

University of Stuttgart

Institute for Modelling Hydraulic and Environmental Systems

Department of Stochastic Simulation and Safety Research for Hydrosystems

Julian Eifert

Evaluating Carrier-Specific Temporal Resolutions in Multi-Energy System Optimization



Bachelor's Thesis

Evaluating Carrier-Specific Temporal Resolutions in Multi-Energy System Optimization

Submitted by

Julian Eifert

Matriculation Number: 3545368

Thesis Number: 208

Examiners: Prof. Dr.-Ing. Wolfgang Nowak

Supervisors: Dr. Karl-Kiên Cao

Institute for Modeling Hydraulic and Environmental Systems
Department of Stochastic Simulation and Safety Research for Hydrosystems
Stuttgart, 17 August 2026

Author declaration

I declare that I have developed and written the enclosed thesis completely by myself and that I have not used sources or means without declaration in the text. Any thoughts from others or literal quotations are clearly marked.

The thesis was not used in the same or in a similar version to achieve an academic grading or is being published elsewhere.

The enclosed electronic version is identical to the printed versions.

Responsible use of AI Tools (Large Language Models)

Statements	Confirmation
I am informed about the possibilities and limitations of the generative AI tools I used.	☒
I have ensured that the results provided by the generative AI tools are correct, or that I have corrected them.	☒
I acknowledge that the responsibility for the work lies with the author, not with the generative AI tools or any other person.	☒

Detailed activities for which generative AI tools (Large Language Models) were used

Activity	Description	Confirmation
Idea, Conceptualization, Methodology	• Development and adaptation of the methodology to the research question(s)	☒
Literature	• Search for relevant literature • Summaries of potentially relevant literature	☒
Coding	• Creation and documentation of code, algorithms, software • Testing and debugging of existing code, algorithms, software • Understanding of existing code, algorithms, software	☒

Activity	Description	Confirmation
Writing	• Structuring of the thesis • Refinement of user-provided bullet points and ideas into continuous prose • Critical review, feedback or revision of the content, organization or grammar of the thesis • Proofreading • Rewording or paraphrasing of text • Shortening/expanding of the text	☒
Presentation	• Structuring a presentation about the thesis • Critical review, feedback or revision of the content, organization of the presentation	☒

17.08.2026
Date

J. Gifert
Signature

Abstract

Increasing temporal, spatial, and technological detail improves the representation of interactions within multi-energy system optimization models but can lead to substantial computational requirements. At the same time, different energy carriers exhibit different characteristic time scales, raising the question of whether applying a uniformly fine temporal resolution throughout a multi-energy system is computationally necessary. Carrier-specific temporal resolution addresses this issue by retaining fine temporal detail where required while representing other carriers at coarser resolutions. Although the approach has previously demonstrated substantial computational benefits, its performance across different model structures, increasing spatial model sizes, and different choices of carrier-specific resolutions has received limited investigation.

This thesis evaluates these aspects using a scalable synthetic multi-regional benchmark model implemented in AnyMOD. Twenty instances containing between 2 and 40 regions are compared using a uniform hourly formulation, a carrier-specific configuration based on the temporal characteristics of the represented carriers, and a deliberately coarse configuration. An equivalent uniform-resolution implementation in REMix is additionally used for cross-framework validation.

The carrier-specific configuration reduces the number of non-zero matrix coefficients by approximately 50%, peak memory requirements by approximately 33%, and total runtime by approximately 54–57% relative to the uniform AnyMOD formulation. The resulting objective value lies approximately 3% below the uniform reference. The computational benefit persists as spatial model size increases, with the clearest scaling improvement occurring in model construction, whereas no systematic improvement in solver-time scaling is observed. Furthermore, reductions in mathematical problem size do not translate proportionally into reductions in computational effort. More aggressive temporal aggregation provides only limited additional problem-size reduction, does not consistently improve runtime, and produces substantially larger deviations from the uniform-resolution solution.

The results demonstrate that carrier-specific temporal resolution can provide substantial computational savings in multi-energy system optimization, while emphasizing that its effectiveness depends on the selected resolutions and the structure of the resulting formulation. Temporal detail should therefore be allocated selectively according to the characteristics and requirements of the modeled system rather than reduced as far as possible.

Acknowledgements

The author gratefully acknowledges the scientific support and HPC resources provided by the German Aerospace Center (DLR). The HPC system CARA is partially funded by "Saxon State Ministry for Economic Affairs, Labour, Energy and Climate Action" and "Federal Ministry of Research, Technology and Space".

SimTech Propaedeuticum

Chapter 2 corresponds to the Propaedeuticum required in the “Simulation Technology” degree program and may be completed in conjunction with the bachelor’s thesis.

Abbreviations

This section lists the abbreviations and acronyms used throughout this thesis.

AC	Alternating Current
AU	AnyMOD Uniform
AV	AnyMOD Variable
AVC	AnyMOD Variable Coarse
CCGT	Combined-Cycle Gas Turbine
CCGT-CHP	Combined-Cycle Gas Turbine Combined Heat and Power
CHP	Combined Heat and Power
CSV	Comma-Separated Values
DC	Direct Current
ESOM	Energy System Optimization Model
HPC	High-Performance Computing
HVDC	High-Voltage Direct Current
MES	Multi-Energy System
NNZ	Number of Non-Zero Matrix Coefficients
PV	Photovoltaic
RES	Renewable Energy Sources
RSS	Resident Set Size
RU	REMix Uniform

Contents

Abstract	i
Acknowledgements	ii
SimTech Propaedeuticum	ii
Abbreviations	iii
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	1
1.3 Research Gap	2
1.4 Objectives and Research Questions	3
1.5 Thesis Structure	3
2 Background	5
2.1 Multi-Energy Systems	5
2.1.1 Components of MES	5
2.1.2 Flexibility of MES	7
2.2 ESOMs	8
2.2.1 Mathematical structure	8
2.2.2 Drivers of complexity	9
2.3 Model Reduction Methods	10
2.3.1 Technological simplification	10
2.3.2 Spatial Aggregation	10
2.3.3 Conventional Temporal Aggregation	11
3 Variable Temporal Resolution Approach	15
3.1 Concept and Motivation	15
3.2 AnyMOD.jl: A Graph-Based Formulation	16
3.2.1 Mathematical Structure	16
3.2.2 Coupling Between Carriers	18
3.2.3 Limitations and Modeling Considerations	18
3.2.4 Comparison to Related Approaches	19
4 Methodology	21
4.1 Experimental Design	21

4.2	Benchmark Energy-System Model	22
4.2.1	Network Topology	23
4.2.2	Mathematical and Technological Scope	23
4.2.3	Input Data	25
4.3	Model Scaling and Instance Generation	25
4.3.1	Model-Size Variation	25
4.3.2	Model-Expansion Procedure	26
4.4	Temporal-Resolution Configurations	27
4.4.1	Uniform Temporal Resolution	28
4.4.2	Variable Temporal Resolution	28
4.5	Cross-Framework Model Equivalence and Validation	29
4.6	Computational Environment	29
4.6.1	Gurobi Presolve	31
4.7	Benchmark Execution Procedure	31
4.7.1	Replication and Result Aggregation	32
4.8	Evaluation Metrics	33
4.8.1	Computational Performance	33
4.8.2	Mathematical Problem Size	33
4.8.3	Objective-Value Consistency	34
5	Results	37
5.1	Objective-Value Consistency	37
5.1.1	Cross-Framework Equivalence	37
5.1.2	Effect of Temporal Resolution	38
5.2	Computational Performance	39
5.2.1	Execution and Solution Time	39
5.2.2	Memory Requirements	43
5.3	Mathematical Problem Size	44
5.3.1	Rows, Columns, and Non-Zero Coefficients	44
5.3.2	Relationship Between Problem Size and Computational Performance	44
5.4	Scaling Behavior	46
6	Discussion	49
6.1	Validity of the Benchmark Comparison	49
6.2	Computational Benefits of Variable Temporal Resolution	49
6.3	Importance of Temporal-Resolution Selection	51
6.3.1	Performance-Solution Trade-Off	52
6.3.2	Interpretation and Acceptability of Deviations	53
6.4	Interaction with Solver	53
6.5	Scaling and Framework-Level Performance	54
6.5.1	Scaling	54
6.5.2	Framework-Level Performance	56
6.6	Limitations and Generalizability	57

7 Conclusion	61
7.1 Research Questions and Main Findings	61
7.2 Implications	63
7.3 Future Work	64
List of Figures	67
List of Tables	69
Bibliography	71

1 Introduction

1.1 Motivation

The transition towards low-carbon energy systems is increasing both the scope and the level of detail required in energy-system analysis. Growing shares of variable renewable energy sources such as wind and solar power increase the importance of adequately representing short-term fluctuations in generation and demand [17, 29]. At the same time, long modeling horizons are required to assess investment pathways and structural changes occurring over several decades [23].

