the problem and increase the dimension of the problems that can be solved by the FPR heuristic. The task of interfacing these two substantially large codes with different data structures was too demanding to be carried out during the scope of the project extension.

	Solved instances	Avg. time in [s]
FPR (COPT, good_obj)	100/100	49.59
FPR (XPRESS, good_obj)	100/100	55.89
FPR (GUROBI, good_obj)	100/100	61.13
FPR (CPLEX, good_obj)	99/100	95.08
GUROBI	100/100	955.85
COPT	100/100	1,938.93

Table 7.1: Numerical results on REMix 10M instances.

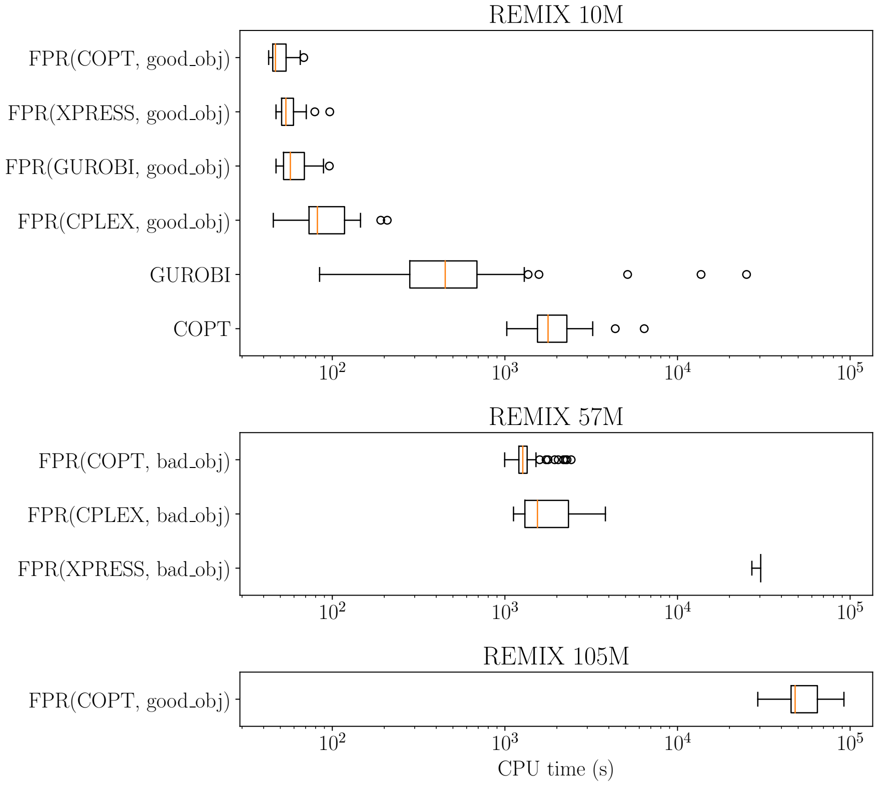


Figure 7.3: Distribution of CPU times on solved REMix instances.

	Solved instances	Avg. time in [s]
FPR (COPT, bad__obj)	69/100	1,346.56
FPR (XPRESS, bad__obj)	2/2	30,330.89
FPR (CPLEX, bad__obj)	81/100	1,823.69
GUROBI	0/100	
COPT	0/100	

Table 7.2: Numerical results on REMix 57M instances.

	Solved instances	Avg. time in [s]
FPR (COPT, good_obj)	9/10	54,636.69
FPR (XPRESS, good_obj)	0/10	
FPR (GUROBI, good_obj)	0/10	
FPR (CPLEX, good_obj)	0/10	
GUROBI	0/10	
COPT	0/10	

Table 7.3: Numerical results on REMix 105M instances.

7.4 Rounding heuristics

7.4.1 A posteriori analysis of REMix instances

We performed an a posteriori analysis of REMix instances with 244k, 2.8M and 47M nonzeros, respectively. In particular, we examined the value of the *"root" duality gap* (the difference between the MIP objective and the root LP objective in which integrality constraints have been relaxed) on 1,000 REMix 2.8M instances for which the MIP and the root LP were solved with GUROBI. The results are portrayed in Figure 7.4: the values of the root duality gap seem to follow a normal distribution with mean ≈ 0.048%. This is a remarkably low value that suggests that the lower bound obtained by solving the root LP is almost optimal for REMix instances.

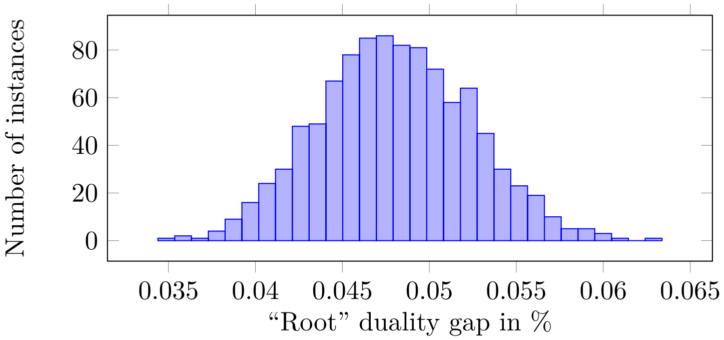


Figure 7.4: Value of the "root" duality gap for 1,000 REMix 2.8M instances.