In parallel, the electrification of sectors such as heating, transport, and industry is strengthening the interactions between previously more isolated parts of the energy system. Multi-energy system optimization models therefore increasingly represent several energy carriers, conversion technologies, storage options, and transmission infrastructures within a common optimization framework. While this enables interactions between sectors to be considered explicitly, it also increases the size and complexity of the resulting mathematical optimization problems [37].

Temporal resolution is an important contributor to this computational complexity. Operational variables and constraints are commonly defined for every modeled time step, such that increasing temporal detail can multiply large parts of the optimization problem [12]. This creates a fundamental trade-off: sufficient temporal detail is required to represent relevant system dynamics, while excessive detail can make large-scale models computationally expensive.

However, the temporal characteristics of different energy carriers can vary substantially. Electricity requires a comparatively detailed temporal representation because supply and demand must be balanced over short time scales and renewable generation can vary rapidly. Other carriers, such as natural gas or coal, may exhibit slower dynamics or greater inherent flexibility [15]. Applying the same fine temporal resolution to all carriers may therefore introduce computational effort in parts of the model where such detail contributes comparatively little to the intended analysis.

1.2 Problem Statement

A common approach in energy-system optimization is to apply a uniform temporal resolution across the modeled energy system. In a multi-energy system, this means

that carriers with substantially different temporal characteristics are represented using the same set of time steps. If hourly resolution is required to represent electricity and variable renewable generation adequately, the same resolution may consequently also be applied to carriers whose relevant dynamics occur over longer time scales.

Carrier-specific temporal resolution provides an alternative by allowing different energy carriers to be represented at different levels of temporal detail. Fine resolution can thereby be retained for temporally sensitive parts of the system while reducing the number of time-dependent variables and constraints associated with other carriers [12, 15].

The computational effect of this reduction, however, cannot be inferred from the reduction in the number of time steps alone. Different carriers contribute unequally to the mathematical formulation, and interactions between carriers with different temporal resolutions alter the structure of the resulting optimization problem. Furthermore, reducing temporal resolution changes the constraints imposed on the modeled system and can therefore affect investment and operational results. The applicability of carrier-specific temporal resolution consequently depends on both its computational benefits and the changes it introduces in the optimization outcome.

1.3 Research Gap

Carrier-specific temporal resolution has previously been implemented in the graph-based AnyMOD framework by Göke [15]. The presented application demonstrates that assigning different temporal resolutions to individual energy carriers can substantially reduce mathematical problem size and computation time while retaining high temporal detail for electricity.

The computational benefits demonstrated for one model, however, cannot necessarily be transferred directly to other multi-energy system models. The share of variables and constraints associated with individual carriers, the technologies coupling these carriers, the representation of storage and transmission, and the overall structure of the optimization problem can differ considerably between applications. The extent to which the reported performance benefits can be reproduced for a different model structure therefore remains uncertain.

In addition, the effect of increasing model size on the benefits of carrier-specific temporal resolution has not been established systematically. It is unclear whether the relative computational advantage remains comparable, increases, or decreases as additional spatial regions and their associated technologies, demands, and network connections are introduced.

Finally, the computational benefit depends on the temporal resolutions assigned to the individual carriers. More aggressive aggregation may further reduce mathematical problem size, but it can also alter the resulting system representation more strongly and

does not necessarily lead to proportional reductions in computation time. A systematic comparison should therefore consider not only whether variable temporal resolution improves performance, but also how the selected resolutions affect both computational behavior and deviations from a uniform-resolution reference.

1.4 Objectives and Research Questions

The objective of this work is to evaluate the computational benefits and limitations of carrier-specific temporal resolution for multi-energy system optimization beyond the model originally used to demonstrate the approach. For this purpose, a scalable multi-regional benchmark model is implemented with both uniform and carrier-specific temporal resolutions and evaluated over a range of model sizes. In addition to computational performance and mathematical problem size, the resulting objective values are compared to assess how the selected temporal resolutions affect the optimization outcome.

The investigation is guided by the following research questions:

1. How does carrier-specific temporal resolution affect the computational performance and mathematical problem size of a multi-energy system optimization model compared with uniform temporal resolution?
2. How do the computational benefits of carrier-specific temporal resolution change with increasing model size?
3. How does the degree of carrier-specific temporal aggregation affect the trade-off between computational performance and deviations from the uniform-resolution solution?

1.5 Thesis Structure

The remainder of this thesis is structured as follows. Chapter 2 introduces the relevant background on multi-energy systems, energy system optimization models, and methods for reducing their computational complexity, with a particular focus on temporal aggregation. Chapter 3 presents the concept of carrier-specific temporal resolution and its graph-based implementation in AnyMOD. Chapter 4 describes the benchmark model, temporal-resolution configurations, computational environment, and experimental procedure used in the evaluation. Chapter 5 presents the results regarding objective-value deviations, computational performance, mathematical problem size, and scaling behavior. Chapter 6 discusses these findings with respect to the selection of temporal resolutions, solver behavior, generalizability, and the limitations of the approach. Finally, Chapter 7 summarizes the main conclusions and identifies directions for future research.

2 Background

2.1 Multi-Energy Systems

In the past, energy system analysis often treated different sectors such as electricity, gas and heating as independent systems [13]. In recent years, a number of factors have emphasized the need for cross-sector integration and contributed to an increasing research interest in multi-energy systems (MES). The increasing penetration of renewable energy sources and the resulting fluctuations in energy supply motivate stronger coupling with energy carriers and infrastructures that can provide additional flexibility, such as gas networks [10]. This, in combination with the electrification of the heating and transport sectors, which were traditionally driven by fossil fuels, highlights the growing interactions between different energy sectors and carriers [37]. By assessing both benefits and constraints that arise from these cross-sectoral interactions, MES aim to address challenges arising from the transformation towards energy systems of the future [25].

While individual technologies and small-scale configurations can often be investigated experimentally, questions concerning the long-term development of entire energy systems must generally be addressed through models. Such analyses involve system configurations that do not yet exist and planning horizons that may span several decades. Models therefore provide simplified representations of reality that capture the system characteristics relevant to a given research question while abstracting from less relevant details.

For MES, this requires a representation in which the coupling between carriers is part of the model itself rather than an assumption imposed on it. A MES can accordingly be understood as an integrated energy system in which multiple energy carriers, together with the infrastructures and conversion technologies linking them, are considered within a common modeling framework [25]. Depending on the context, the spatial scale of such systems ranges from individual buildings or districts to entire nations or continents, which may be subdivided into multiple regions.

2.1.1 Components of MES

To gain a further understanding of MES, a look at its key components is warranted. In the following, these are exemplified through a simplified regional multi-carrier energy system centered on a combined heat and power (CHP) plant. The CHP plant supplies electricity and usable heat to meet regional demand and thereby links the natural-gas, electricity, and heat sectors within the system.

Energy carriers

The presence of multiple energy carriers is a defining characteristic of MES. Energy carriers are forms of energy that can be transported, converted, and used to provide energy services, such as electricity, natural gas, hydrogen, heat, synthetic fuels, or biomass. Depending on the system specifics, a carrier may enter the system as an input, be transferred between different parts of the system, be produced through conversion, or, where technically feasible, be stored for later use [26].

In the example system, natural gas enters as an input to the CHP plant and is converted into electricity and usable heat, which are subsequently used to meet regional demand [2]. The three carriers are therefore directly connected through the operation of the CHP plant.

External Demand

Energy demand is a key element of MES, as the main purpose of the system is to provide energy quantities or services. In the example system, this is represented by the regional demand for electricity and usable heat. More generally, energy demands commonly include electricity and heat, but may also extend to cooling, hot water, hydrogen, industrial process heat, or even transport [2, 25].

A central characteristic of MES is that a given demand can be satisfied through multiple different energy carriers and conversion technologies. In the example system, heat demand may be supplied by the CHP plant, but additional technologies such as a gas boiler or an electric heat pump could provide alternative supply pathways. Similarly, household heat demand may generally be fulfilled by a gas boiler, an electric heat pump, a district-heating network, or heat produced by a CHP unit. This substitutability introduces additional flexibility and allows the system to satisfy demand using an optimal combination of technologies under the given conditions, exploiting interactions between different sectors.

Conversion technologies

The conversion between different energy carriers is enabled by dedicated conversion technologies. In the example system, the CHP plant constitutes the central conversion technology. This could, for example, be a combined-cycle gas turbine (CCGT) operated in CHP mode, converting the chemical energy of natural gas into both electricity and usable heat [36]. Additional conversion technologies, such as gas boilers or heat pumps, could provide alternative pathways for satisfying regional heat demand. Beyond technologies that convert one carrier into another, MES can also include renewable generation technologies such as wind turbines and photovoltaic systems (PV). These convert primary renewable energy, in this case wind and solar irradiance, into electricity without consuming another energy carrier.

Storage technologies

Daily and seasonal fluctuations in energy supply and demand necessitate the inclusion of technologies capable of storing energy at one point in time, retaining it, and releasing it at a later time with associated conversion and storage losses. In the example system, electricity generated by the CHP plant or renewable technologies could be stored temporarily in batteries, while natural gas could be stored over longer periods. Typical examples of long-term storage include gas caverns used to meet higher demand during colder seasons. Systems with a high share of renewable energy sources (RES) are particularly dependent on short-term storage technologies such as pumped hydro or batteries to buffer variability inherent in RES generation [26].

Transmission infrastructure

Depending on the spatial scope, MES can include networks that transport energy carriers between regions. In the example system, the electricity transmission grid could transport electricity produced by the CHP plant to neighboring regions, while a gas pipeline network could supply the natural gas required by the CHP plant or exchange gas between regions. Electricity grids and gas pipelines are among the most common forms of interregional energy infrastructure, but other energy carriers may also be transported depending on the system considered, for example through district-heating networks, coal transport infrastructure, or networks for hydrogen and other fuels.

Traditionally, the networks of individual carriers were planned and operated largely independently, using domain-specific representations such as AC or DC power flow for electricity, the Weymouth or Panhandle equations for gas, and thermal-hydraulic models for district heating [28]. The MES approach instead integrates the transmission networks of the different carriers together with the conversion technologies that couple them. In the example system, this means that electricity and gas networks are considered jointly with the CHP plant and other conversion technologies, thereby exploiting cross-sectoral synergies and assessing the system as an interconnected whole [28, 30].

2.1.2 Flexibility of MES

It is important to highlight that the MES concept is not limited to a fixed set of components. Depending on the context and purpose of the analysis, a model may include any relevant combination of energy carriers, storage and conversion technologies, network infrastructures, and demand profiles. Its level of technological, spatial, and temporal detail can therefore be adapted to the specific characteristics of the system under investigation. This flexibility allows MES models to be applied to a wide range of contexts, from individual buildings and industrial sites [13, 25] to regional, national, or continental energy systems [25, 35].

2.2 ESOMs

While the concept of MES describes the structure and interactions of the system under investigation, Energy System Optimization Models (ESOMs) provide a quantitative framework for analyzing its design and operation. Baader et al. (2023, p.2) [5] summarize their purpose as follows: "In a nutshell, ESOMs are mathematical descriptions of energy systems that identify optimal investment and operation strategies to meet given cost or decarbonization targets."

A complementary definition is provided by DeCarolus et al. (2017, p.185) [9]: "ESOMs include detailed, bottom-up technology specifications and utilize linear programming techniques to minimize the system-wide present cost of energy provision by optimizing the installation of energy technology capacity and its utilization."

Several key characteristics of ESOMs emerge from these definitions. First, ESOMs are mathematical representations of energy systems, typically using a bottom-up description of relevant energy carriers, conversion technologies, storage units, networks, and demands. Mirroring the flexibility of the MES concept, the level of technological, spatial, and temporal detail may vary considerably depending on the purpose and scope of the analysis.

Second, ESOMs identify an optimal system configuration according to a specified objective. In many applications, this objective is the minimization of total system cost. However, models may also represent a number of other objectives including environmental, social, or political factors, provided they can be expressed through suitable quantitative indicators. Previously used examples include greenhouse-gas emissions, political support, societal acceptance [3], or even security of supply [24]. Multiple objectives can be considered simultaneously through multi-objective optimization or incorporated through additional constraints.

The decisions optimized in ESOMs can broadly be divided into capacity-expansion decisions, also referred to as investment or planning decisions, and dispatch, or operational, decisions [29]. Capacity expansion decisions determine which technologies should be installed, when and where they should be installed, and at what capacity. Operational decisions determine how the available technologies should be dispatched to satisfy energy and service demands under the given technical and resource constraints.

The resulting optimization problem is commonly formulated and solved using linear programming, mixed-integer linear programming, or related mathematical optimization techniques. The choice of formulation affects both the types of system characteristics that can be represented and the computational effort required to solve the model.

2.2.1 Mathematical structure

To understand the computational complexity of ESOMs, it is useful to briefly consider their mathematical structure. In a generic formulation, investment decisions are represented by variables describing, for example, the installed capacity of a technology

in a given region and planning period, while operational variables describe its production, fuel use, storage activity, or energy flows in each time step. These variables are linked through constraints such as energy-balance equations, capacity limits, conversion efficiencies, storage balances, resource availability, and emission limits. Detailed mathematical formulations of these components can be found in Miehl et al. [27] and Howells et al. [20].

As variables and constraints are created for all technologies, energy carriers, regions, time steps, and investment periods, increasing the scope or resolution of any of these dimensions can substantially enlarge the resulting optimization problem.

2.2.2 Drivers of complexity

Owing to their broad analytical scope and continuing efforts to represent real-world energy systems more accurately, ESOMs have become increasingly large and complex [23]. Long modeling horizons are often required to explore energy-system transformation pathways and assess the long-term effects of investment and policy decisions [23]. As DeCarolis et al. [9] observe, additional model features are frequently introduced to improve the realism of internal system dynamics. Expanding the technological and sectoral scope, increasing spatial and temporal resolution, representing uncertainty through multiple scenarios, and adding operational, environmental, or policy-related constraints all increase the number of variables and constraints in the resulting optimization problem. Computational difficulty may be further increased by intertemporal relationships, such as storage balances or emission limits, as well as by the use of integer variables or nonlinear formulations. The complexity of an ESOM is therefore determined not only by its size, but also by its mathematical problem class and the extent to which its variables and constraints are interconnected [23].

Among these factors, temporal discretization is often the dominant contributor to model size and computational cost [12]. Most operational variables and constraints, including energy and storage balances, and inter-regional flows, must be defined for every represented time step. Consequently, increasing the temporal resolution or extending the modeled time horizon can multiply large parts of the optimization problem. At the same time, sufficient temporal detail becomes increasingly important as the share of variable renewable energy sources rises. Unlike conventional dispatchable generation, wind and solar output is weather-dependent and may vary considerably across hours, days, and seasons. Future systems may additionally rely more strongly on the active management of demand, storage, and sector-coupling technologies, whose operation can only be represented adequately when the relevant temporal dynamics are resolved [17, 29].