The evolution of the lower and upper bounds during the MIP optimization process is schematically represented in Figure 7.5: the upper bound decreases much more than the lower bound increases. Consequently, the standard MIP techniques aiming at improving the lower bound, such as branching or cutting planes, are largely irrelevant in the context of UNSEEN. In order to improve the duality gap, we should invest effort into developing primal MIP heuristics that seek integer-feasible points.

Furthermore, we examined the (normalized) Euclidean distance between the root LP solution z^{LP} and the MIP solution z^{MIP}. Figure 7.6 shows their proximity for 1,000 REMix 2.8M instances, with an average normalized distance of $2.2 \cdot 10^{-3}$. These results suggest that rounding the root LP solution to the nearest integer may be a cheap and efficient way to approximate the MIP solution.

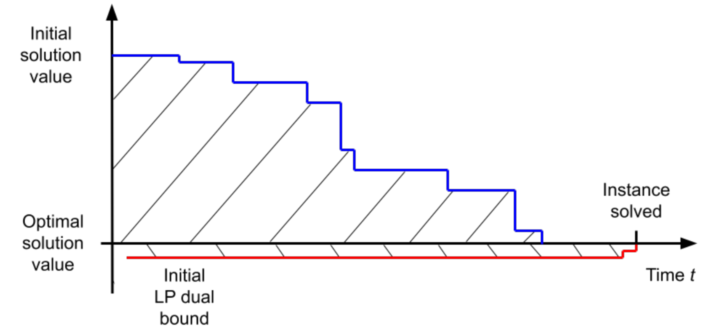


Figure 7.5: Schematic evolution of the lower and upper bounds for UNSEEN instances.

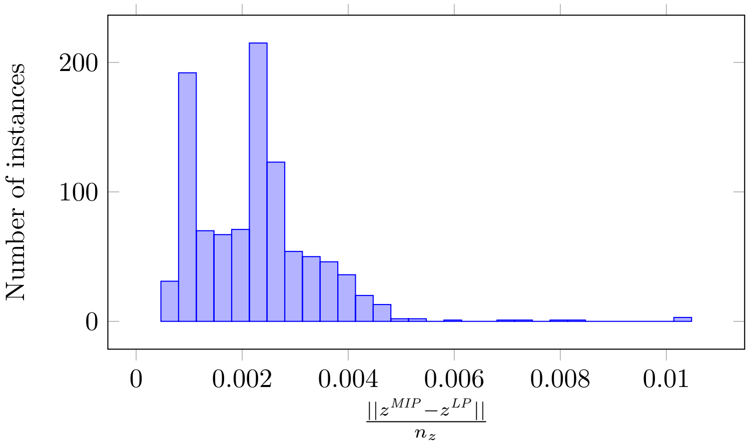


Figure 7.6: Normalized Euclidean distance between the root LP solution and the MIP solution for 1,000 REMix 2.8M instances.

The amount of integer values in the fractional root LP solution suggests a high degree of agreement with the MIP solution. Figure 7.7 shows how close rounding the LP solution can get us to the MIP solution. The rounded LP solution showed on average 93.1% agreement with the MIP solution. Motivated by the encouraging results and the potential of such a cheap heuristic, we developed two methods based on rounding the root LP solution: "relax and round" and "learn how to round".

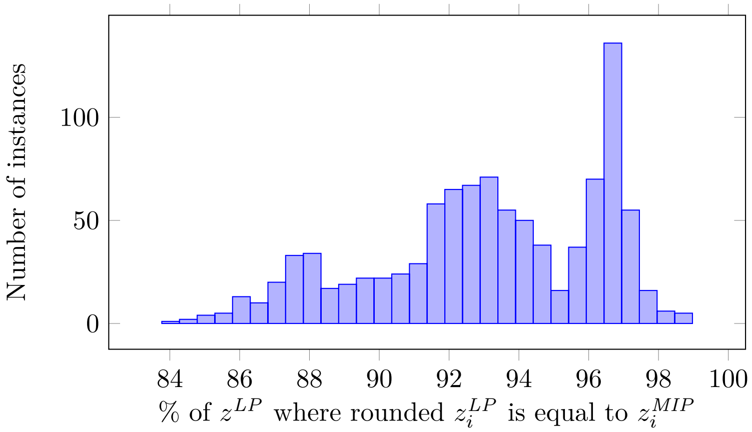


Figure 7.7: Distribution of percentages of rounded root LP solution that agrees with MIP solution for 1,000 REMix 2.8M instances.