Insufficient temporal resolution can smooth fluctuations and extreme events in both renewable generation and demand, thereby distorting estimates of the flexibility, storage, and dispatchable capacity required by the system. In particular, models that do not capture the full variability of supply and demand may overestimate the share of demand that can be met by fluctuating renewable sources [17]. Spatial resolution is similarly

important because renewable-resource availability, capacity factors, generation costs, demand patterns, and network constraints differ between locations [11]. Moreover, the spatial distribution of renewable generation determines the extent to which regional fluctuations can balance one another [29]. ESOM development therefore involves a fundamental trade-off between representing the energy system with sufficient detail to obtain meaningful results and limiting model complexity to maintain computational tractability.

To manage this trade-off, a range of model-reduction methods has been developed that aim to reduce the size or complexity of ESOMs while preserving the system characteristics most relevant to the analysis.

2.3 Model Reduction Methods

2.3.1 Technological simplification

Technological simplification reduces model complexity by limiting both the diversity of represented technologies and the level of detail with which their behavior is described. At a higher level of abstraction, individual plants or closely related technology variants may be grouped into representative technology classes, while distinctions between parameters like rated power or conversion efficiency may be omitted when they are not relevant to the research question [8]. This reduces the number of technology-specific investment and operational variables and constraints. Further simplification can be achieved by reducing the operational detail of individual technologies. For example, part-load-dependent efficiencies may be replaced by constant conversion efficiencies, while minimum part-load requirements, start-up and shut-down costs, minimum up- and down-times, and ramping restrictions may be neglected [23]. Individual generating units may likewise be represented as an aggregated fleet, and discrete unit additions or commitment decisions may be replaced by continuous capacity and dispatch variables, avoiding integer variables. Similar choices arise for storage technologies, for which simplified formulations may assume constant charging and discharging efficiencies and neglect degradation, state-dependent losses, or detailed operating restrictions.

The representation of transmission and transport technologies can also vary substantially. Simple capacity-constrained transport models may represent energy flows without considering the underlying physical network behavior, whereas more detailed formulations may, for example, include nonlinear power-flow equations [8].

2.3.2 Spatial Aggregation

Spatial aggregation reduces model complexity by representing the energy system through a simplified network. Simplification may involve reducing the number of geographical

regions, the number of links connecting them, or both. Regions may be grouped according to geographical proximity, administrative boundaries, similarities in demand and renewable-resource profiles, or characteristics of the underlying transmission network [23]. The resulting aggregated regions must then be represented through suitable parameters. Some quantities, such as demand or installed capacity, are commonly summed [8], whereas others may require weighted averaging or other more advanced aggregation functions [23].

Spatial aggregation also requires a simplified representation of the connections between the resulting regions. Internal transmission links and network constraints within an aggregated region are commonly omitted [8], while the remaining inter-regional connections are often combined into representative transfer capacities and distances. Although this substantially reduces the number of flow variables and network constraints, it can conceal internal bottlenecks and imply that energy can be transferred freely between locations contained within the same region [11]. Aggregating locations with different renewable-resource availability, demand patterns, technology capacities, or costs may also affect the estimated geographical distribution of investments and operation.

Increasing the level of spatial and technological abstraction can therefore reduce the number and connectivity of variables and constraints and may allow detailed formulations to be replaced by less computationally demanding representations. However, these performance improvements come at the cost of reduced spatial and technological detail and a less accurate representation of regional heterogeneity and network limitations. The appropriate degree of simplification must consequently be selected according to the research question and validated to ensure that omitted details do not negatively affect the resulting investment and operational decisions.

2.3.3 Conventional Temporal Aggregation

Temporal aggregation reduces the computational complexity of ESOMs by simplifying the time-dependent input data used to represent system operation, such as demand profiles or renewable-energy availability. By replacing the original time series with a temporally reduced representation, the number of time steps for which operational variables and constraints must be defined is decreased [32]. This can be achieved either by combining consecutive observations into time steps of lower resolution or by replacing the original chronological time series with a smaller set of typical time steps or periods. The resulting reduction in model size can substantially improve computational performance, but it may also remove temporal variations, extreme events, correlations between input profiles, or chronological relationships that are relevant to system design and operation. Temporal-aggregation methods therefore differ not only in the number of time steps retained, but also in the characteristics of the original input time series that they preserve [18].

Downsampling

Downsampling reduces temporal resolution by combining a fixed number of consecutive time steps into a single time step. The values of the original observations are commonly replaced by their mean, although other aggregation functions may be used depending on the represented quantity [18]. For example, four consecutive hourly observations may be combined into one four-hour time step. Because all resulting time steps have the same duration and remain in their original order, downsampling preserves the overall chronological sequence of the time series. However, averaging smooths temporal variations and may suppress demand peaks and renewable-generation extremes. Consequently, capacity and cost of RES, which are particularly subject to intermittency, are often underestimated [18, 23].

Segmentation

Segmentation also combines adjacent time steps, but allows the resulting time steps to have different durations. Periods in which the time series changes only slightly can be represented by longer segments, whereas periods containing large gradients, peaks, or other relevant variations can be retained at a finer resolution [23]. Each segment is assigned a duration corresponding to the number of original time steps it represents. In contrast to uniform downsampling, segmentation can therefore allocate more temporal detail selectively while preserving the chronological order of the original observations. In the example examined by Hoffmann et al. [18], the intra-daily variation of a photovoltaic generation profile was represented considerably more accurately using 1,460 irregular segments than through uniform downsampling to the same number of time steps. Nevertheless, segmentation still leads to a loss of intra-time step variability, strongly affecting model results for systems with a high share of RES.

Typical Periods

Typical period methods are based on the assumption that an energy system responds similarly to recurring combinations of external conditions, such as energy demand, wind availability, or solar irradiance [18]. Rather than repeatedly modeling many periods with comparable input profiles, these periods are grouped and represented by a smaller set of typical periods [18, 19]. This reduces the number of operational variables and constraints while retaining the main recurring patterns of the original time series.

To construct typical periods, the original time series is generally divided into periods of a predefined length. Similar periods are subsequently grouped, and one typical period is selected or constructed for each group. Typical periods may be selected manually, using predefined calendar-based categories, statistical sampling, or clustering algorithms. Clustering approaches group periods according to similarities in input profiles. Common algorithms include k-means, k-medoids, and hierarchical clustering [18]. The specific implementation of typical period methods varies with the approach employed, including

the metric used to quantify similarity, the clustering or selection algorithm, and the procedure used to select or construct the representative period. Regardless of the method, each typical period receives a weight indicating how frequently the corresponding group occurs in the original time series.

The choice of the number and duration of typical periods involves a trade-off between accuracy and computational performance. A larger number of typical periods generally reproduces the original time series more accurately but provides a smaller reduction in model size [23, 32]. It is important to note that the appropriate number cannot be determined solely from statistical measures of clustering quality. It should also be evaluated according to its effect on relevant model outputs, including total system cost, installed capacities, renewable curtailment, or storage requirements [19]. As a result, the appropriate number of periods is still subject to research [23].

Typical period methods are subject to broader limitations concerning the representation of extreme events, chronology, and intertemporal processes, which are discussed in the following subsection.

Limitations and Challenges of Temporal Aggregation

Although temporal aggregation can substantially reduce the computational requirements of ESOMs, it inevitably removes part of the information contained in the original time series and may therefore affect both investment and operational results. The type of information lost depends on the selected method. Downsampling and segmentation preserve the original chronological sequence but smooth variations within the aggregated time steps. Typical-period methods can achieve greater reductions in model size, but generally preserve chronology only within each typical period and not between them.

A central challenge is the representation of variable renewable energy sources. Downsampling, segmentation, and typical periods may smooth fluctuations and generation peaks inherent to RES. This may distort the residual load that must be met by storage, transmission, or dispatchable generation and consequently affect estimates of renewable capacity, curtailment, flexibility requirements, and total system cost. Rare but system-critical conditions, such as peak-demand periods or prolonged periods of low wind and solar generation, may also be omitted because aggregation and clustering methods primarily represent frequently occurring conditions. Such extreme periods may therefore need to be included explicitly [22, 32].