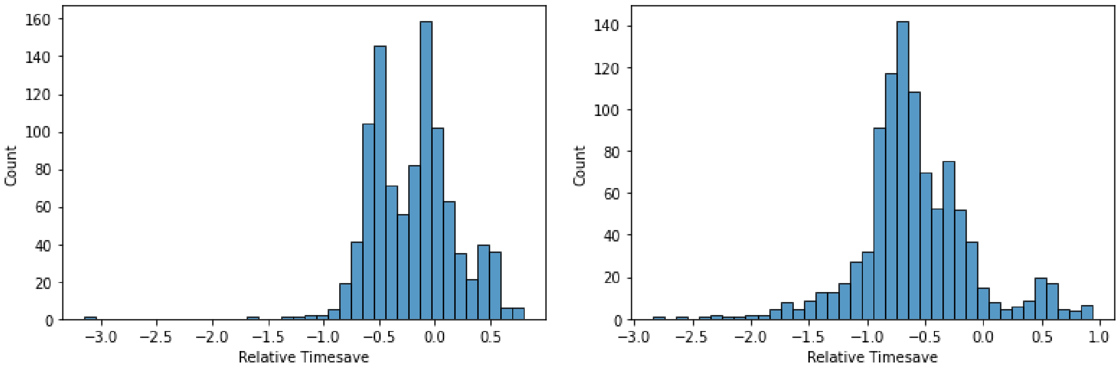
These results taught us a lot about the nature of UNSEEN instances. They suggest a strong proximity between the fractional root LP solution (that can be solved in polynomial time) and the MIP solution (NP-hard), thus reducing the gap between complexity classes. This motivated the development of two approaches based on rounding: the "relax and round" approach (Section 7.4.2) and the the "learn how to round" neural approach (Section 7.4.3).

7.4.2 The "relax and round" approach

The "relax and round" approach is based on the following steps:

1. relax the integrity constraints in the original MIP and solve the root LP;
2. round to nearest the integer variables in the (fractional) root LP solution and fix their values in the original MIP;
3. solve the resulting reduced LP.

Although this does not guarantee that the integer fixing results in a feasible reduced LP, we observed that all REMix instances with 244k and 2.8M nonzeros remain feasible. Figure 7.8 provides the relative CPU time save $\frac{\text{MIP time}-\text{“relax and round” time}}{\text{MIP time}}$ on two different machines (a ZIB desktop machine and the ZIB cluster) provided by the "relax and round" strategy compared to a MIP solver (GUROBI) on 1,000 REMix 244k instances. By definition, the time save is positive if the heuristic is faster than the MIP solver, and negative if it is slower. Surprisingly, the trend is out of favor for the heuristic: on the desktop machine, most instances see a slowdown when the heuristic is used. When run on the Z1 cluster, the heuristic is virtually consistenstly worse than the MIP solver.



(a) On a ZIB desktop machine

(b) On the ZIB Z1 cluster

Figure 7.8: Relative time save of "relax and round" compared to GurobiPy.

We reckon that this is due to the small size (244k nonzeros) of the instances, which are not particularly computationally intensive. Moreover, the MIP solver can generally fully exploit large-scale parallelism when run on a cluster.

7.4.3 The "learn how to round" neural approach

We investigated with a learning strategy based on a simple multi-layer perceptron neural network architecture. The input to the network is the integer part of the LP solution. The output is a classification of variables: +1 corresponds to "round up", -1 corresponds to "round down", and 0 is "leave alone" (for variables which cannot be rounded accurately). We used ReLU units in the hidden layers and Tanh units in the output layer. The number of layers and the number of hidden units per layer were determined in a validation phase before learning. Training data was generated by solving the instances with GUROBI. We used an ℓ_1 loss function and the Adam optimizer (Kingma & Ba, 2014). The implementation was done using the machine learning library PyTorch (Paszke et al., 2019). The classification accuracy (on average, how many variables are correctly rounded) during the training phase is shown in Table 7.4. As expected, the larger the instances, the lower the achieved classification accuracy.

	Classification accuracy	Size of training dataset	(Hidden Layers) × (Neurons)
Small	95.9%	900	3 × 400
Medium	88.7%	9,000	4 × 300
Large	83.5%	956	3 × 300

Table 7.4: "Learn how to round": results of the training phase.

The full neural strategy, implemented as a heuristic in SCIP (Bestuzheva et al., 2021) using the Python interface PySCIPOpt (Maher et al., 2016) and GUROBI as LP solver, is described in Algorithm 3. After fixing the integers to the values predicted by the neural network, the SCIP constraint propagation algorithm is invoked to reduce the ranges of the variables. Backtracking mitigates the risk of infeasibility caused by the predicted integer fixings. When all integer variables have been fixed (either actively through rounding or passively during propagation), we solve the resulting reduced LP.

```
Algorithm 3: Integer fixing using machine learning.
Data: MIP with integer variables $z$, neural network $NN$
Result: MIP solution
Presolve MIP
Solve root LP to obtain $z^{LP}$
Compute rounding instructions $\phi = NN(z^{LP})$
for $z_i$ in $z$ do
    if $\phi_i = 1$ then
        round $z_i^{LP}$ up and fix
    else if $\phi_i = -1$ then
        round $z_i^{LP}$ down and fix
    infeasible integer fixing ← propagate(MIP, $z$)
    if infeasible integer fixing then
        backtrack on fixing
    end
end
Solve resulting reduced LP
return MIP solution
```

The results of the neural strategy on instances with 244k and 2.8M nonzeros are summarized in Table 7.5. The neural approach delivers decent results on test instances, albeit solves fewer instances than promised by our network's classification accuracy. In most of the cases, infeasibility is addressed by the propagation phase. As the small average duality gaps suggest, the neural strategy succeeds in mimicking the behavior of a MIP solver reasonably well for the considered instances. Two tricks vastly improved performance:

1. we analyzed the network's performance and ordered the integer fixing according to the best predicted (on average) integer variables. This prevents integers to be prematurely fixed, which causes unrecoverable infeasibility;

2. we increased the maximal number of propagation rounds in SCIP. It is therefore more likely that infeasibility is detected early and addressed through backtracking.