The treatment of chronology is particularly important for intertemporal constraints. Typical days retain the sequence of their internal time steps but generally lose their original ordering within the year. Consequently, intertemporal components, such as storage systems, which link consecutive periods, may not be properly modeled [18]. While short-duration storage may still be represented within a typical day, this is particularly problematic for long-term and seasonal storage, which is especially important for energy systems with a high share of RES [18]. Linking formulations or reconstructed chronological sequences can mitigate this limitation, but generally require additional

2 Background

inter-period state variables and linking constraints, thereby partly offsetting the computational advantages of temporal aggregation [7, 32].

Finally, a close statistical approximation of the original input time series does not necessarily ensure accurate optimization results. Temporal-aggregation methods should therefore be assessed according to their effects on relevant model outputs, including total system cost, installed capacities, storage requirements, renewable curtailment, and unmet demand. Where computationally feasible, the system configuration obtained from the aggregated model should additionally be evaluated against the original full-resolution time series [19, 32]. Temporal aggregation thus involves a fundamental trade-off between computational performance and the preservation of temporal information relevant to the representation of the energy system. The suitability of a particular aggregation method consequently depends on which temporal characteristics must be retained for the system and research question under investigation [18, 19].

3 Variable Temporal Resolution Approach

As discussed in Section 2.3.3, conventional temporal-aggregation methods generally reduce the number of time steps uniformly across the entire energy system. Consequently, the same temporal resolution is typically applied to all energy carriers, technologies, and associated operational variables, even though their temporal characteristics may differ substantially [12]. This chapter introduces a variable temporal resolution approach that allows different energy carriers to be represented at different levels of temporal detail. This approach is formulated by Göke and implemented within the AnyMOD.jl modeling framework [14] and forms the methodological basis of the present work.

3.1 Concept and Motivation

The central concept of the variable temporal resolution approach is to assign each energy carrier an individual temporal resolution. This makes it possible to account for differences in the physical characteristics and operational dynamics associated with each energy carrier.

Electricity, for example, generally requires a comparatively high temporal resolution. Electricity supply and demand must remain balanced over short time intervals, while generation from variable renewable energy sources such as wind and solar power can change considerably within a single day. A sufficiently detailed temporal representation is therefore necessary to capture fluctuations in renewable generation as well as the operation of short-term storage technologies and dispatchable generation capacities used to compensate for these fluctuations [17, 29].

Other energy carriers, such as natural gas or coal, may often be represented at a lower temporal resolution without losing the information relevant to long-term system planning. Their supply, transport, and consumption patterns often vary more slowly and are thus treated with less temporal detail [15]. Applying the same high temporal resolution used for electricity to these carriers can therefore introduce additional variables and constraints without providing a corresponding improvement in model accuracy.

Carrier-specific temporal resolutions can also reflect the physical flexibility inherent in certain energy infrastructures. Gas networks, for instance, can temporarily store gas through variations in pipeline pressure, a phenomenon referred to as linepack. This

provides short-term flexibility and reduces the need for gas inflows and withdrawals to be balanced instantaneously [4, 28]. Similarly, heat systems exhibit thermal inertia and storage capacity at several levels. The thermal mass of buildings can absorb and release heat, allowing heat delivery to be varied over limited periods while indoor-temperature requirements remain satisfied. District-heating networks can also store thermal energy in the water and pipe mass and introduce delays between heat production and consumption [33]. These properties allow certain carriers to be represented at a lower temporal resolution while preserving sufficient model accuracy.

Electricity networks do not possess comparable inherent storage capabilities. Unless dedicated storage technologies are included, electricity generation and consumption must be balanced almost instantaneously. This provides an additional motivation for representing electricity at a higher temporal resolution than carriers whose infrastructures inherently provide greater temporal flexibility.

The variable temporal resolution approach consequently retains temporal detail where it is necessary while allowing greater abstraction where short-term dynamics are less relevant. In this way, it seeks to reduce the computational complexity of energy system optimization models while preserving the carrier-specific temporal characteristics that influence system design and operation.

3.2 AnyMOD.jl: A Graph-Based Formulation

AnyMOD.jl (or AnyMOD) by Leonard Göke is an energy system modeling framework that implements a graph-based formulation for the representation and optimization of multi-energy systems. In contrast to approaches based on independent representative periods, AnyMOD retains continuous chronological time series. At the same time, it allows the temporal and spatial granularity of the model to be defined individually for each energy carrier. AnyMOD is developed in Julia and utilizes its mathematical optimization package JuMP.

A defining feature of AnyMOD is that its principal model sets, including regions, technologies, carriers, and time steps, are organized as rooted trees, making it graph-based. This hierarchical organization makes it possible to represent different levels of aggregation within a common model structure. The relationships between these levels are retained through the parent-child structure of the corresponding tree.

3.2.1 Mathematical Structure

The general mathematical structure of AnyMOD follows that of conventional energy system optimization models. The model consists of decision variables, technical and operational constraints, carrier balances, and an objective function, which commonly minimizes total system costs. Its distinguishing feature is therefore not the general optimization structure, but the hierarchical organization of the underlying sets.

Each set is represented as a rooted tree, which is a connected acyclic graph with one designated root node. Nodes at higher levels represent more aggregated elements, while their child nodes represent subdivisions at a finer level of detail. In the temporal tree, for example, a year may be divided into days, each day into four-hour intervals, and each four-hour interval into individual hours.

In the present work, only the temporal dimension of this graph-based formulation is considered. The temporal tree determines which time steps are available in the model and how time steps at different levels of resolution are related. A parent time step represents the complete interval covered by its children, while the child time steps form a non-overlapping partition of that parent interval.

The tree structure restricts the temporal resolutions that can be represented. In AnyMOD, the temporal resolution of each carrier is defined by assigning it to a fixed level of the time-step tree. Within a superordinate dispatch period, the time steps at a given level are assigned a common temporal scaling factor based on the number of time steps at that level. Consequently, arbitrary combinations of differently sized intervals at the same level, such as one eight-hour time step followed by sixteen one-hour time steps, cannot be represented directly [15].

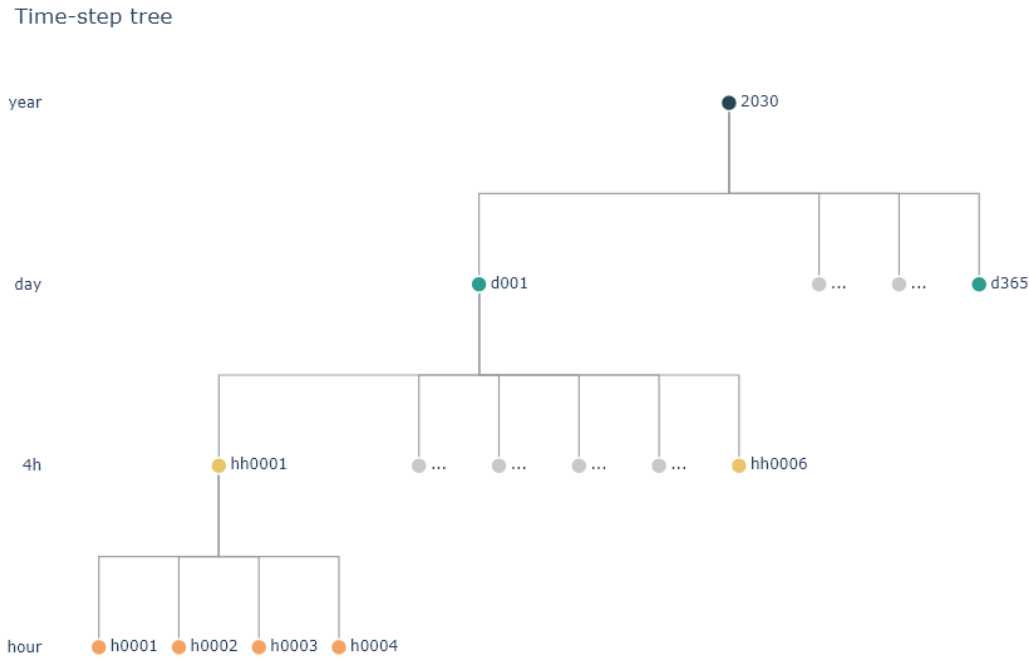


Figure 3.1: Reduced rooted tree of time steps

Figure 3.1 illustrates a reduced temporal tree. The optimization horizon consists of one year, which is divided into 365 days. Each day is subdivided into six four-hour time steps, and each four-hour time step is further divided into four one-hour time steps.