Table 7.5 also illustrates the effect of the number of propagation rounds (rows M with 1, 5 and 10 rounds): when more propagation is allowed, the number of solved instances dramatically increases and the duality gap is marginally reduced.

	Propagation rounds	Instances solved	Average duality gap (solved instances)
S	10	90%	0%
M	1	10%	0.18%
	5	55%	0.13%
	10	85%	0.13%

Table 7.5: Machine learning heuristic results on test set instances.

While promising, the approach does not scale well with the dimension of the problem. Indeed, generating and storing a large amount of training data to train an accurate large-scale classifier is computationally costly. While GUROBI was able to solve small LPs and MIPs fairly quickly (therefore data generation was not an issue), it required on average 3 hours to solve some REMix 47M instances, and in some cases timed out after 24 hours. We have also empirically observed (and it was expected given the training accuracy) that for the REMix 2.8M instances, the best possible classification accuracy was 7 points below that for the REMix 2.8M instances, and the classification accuracy for REMix 47M instances was lower still.

More sophisticated network architectures and learning methods may well scale better for our problem. However they require more trial and error and it is difficult to know a priori how feature selection and hyperparameter selection affect learning. They may however address the scaling issue: a different neural network was trained for every instance size, even though all the instances model similar energy networks with common structure. This fails to transfer the knowledge from one to the other. Graph Convolutional Neural Networks (GCNN, see Section 7.1.2) may be able to address this issue, provided that there are only binary variables and no integer variables. The fact that GCNNs take as input graph representations of variable size means that a single GCNN can be trained on instances of smaller size and make predictions for (to some extent) larger instances. In the UNSEEN context in which large-scale training data is harder and more expensive to generate, this is a very advantageous characteristic.

Unfortunately, REMix instances appear to be noticeably more computationally challenging than REMix instances. Scalability capabilities of the rounding-based approaches seem therefore less attractive than for REMix instances. While a fruitful option for future research, neural rounding strategies exhibit severe drawbacks. Even with high accuracy (above 96% for REMix 244k instances), the predicted integer fixings consistently cause unrecoverable infeasibility despite the constraint propagation phase. This trend accentuates with the dimension of the problem, which renders the rounding-based methods as yet unpractical. The approach also relies on solving two LPs (the root LP and the reduced LP), which becomes extremely costly for larger instances.

Chapter 8

Conclusion

8.1 The value of Neural Networks

We established a reinforcement loop and learned that focusing on integer fixing using neural networks (NN) would be beneficial. Specifically, we developed NN for integer fixing. Neural networks prove useful for learning structures beyond the block structure needed for in PIPS-IPM++. In our case, we successfully trained neural networks to learn rounding strategies (see section 7). These learned structures helped us to identify suitable heuristics from the numerous strategies published in literature.

8.2 Strategies to solve ESOMs based on mixed-integer programs

We implemented and created scalable model instances for energy system optimization models ESOMs with discrete investment and unit commitment Unit Commitment (UC). During testing, we explored different implementations, including exclusively using binary formulations. Additionally, we generated model instances to serve as benchmarks, with a runtime of 24 hours on JUWELS. From an energy system analysis perspective (using a commercial solver), the most meaningful instances have varying spatial resolutions, whereas higher granularity is applied to regions requiring more detailed examination. In these highly resolved regions, integer variables are employed and specific technology modeling with UC is considered. Thus, we structured the MIPs in a way that is conceptually meaningful and suitable for algorithm testing. From the algorithmic side we learned that models should be formulated without slack power generation.

Our next step involves integrating an effective LP solver into a MIP framework that incorporates rounding heuristics. PIPS-IPM++ is a potential choice for this integration. Two approaches are under consideration:

1. Parallel Heuristics: Integrating a solution heuristic into PIPS-IPM++. However, this poses challenges in a scientific environment, requiring significant solver development and coding skills to parallelize an existing heuristic like PIPS-IPM++.

2. Out-of-the-box Heuristics: Combining the workflow from PIPS-IPM++ with a rounding heuristic. Key requirements for this approach include: Minimizing the frequency of LP solves, utilizing FPR heuristics with infrequent LP solves, leveraging PIPS-IPM++' strengths in parallelism and propagation, even if it conflicts with the FPR system's structure, selecting a heuristic capable of modifying/adapting multiple variables simultaneously[1] and identifying the advantages of integrating the FPR heuristic into PIPS-IPM++[2].