3 Variable Temporal Resolution Approach

This constitutes a permissible temporal tree because the children at each level have an equal duration and collectively cover the full interval represented by their parent.

The tree does not imply that every model variable must be defined at the lowest available level. Instead, variables associated with different carriers may be placed at different levels of the same temporal hierarchy. Electricity-related variables may, for instance, be defined at the hourly level, whereas gas-related variables may be defined at the four-hour or daily level. In this way, the tree provides the common temporal structure required to combine carrier-specific resolutions within one optimization model.

3.2.2 Coupling Between Carriers

A central challenge arises when a conversion technology links energy carriers represented at different temporal resolutions. In a conventional model with a uniform temporal structure, the input and output flows of a conversion technology can be related within each common time step. With carrier-specific temporal resolutions, however, the corresponding time steps may differ in both number and duration.

AnyMOD resolves this issue by formulating the conversion relationship at the least detailed, and therefore coarsest, temporal resolution among the carriers involved. Flows associated with a more finely resolved carrier are aggregated to the temporal level of the more coarsely resolved carrier before the conversion constraint is evaluated. In AnyMOD, temporal aggregation between levels is performed by summation. This is appropriate for variables that represent quantities accumulated over a time interval, such as produced, consumed, stored, or converted energy. The resulting aggregated quantity can then be used in the conversion equation at the coarser temporal level.

Consider, for example, a gas-fired power plant that converts natural gas into electricity. If electricity is represented at an hourly resolution while natural gas is represented using four-hour time steps, the conversion constraint is formulated for each four-hour interval. The four hourly electricity outputs contained within that interval are summed and related to the gas input defined for the corresponding four-hour time step.

This formulation preserves consistency between carriers with different temporal resolutions. At the same time, it means that the conversion constraint only enforces the relationship between input and output over the coarser interval.

3.2.3 Limitations and Modeling Considerations

Although the formulation presented by Göke permits different energy carriers to be represented at different temporal resolutions, the temporal structure is not fully flexible. All time steps are organized within a common rooted tree, and each carrier must be assigned to one of its predefined depth levels. In the absence of a weighting scheme for aggregating time steps of unequal duration, all time steps at a given level must follow

a uniform temporal resolution. Consequently, the resolution assigned to an individual carrier cannot vary over the course of the modeled horizon.

Most importantly, the resolution assigned to a carrier constitutes an assumption about its inherent flexibility. When a carrier is represented using longer time steps, variations and operational restrictions within these intervals are not resolved. In interactions between carriers with different resolutions, the finer flows are aggregated to the coarser resolution of the conversion balance. This effectively assumes that the coarsely represented carrier can accommodate variations occurring within the aggregated interval. While this may be appropriate for flexible carriers such as gas, an excessively coarse resolution may overstate flexibility and affect dispatch, storage operation, capacity expansion, and system costs. Conversely, using unnecessarily fine resolution reduces the computational benefits of the approach. The selected resolutions should therefore be justified according to the physical characteristics of each carrier and evaluated individually for each model.

Finally, a reduction in the number of variables, constraints, and non-zero matrix elements does not necessarily result in a proportional reduction in solution time. Computational performance also depends on the structure of the resulting optimization matrix and the solver algorithm employed. The benefits of variable temporal resolution should consequently be assessed using both measures of model size and observed computational performance.

3.2.4 Comparison to Related Approaches

Similar approaches to varying temporal resolution across different parts of an energy-system model are implemented in SpineOpt (Ihlemann et al. [21]) and proposed by Gao et al. [12]. SpineOpt uses temporal blocks that can be assigned separately to different sectors, regions, technologies, and decision-variable types [21]. This allows, for example, electricity flows to be represented at a finer resolution than gas flows, or unit-commitment decisions at a coarser resolution than the corresponding energy flows. The resolution may also vary over the modeling horizon. When components with different resolutions interact, variables defined at the finer resolution are aggregated to the coarser resolution of the linking constraint using relationships between overlapping time slices. In the electricity-gas case study presented by Ihlemann et al., selectively reducing temporal resolution decreased computation time substantially while causing only minor changes in total system cost.

The formulation proposed by Gao et al. [12] generalizes this principle further. Rather than assigning temporal resolution primarily by carrier or model component, each decision variable and constraint may be defined at an individual temporal resolution. This enables temporal detail to be removed selectively from parts of the mathematical formulation where it contributes little to solution quality, while retaining high resolution for variables and constraints that depend strongly on short-term dynamics. Applied to

3 Variable Temporal Resolution Approach

an interconnected European energy-system model, the approach improves the trade-off between computational performance and solution accuracy.

Compared with these approaches, the original AnyMOD formulation assigns temporal resolution at the level of energy carriers through predefined depths of a common time-step tree [15]. SpineOpt provides greater flexibility by assigning temporal blocks to individual model components and allowing resolution to vary over time, while the formulation of Gao et al. offers the most general approach by defining temporal resolution separately for every variable and constraint. The approaches therefore share the objective of applying high temporal detail only where required, but differ in the granularity and flexibility with which temporal resolution can be assigned.

4 Methodology

4.1 Experimental Design

The experimental design aims to evaluate the applicability, solution consistency, and computational performance of the variable temporal resolution approach implemented in AnyMOD. While the original formulation was demonstrated by Göke [15] using a specific energy-system model, the present study applies the approach to a newly developed benchmark model with a different system structure. This allows the transferability of the formulation to be assessed beyond the model originally used to introduce it.

The analysis addresses three main aspects. First, the implementation of the benchmark model in AnyMOD is compared with an equivalent implementation in the established REMix modeling framework [35]. For this comparison, both frameworks use a uniform temporal resolution. The purpose of the cross-framework comparison is to verify that the benchmark model produces consistent optimization results despite differences in the internal formulations and implementations of the two frameworks.

Second, the effect of carrier-specific temporal resolution is investigated within AnyMOD. The uniform-resolution AnyMOD model (or AU) serves as the reference configuration against which the variable-resolution configurations are evaluated. The analysis examines whether reducing the temporal resolution of selected energy carriers decreases the size of the resulting optimization problem and improves computational performance while preserving the relevant model results. This reflects the central objective of the approach: to retain detailed temporal representation where it is required while avoiding unnecessary operational variables and constraints for carriers characterized by slower dynamics.

Two variable-resolution configurations are considered. In the first, the temporal resolution assigned to each carrier is selected according to its physical and operational characteristics. This configuration represents the intended application of the approach and will be referred to as AV. In the second, deliberately coarse temporal resolutions are assigned to selected carriers. This configuration constitutes a stress test intended to illustrate the consequences of excessive temporal aggregation and to examine the trade-off between computational performance and solution consistency. This configuration will be referred to as AVC. Since only these two predefined cases are investigated, the comparison is not treated as a comprehensive sensitivity analysis. The deliberately coarse configuration is particularly relevant because an insufficient temporal resolution can overstate carrier flexibility and affect dispatch, storage operation, capacity expansion, and total system costs.

Third, the scaling behavior of the frameworks and temporal configurations is examined by gradually increasing the spatial size of the benchmark model. A total of 20 model instances are generated, ranging from 2 to 40 regions in increments of two:

$$N_{\text{regions}} \in \{2, 4, 6, \dots, 40\}. \quad (4.1)$$

Applying the same benchmark configurations to models of increasing size makes it possible to assess how computational time, memory requirements, and mathematical problem size develop as the number of regions grows. The evaluation therefore considers both direct measures of computational performance and structural measures such as the number of variables, constraints, and non-zero matrix entries. This distinction is important because a reduction in mathematical problem size does not necessarily lead to a proportional reduction in solution time.

Table 4.1 summarizes the four configurations considered in the benchmark.