Furthermore, we learned that artificial solutions (using NN) can be used as tools to find algorithmic solutions that outperform the artificial solutions themselves. In particular, with our generic approach, we can solve MIP instances in real-world applications that exceed our technical capabilities by utilizing commercial solvers. Finally, we discovered the significance of good rounding. However, a direction for future research is to investigate, if this a general phenomenon [3] LP-model HPC Workflow:

8.3 New horizons in Energy System Optimization

In the field of energy system analysis, we have reached the cutting edge when it comes to ensemble calculations. One important aspect of this progress is the coupling of models, which is necessary for identifying blind spots in our understanding. Through effective automation driven by the requirements of HPC environments, we have successfully accomplished model coupling. Automating the coupling process is particularly important as ESOMs become increasingly complex.

The Scenario Driver has been designed to be as generic as possible, but it remains highly specific to the REMix model. It is not feasible or meaningful to create a model-agnostic Scenario Driver. However, it provides very useful features. It offers flexible control through parameters, allowing for the management of modules and folder structures and runs reliably and consistently. The Scenario Generator component is already highly generic, using GAMS-readable symbols in CSV format along with corresponding information for sampling.

When it comes to validating our findings and ensuring the robustness of our experiments, we have learned valuable lessons from past experiences. Now, we can focus on implementing change management strategies. Conducting initial tests on small prototype models and performing both unit testing and integration testing are straightforward approaches that are essential for improving validation. Additionally, visualizing the workflow, including the relevant data, is of utmost importance. However, keeping the visualization up to date can be challenging. It is crucial to collaborate with experts in model coupling and foster strong interdisciplinary cooperation to navigate the complexities of these experiments successfully. Furthermore, we successfully solved the largest LP problem in the energy system analysis domain, PyPSA-Eur (Cao, Wetzel, Kempke, & Koch, 2021). However, for a successful technology transfer, PIPS-IPM++ should be made more resilient and less prone to errors.

Looking ahead, our goal is to fully integrate, encapsulate, and automate the workflow for end-users. However, achieving this objective will likely require a significant amount of non-scientific work.

[1] Although the heuristic section may not be parallel, it is unlikely to be the bottleneck.

[2] assuming it is not the bottleneck, parallel solving might not be necessary

[3] Note: In model experiments we conducted for different ESOMs, we found only a few LP models behaved well in terms of PIPS-IPM++ being a suitable solver for them.

Each individual component of the workflow will need to be generalized and made more robust to ensure seamless and efficient operation.

Bibliography

Abdin, I. F., & Zio, E. (2018, July). An integrated framework for operational flexibility assessment in multi-period power system planning with renewable energy production. *Applied Energy*, *222*, 898–914. (Publisher: Elsevier BV) doi: 10.1016/j.apenergy.2018.04.009

Achterberg, T., Bixby, R., Gu, Z., Rothberg, E., & Weninger, D. (2019, 11). Presolve reductions in mixed integer programming. *INFORMS Journal on Computing*, *32*. doi: 10.1287/ijoc.2018.0857

AEBIOM. (2019). *Annual statistical report on the contribution of biomass to the energy system in the eu28* (Tech. Rep.). Bioenergy Europe. Retrieved from `http://www.europeanbioenergyday.eu/bioenergy-facts/bioenergy-in-europe/what-are-the-volumes-of-biomass-used-in-the-eu28-to-produce-energy/`

Andersen, E. D., & Andersen, K. D. (1995). Presolving in linear programming. *Mathematical Programming*, *71*(2), 221–245.

Anderson, L., Turner, M., & Koch, T. (2021). Generative deep learning for decision making in gas networks. *arXiv preprint arXiv:2102.02125*.

Berntsen, P. B., & Trutnevyte, E. (2017). Ensuring diversity of national energy scenarios: Bottom-up energy system model with modeling to generate alternatives. *Energy*, *126*, 886–898. doi: 10.1016/j.energy.2017.03.043

Bertsimas, D., & Stellato, B. (2020, June). Online Mixed-Integer Optimization in Milliseconds. *arXiv:1907.02206 [cs, math]*. Retrieved 2020-09-04, from `http://arxiv.org/abs/1907.02206` (arXiv: 1907.02206)

Bestuzheva, K., Besançon, M., Chen, W.-K., Chmiela, A., Donkiewicz, T., van Doornmalen, J., … others (2021). The scip optimization suite 8.0. *arXiv preprint arXiv:2112.08872*.

Bird, L., Cochran, J., & Wang, X. (2014). *Wind and solar energy curtailment: experience and practices in the united states* (Tech. Rep.). National Renewable Energy Lab.(NREL), Golden, CO (United States).

Blanco, M. P., Spisto, A., Hrelja, N., & Fulli, G. (2016). Generation adequacy methodologies review. *JRC Science for Policy report*.