Table 4.1: Overview of the benchmark configurations

Configuration	Framework	Temp. representation	Role in the analysis
AU	AnyMOD	Uniform	Reference configuration
AV	AnyMOD	Carrier-specific	Evaluation of the proposed approach
AVC	AnyMOD	Carrier-specific, coarse	Stress test
RU	REMix	Uniform	Cross-framework validation

The comparison between the uniform AnyMOD (AU) and REMix (RU) configurations primarily evaluates cross-framework consistency. In contrast, the comparison among the three AnyMOD configurations isolates the effect of changing temporal resolution while keeping the modeling framework and the remaining model assumptions constant. Across AU, AV, and AVC, all model data, technologies, network topology, constraints, and solver settings remain unchanged. Only the temporal resolution assigned to individual carriers is varied. Together, these comparisons allow the proposed approach to be evaluated with respect to solution consistency, computational performance, and scaling behavior.

4.2 Benchmark Energy-System Model

The computational experiments are based on a synthetic multi-energy system model developed specifically for the benchmark. The model represents the joint planning and operation of electricity, heat, natural gas, and coal across a variable number of interconnected regions. Its structure is kept sufficiently general to investigate the computational effects of model size and temporal resolution without reproducing one particular real-world energy system.

4.2.1 Network Topology

The benchmark instances contain between 2 and 40 regions connected through an inter-regional network. Constructing a consistent set of differently sized models from real-world energy systems would require suitable datasets for every model size and could introduce structural differences unrelated to the benchmark objective. A synthetic network topology is therefore used to enable the systematic and automated scaling of the model while retaining a common construction principle across all instances.

The regional connections are generated using Delaunay triangulation. For a given set of regional nodes, the triangulation creates links between neighboring nodes while avoiding unnecessarily dense connectivity. Delaunay triangulation has also been applied in the construction of synthetic power grids designed to reproduce selected statistical characteristics of real-world networks [6]. In the present benchmark, it provides a reproducible method for generating increasingly large but structurally comparable network topologies. Further details on the generation of model instances with different numbers of regions and the application of the triangulation algorithm are provided in Section 4.3.2.

Each resulting network link may accommodate electricity transmission through high-voltage direct-current (HVDC) connections and natural-gas transport through methane pipelines. The same underlying topology is therefore used for both carrier networks, while the corresponding transmission technologies and flows are represented separately in the energy-system model.

4.2.2 Mathematical and Technological Scope

The benchmark model combines capacity-expansion and operational optimization. Investment variables determine the capacities of generation, conversion, storage, and transmission technologies, while dispatch variables describe their operation throughout the modeled period. All investment and operational variables are continuous, such that the benchmark is formulated as a linear programming problem. The optimization horizon comprises one year represented by 8,760 consecutive hourly time steps. This full-resolution temporal representation serves as the basis from which the carrier-specific temporal configurations described in Section 4.4 are derived.

The system is driven by exogenously specified electricity and heat demands in each region. These demands must be satisfied through the available supply, conversion, storage, and transmission options. Four energy carriers are represented: electricity, heat, natural gas, and coal. Their interactions allow electricity and heat demand to be met through multiple technological pathways and introduce coupling between the different parts of the system.

Electricity may be generated from photovoltaic systems, wind turbines, coal-fired power plants, and combined-cycle gas turbines (CCGTs). Heat may be supplied by gas boilers, electric heat pumps, and combined heat and power plants based on CCGT technology (CCGT-CHP). The latter jointly produce electricity and heat from natural gas and

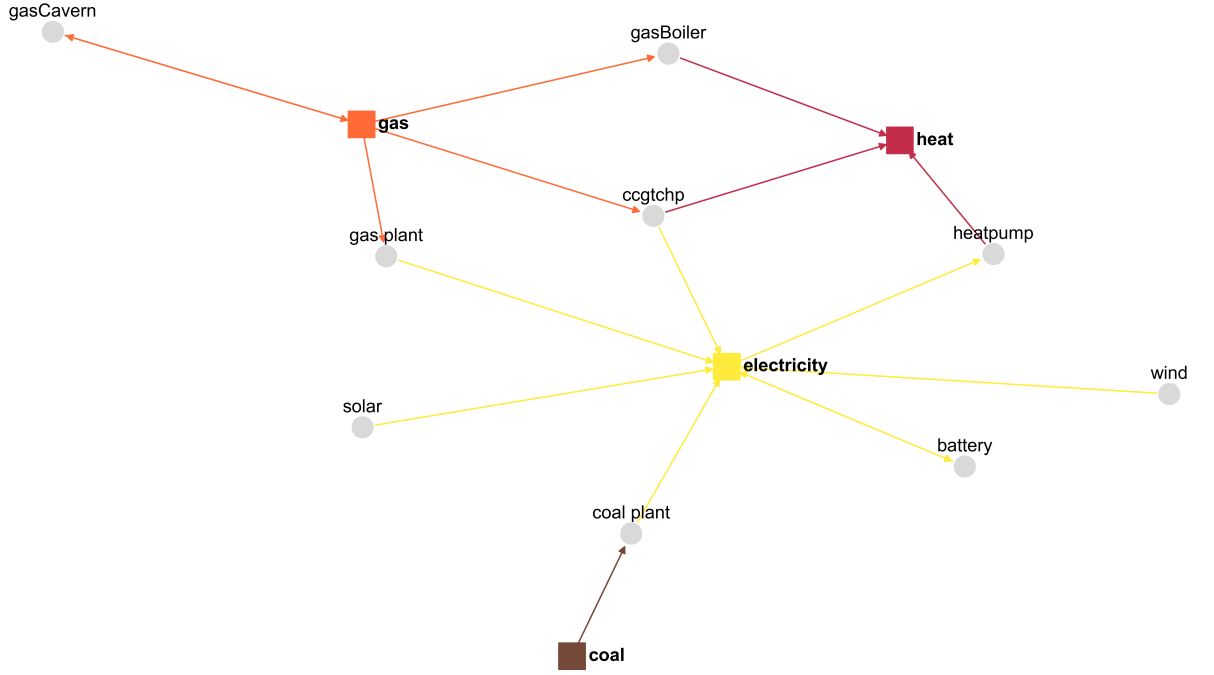


Figure 4.1: Interactions of carriers and technologies.

therefore couple the corresponding carrier balances. Battery storage provides short-term electricity storage, while gas caverns allow natural gas to be stored over longer periods. Electricity and natural gas can additionally be exchanged between regions using HVDC connections and methane pipelines along the links of the synthetic network.

The objective function minimizes total system cost, including the relevant investment and operational cost components. A system-wide emissions constraint limits aggregate emissions across all regions and technologies. The constraint introduces an additional relationship between otherwise spatially distributed investment and dispatch decisions, as emissions generated in one region affect the remaining emissions budget available to the entire system.

The benchmark model is designed to include system characteristics that are particularly relevant to multi-energy systems with a high share of renewable energy sources. The represented carriers exhibit different characteristic time scales, providing a suitable basis for evaluating carrier-specific temporal resolution. Sector-coupling technologies such as heat pumps and CHP units link the individual carrier balances, while battery storage and gas caverns represent short- and long-term storage options. Interregional electricity and natural-gas networks additionally introduce spatial coupling between regions. Together, these components reflect technologies and interactions commonly considered in analyses of future energy systems with increasing reliance on variable renewable generation, while also introducing the technological, temporal, and spatial complexity required to assess the computational effects of variable temporal resolution.

4.2.3 Input Data

The input data for both AnyMOD and REMix are provided through sets of CSV files that define the structure and parameters of the respective model instances. Owing to differences in the data structures of the two frameworks, the organization of these files and the names of individual parameters are not identical. A complete REMix instance of the given model comprises 37 CSV files, whereas the corresponding AnyMOD instance is defined through 12 CSV files. Despite these structural differences, both implementations use equivalent input data for all technological parameters and time-dependent profiles to ensure comparability between the frameworks.

The parameter data include technology-specific investment and operational costs, conversion efficiencies, capacity limits, technical lifetimes, interest rates, and transmission-related parameters. These data are based on the AnyMOD demo model [16]. Availability profiles for wind and photovoltaic generation, as well as electricity and heat demand profiles, are represented by normalized hourly time series and are sourced from Wetzel et al. [34]. The normalized demand profiles are subsequently scaled according to the annual demand assigned to each region. In addition, the coefficient of performance of heat pumps is represented by an hourly time series. Unlike the availability and demand profiles, this series contains directly applicable coefficient values and is therefore not normalized.