Breuer, T., Lührs, S., Klasen, A., & Wellmann, J. (2022, August). *Jube*. Zenodo. Retrieved from `https://doi.org/10.5281/zenodo.7534373` doi: 10.5281/zenodo.7534373

Burke, D. J., & O'Malley, M. J. (2011). Factors influencing wind energy curtailment. *IEEE Transactions on Sustainable Energy*, *2*(2), 185–193.

Cao, K.-K., Wetzel, M., Kempke, N.-C., & Koch, T. (2021). Pushing computational boundaries: Solving integrated investment planning problems for large-scale energy systems with pips-ipm+. In *Operations research 2021.* Retrieved from `https://elib.dlr.de/143744/`

Colombo, M., & Gondzio, J. (2008, Dec 01). Further development of multiple centrality correctors for interior point methods. *Computational Optimization and Applications*, *41*(3), 277–305. Retrieved from `https://doi.org/10.1007/s10589-007-9106-0` doi: 10.1007/s10589-007-9106-0

Dagoumas, A. (2021). *Mathematical modelling of contemporary electricity markets edited by athanasios dagoumas.* London: Academic Press.

DeCarolis, J. F. (2011). Using modeling to generate alternatives (mga) to expand our thinking on energy futures. *Energy Economics*, *33*(2), 145–152. doi: 10.1016/j.eneco.2010.05.002

DeCarolis, J. F., Babaee, S., Li, B., & Kanungo, S. (2016). Modelling to generate alternatives with an energy system optimization model. *Environmental Modelling & Software*, *79*, 300–310. doi: 10.1016/j.envsoft.2015.11.019

Du, H., Chen, Z., Brown, M. A., Yin, Y., Zuo, J., & Chai, L. (2020). How secure are national energy systems: A dynamic assessment approach. *Ecological Indicators*, *108*, 105666. Retrieved from `http://www.sciencedirect.com/science/article/pii/S1470160X19306594` doi: https://doi.org/10.1016/j.ecolind.2019.105666

Dupačová, J., Gröwe-Kuska, N., & Römisch, W. (2003). Scenario reduction in stochastic programming. *Mathematical Programming*, *95*(3), 493-511.

Elsland, R., Pfluger, B., Tersteegen, B., Franke, B., Boßmann, T., Fleiter, T., … Reiter, U. (2017, 09). *Langfristszenarien für die transformation des energiesystems in deutschland* (Tech. Rep.).

ENTSO-E. (2014). *2015 scenario outlook and adequacy forecast* (Tech. Rep.). European Network of Transmission System Operators for Electricity.

Fette, M., Brandstätt, C., Gils, H. C., Gardian, H., Pregger, T., Schaffert, J., … Brücken, N. (2020, August). *Multi-sektor-kopplung - modellbasierte analyse der integration erneuerbarer stromerzeugung durch die kopplung der stromversorgung mit dem wärme-, gas- und verkehrssektor* (Tech. Rep.). Retrieved from `https://elib.dlr.de/135971/`

Fey, M., & Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. In *Iclr workshop on representation learning on graphs and manifolds.*

Gamrath, G., Koch, T., Martin, A., Miltenberger, M., & Weninger, D. (2015, 06). Progress in presolving for mixed integer programming. *Mathematical Programming Computation*, *7*. doi: 10.1007/s12532-015-0083-5

GAMS. (n.d.). *Stochastic programming.* `https://www.gams.com/39/docs/UG_EMP_SP.html`.

GAMS. (2022). *39 distribution.* `https://www.gams.com/43/docs/RN_39.html`.

Garcia-Teruel, A., Scholz, Y., Weimer-Jehle, W., Prehofer, S., Cao, K.-K., & Borggrefe, F. (2022). Teaching power-sector models social and political awareness. *Energies*, *15*(9), 3275.

Gasse, M., Chételat, D., Ferroni, N., Charlin, L., & Lodi, A. (2019). Exact combinatorial optimization with graph convolutional neural networks. *Advances in neural information processing systems*, *32*.

Gemander, P., Weikun, C., Weninger, D., Gottwald, L., Gleixner, A., & Martin, A. (2020, 08). Two-row and two-column mixed-integer presolve using hashing-based pairing methods. *EURO Journal on Computational Optimization*, *8*. doi: 10.1007/s13675-020-00129-6

Gleixner, A., Kempke, N.-C., Koch, T., Rehfeldt, D., & Uslu, S. (2020). First experiments with structure-aware presolving for a parallel interior-point method. In *Operations research proceedings 2019: Selected papers of the annual international conference of the German Operations Research society (GOR), dresden, germany, september 4-6, 2019* (pp. 105–111).

Gondzio, J. (1997). Presolve analysis of linear programs prior to applying an interior point method. *INFORMS Journal on Computing*, *9*(1), 73-91.

Guo, Z., Wei, W., Chen, L., Dong, Z. Y., & Mei, S. (2020). Impact of energy storage on renewable energy utilization: A geometric description. *IEEE Transactions on Sustainable Energy*, *12*(2), 874–885.

Heitsch, H., & Römisch, W. (2003). Scenario reduction algorithms in stochastic programming. *Computational Optimization and Applications*, *24*, 187-206.

IEA. (2014, Nov). World energy outlook 2014. *World Energy Outlook.* Retrieved from `http://dx.doi.org/10.1787/weo-2014-en` doi: 10.1787/weo-2014-en

IEA. (2019, Nov). World energy outlook 2019. *World Energy Outlook.* Retrieved from `http://dx.doi.org/10.1787/caf32f3b-en` doi: 10.1787/caf32f3b-en

Jacobsen, H. K., & Schröder, S. T. (2012). Curtailment of renewable generation: Economic optimality and incentives. *Energy Policy*, *49*, 663–675.

Joe, H. (2016). *Dependence modeling with copulas.* Chapman and Hall/CRC.

Jülich Supercomputing Centre. (2021). JUWELS Cluster and Booster: Exascale Pathfinder with Modular Supercomputing Architecture at Juelich Supercomputing Centre. *Journal of large-scale research facilities*, *7*(A138). Retrieved from `http://dx.doi.org/10.17815/jlsrf-7-183` doi: 10.17815/jlsrf-7-183

Junne, T., Cao, K.-K., Miskiw, K. K., Hottenroth, H., & Naegler, T. (2021). Considering life cycle greenhouse gas emissions in power system expansion planning for europe and north africa using multi-objective optimization. *Energies*, *14*(5), 1301.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. *arXiv.*

Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, *220*(4598), 671–680.

Kisel, E., Hamburg, A., Härm, M., Leppiman, A., & Ots, M. (2016). Concept for energy security matrix. *Energy Policy*, *95*, 1 - 9. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0301421516302038` doi: https://doi.org/10.1016/j.enpol.2016.04.034

Lannoye, E., Flynn, D., & O'Malley, M. (2012, May). Evaluation of Power System Flexibility. *IEEE Transactions on Power Systems*, *27*(2), 922–931. (Publisher: Institute of Electrical and Electronics Engineers (IEEE)) doi: 10.1109/tpwrs.2011.2177280

Lesche, S. (2022). *Entwicklung der Brennstoffpreise von 2013 bis 2022* (Tech. Rep.). Technologie- und Förderzentrum im Kompetenzzentrum für Nachwachsende Rohstoffe. Retrieved from `https://www.tfz.bayern.de/mam/cms08/festbrennstoffe/dateien/merkblatt_entwicklung_der_brennstoffpreise.pdf`

Ma, J., Silva, V., Belhomme, R., Kirschen, D. S., & Ochoa, L. F. (2012, October). Exploring the use of flexibility indices in low carbon power systems. In *2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe).* IEEE. doi: 10.1109/isgteurope.2012.6465757

Maher, S., Miltenberger, M., Pedroso, J. P., Rehfeldt, D., Schwarz, R., & Serrano, F. (2016). PySCIPOpt: Mathematical programming in python with the SCIP optimization suite. In *Mathematical software – ICMS 2016* (pp. 301–307). Springer International Publishing. doi: 10.1007/978-3-319-42432-3_37

Mavrotas, G. (2009). Effective implementation of the epsilon-constraint method in multi-objective mathematical programming problems. *Applied Mathematics and Computation*, *213*(2), 455–465. doi: 10.1016/j.amc.2009.03.037

Michi, L., Donnini, G., Aluisio, B., Migliori, M., Vergine, C., Dicorato, M., & Forte, G. (2019).

The dc power planning for network flexibility in the multi-area power system. In *2019 aeit international annual conference (aeit)* (pp. 1–5).

Michi, L., Migliori, M., Bugliari, A. C., Aluisio, B., Giannuzzi, G. M., & Carlini, E. M. (2018). Transmission network expansion planning: towards enhanced renewable integration. In *2018 aeit international annual conference* (pp. 1–5).

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., … Kavukcuoglu, K. (2016). *Asynchronous methods for deep reinforcement learning.*

Nair, V., Bartunov, S., Gimeno, F., von Glehn, I., Lichocki, P., Lobov, I., … Zwols, Y. (2020, December). Solving Mixed Integer Programs Using Neural Networks. *arXiv:2012.13349 [cs, math]*. Retrieved 2021-01-03, from http://arxiv.org/abs/2012.13349 (arXiv: 2012.13349)

Nelsen, R. B. (2006). *An introduction to copulas.* Springer-Verlag New York.

Neumann, F., & Brown, T. (2021). The near-optimal feasible space of a renewable power system model. *Electric Power Systems Research*, *190*, 106690. Retrieved from https://www.sciencedirect.com/science/article/pii/S0378779620304934 doi: 10.1016/j.epsr.2020.106690

Nitsch, J., Pregger, T., Naegler, T., Heide, D., Luca de Tena, D., Trieb, F., … others (2012). Langfristszenarien und strategien für den ausbau der erneuerbaren energien in deutschland bei berücksichtigung der entwicklung in europa und global.

Papalexopoulos, A., Hansen, C., Frowd, R., Tuohy, A., & Lannoye, E. (2016, June). Impact of the transmission grid on the operational system flexibility. In *2016 Power Systems Computation Conference (PSCC).* IEEE. doi: 10.1109/pscc.2016.7541027

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Poncela, M., Purvins, A., & Chondrogiannis, S. (2018). Pan-european analysis on power system flexibility. *Energies*, *11*(7), 1765.

Price, J., & Keppo, I. (2017). Modelling to generate alternatives: A technique to explore uncertainty in energy-environment-economy models. *Applied Energy*, *195*, 356–369. doi: 10.1016/j.apenergy.2017.03.065

Rehfeldt, D. e. a. (2021). A massively parallel interior-point solver for lps with generalized arrowhead structure, and applications to energy system models. *European Journal of Operational Research.* Retrieved from https://www.sciencedirect.com/science/article/pii/S0377221721005944 doi: https://doi.org/10.1016/j.ejor.2021.06.063

Rodrigues, E., Osório, G., Godina, R., Bizuayehu, A., Lujano-Rojas, J., Matias, J., & Catalão, J. (2015). Modelling and sizing of nas (sodium sulfur) battery energy storage system for extending wind power performance in crete island. *Energy*, *90*, 1606–1617.

Ruiz, P., Sgobbi, A., Nijs, W., Thiel, C., Dalla Longa, F., Kober, T., … Hengeveld, G. (2015). The jrc-eu-times model. bioenergy potentials for eu and neighbouring countries. *JRC Science for Policy Report, European Commission.*

Salvagnin, D., Roberti, R., & Fischetti, M. (2022). A fix-propagate-repair heuristic for mixed integer programming.

Sasanpour, S., & Cao, K.-K. (2022). Quantifying capacity adequacy in energy system modelling through stochastic optimization.

Schimeczek, C., Nienhaus, K., Frey, U., Sperber, E., Sarfarazi, S., Nitsch, F., … Ghazi, A. A. E.

(2023). Amiris: Agent-based market model for the investigation of renewable and integrated energy systems. *Journal of Open Source Software*, *8*(84), 5041. Retrieved from https://doi.org/10.21105/joss.05041 doi: 10.21105/joss.05041

Scholz, Y. (2012). *Renewable energy based electricity supply at low costs : development of the remix model and application for europe* (Doctoral dissertation). doi: 10.18419/OPUS-2015

Simon, S., & Xiao, M. (2022). *A multi-perspective approach for exploring the scenario space of future power systems: Input data.* https://b2share.eudat.eu. Retrieved from https://b2share.eudat.eu/records/4e5e2d11b8224fb8809cdc2d07eeff04 doi: 10.23728/B2SHARE.4E5E2D11B8224FB8809CDC2D07EEFF04

Stetter, D. (2014). *Enhancement of the remix energy system model: Global renewable energy potentials, optimized power plant siting and scenario validation* (Unpublished doctoral dissertation). Universtität Stuttgart.

Stirling, A. (1994). Diversity and ignorance in electricity supply investment: Addressing the solution rather than the problem. *Energy Policy*, *22*(3), 195 - 216. Retrieved from http://www.sciencedirect.com/science/article/pii/0301421594901597 doi: https://doi.org/10.1016/0301-4215(94)90159-7

Taibi, E., Nikolakakis, T., Gutierrez, L., Fernandez, C., Kiviluoma, J., Rissanen, S., & Lindroos, T. J. (2018). *Power system flexibility for the energy transition: Part 2, irena flextool methodology* (Tech. Rep.). International Renewable Energy Agency (IRENA).

Thrän, D., Lauer, M., Dotzauer, M., Kalcher, J., Oehmichen, K., Majer, S., … Jordan, M. (2019). Technoökonomische analyse und transformationspfade des energetischen biomassepotentials (tatbio). *Techno-Economic Analysis and Transformation Pathways of Biomass Energy Potential (TATBIO): Endbericht/Final Report.*

Trutnevyte, E. (2016). Does cost optimization approximate the real-world energy transition? *Energy*, *106*, 182–193. doi: 10.1016/j.energy.2016.03.038

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*(3-4), 279–292.

Weimer-Jehle, W. (2023). *Cross-impact balances (cib) for scenario analysis: Fundamentals and implementation.* Springer Cham. doi: https://doi.org/10.1007/978-3-031-27230-1

Wetzel, M., Ruiz, E. S. A., Schmugge, J., Rubbert, A., Witte, F., Sasanpour, S., … Gils, H. C. (in prep.). Remix: A gams-based framework for energy system optimization models. *Journal of Open Source Software.*

Wyrwa, A., Suwała, W., Pluta, M., Raczyński, M., Zyśk, J., & Tokarski, S. (2022). A new approach for coupling the short-and long-term planning models to design a pathway to carbon neutrality in a coal-based power system. *Energy*, *239*, 122438.

Yang, Y., Bremner, S., Menictas, C., & Kay, M. (2018). Battery energy storage system size determination in renewable energy systems: A review. *Renewable and Sustainable Energy Reviews*, *91*, 109–125.

Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019, November). Graph convolutional networks: a comprehensive review. *Computational Social Networks*, *6*(1), 11. Retrieved 2022-09-16, from https://doi.org/10.1186/s40649-019-0069-y doi: 10.1186/s40649-019-0069-y