4.3 Model Scaling and Instance Generation

4.3.1 Model-Size Variation

The benchmark comprises 20 model instances containing between 2 and 40 regions in increments of two. The technological structure remains unchanged across all model sizes. Each region contains the same set of available technologies and is assigned the same technology-specific expansion costs, operational costs, interest rates, lifetimes, conversion efficiencies, and upper and lower capacity bounds.

The spatial dimension was selected for model-size variation because adding regions provides a comparatively direct way of increasing the number of variables and constraints while leaving the technological and temporal structure unchanged. It is also straightforward to implement systematically across a large number of benchmark instances.

Regional heterogeneity is introduced through the time-dependent input data. Each region is assigned individual hourly profiles for wind availability, photovoltaic availability, electricity demand, and heat demand. The annual electricity and heat demands used to scale the normalized demand profiles also differ between regions. Consequently, adding regions increases not only the spatial extent of the model but also the diversity of the represented renewable-resource and demand conditions.

4.3.2 Model-Expansion Procedure

The model instances are generated automatically by manipulating the CSV files that define the AnyMOD and REMix models. For each model size, a new and complete set of input files is created. This procedure enables the same expansion logic to be applied consistently across both frameworks while accounting for differences in their respective input-data structures.

Nodes and Links

The spatial topology is generated using Delaunay triangulation, following an approach that has also been applied in the construction of synthetic power grids [6]. The triangulation is implemented using the `scipy.spatial.Delaunay` module [31]. Regional coordinates are sampled from a uniform distribution within the unit square. A fixed random seed is used so that the generated coordinates, and therefore the resulting model instances, are reproducible.

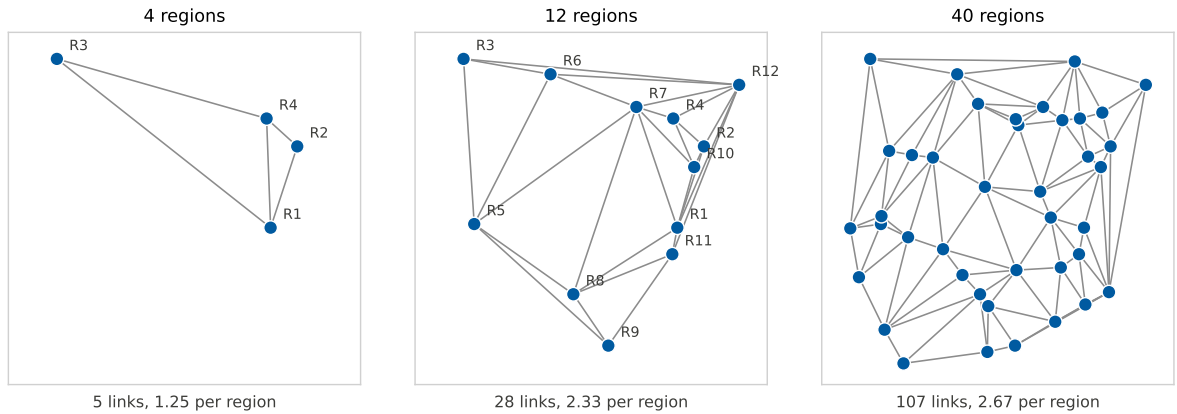


Figure 4.2: Delaunay graphs for 4, 12, and 40 regions.

The Delaunay algorithm partitions the set of regional coordinates into triangular simplices, from which the edges of the inter-regional network are derived. The resulting graph is planar and connects primarily neighboring regions [6], as shown in Figure 4.2. The number of inter-regional links increases with model size, and the average number of links per region rises accordingly. The four-region instance contains five links, corresponding to an average of 1.25 links per region, while the 12-region instance contains 28 links, or 2.33 links per region. For the largest 40-region instance, the network comprises 107 links, corresponding to an average of 2.675 links per region. The two-region instance represents a special case, as Delaunay triangulation cannot be applied to only two points. Its two regions are therefore connected directly by a single link.

The triangulation is recalculated separately for each model size. As a result, a smaller model instance is not necessarily an exact subnetwork of a larger instance. The relative arrangement of regions and the links between them may change when additional

regions are included. The benchmark therefore represents a sequence of independently generated, structurally comparable networks rather than a strictly nested expansion of a single network.

Global Parameters

Parameters that apply identically to all regions are defined globally where supported by the respective framework. AnyMOD permits a larger share of these parameters to be specified globally. In REMix, some region-dependent entries, including capacity limits for conversion and storage technologies and the configuration of sources and sinks, must instead be duplicated when a new region is introduced. The expansion procedure accounts for these framework-specific requirements when generating the corresponding CSV files.

Regional Time Series

Regional time series are selected from a predefined collection containing wind, photovoltaic, electricity-demand, and heat-demand profiles for multiple regions. When a model is expanded, the profiles associated with the newly added regions are inserted into the corresponding input files. A two-region model therefore uses the profiles of the first two regions in the collection, while expanding the model to four regions additionally introduces the profiles assigned to regions three and four. Each region consequently retains its own time-dependent input data across all model instances in which it is included.

Import limits and finite upper bounds on technology expansion are scaled according to the total demand of the respective region. This prevents regions with substantially different demand levels from receiving identical absolute limits and ensures that the associated constraints remain proportional to the scale of the regional energy system.

4.4 Temporal-Resolution Configurations

Three temporal-resolution configurations are considered in the benchmark. The uniform configuration is implemented in both AnyMOD and REMix and serves as the common full-resolution reference case. The two variable-resolution configurations are implemented only in AnyMOD and differ in the degree to which temporal detail is reduced for heat, natural gas, and coal. Table 4.2 summarizes the temporal resolution assigned to each carrier.

Table 4.2: Overview of the temporal-resolution configurations

Carrier	Uniform (AU/RU)	Variable (AV)	Variable coarse (AVC)
Electricity	1 h	1 h	1 h
Heat	1 h	4 h	1 year
Natural gas	1 h	24 h	1 year
Coal	1 h	1 year	1 year

4.4.1 Uniform Temporal Resolution

The uniform temporal-resolution configuration is used for both AnyMOD Uniform (AU) and REMix Uniform (RU). The optimization horizon comprises one year represented by 8,760 consecutive hourly time steps. The same hourly resolution is applied to all four energy carriers, and the original chronological sequence of the input time series is retained. No temporal aggregation is applied in this configuration. Since all time steps have a duration of one hour, they are assigned equal temporal weights internally.

The uniform configurations provide two reference points for the analysis. The comparison between AU and RU is used to assess the equivalence of the benchmark implementations across the two modeling frameworks. Within AnyMOD, AU serves as the full-resolution reference against which the effects of the two variable-resolution configurations are evaluated.

4.4.2 Variable Temporal Resolution

Carrier-specific temporal resolutions are applied in the AnyMOD Variable (AV) and AnyMOD Variable Coarse (AVC) configurations. In both cases, electricity retains an hourly resolution because its supply and demand must be balanced over short intervals and because the operation of variable renewable generation and battery storage depends strongly on intra-day variations.

In the AV configuration, heat is represented using four-hour time steps, natural gas using daily time steps, and coal using a single annual time step. These resolutions are selected according to the carrier-specific temporal characteristics discussed in Section 3.1 and reproduce the resolutions used by Göke [15]. The configuration is intended to reduce the number of operational variables and constraints associated with carriers whose relevant dynamics occur over longer time scales, while retaining hourly detail for electricity.

The AVC configuration applies a deliberately coarse temporal representation. Electricity remains hourly, whereas heat, natural gas, and coal are each represented by a single time step covering the full year. This configuration is not intended to represent a recommended application of the variable-resolution approach. Instead, it serves as a stress test for examining the computational benefits and potential distortions associated with

assigning excessively coarse resolutions to carriers whose intra-annual behavior may remain relevant. Comparing AVC with AU and AV therefore provides insight into the trade-off between computational performance and the preservation of model results.

4.5 Cross-Framework Model Equivalence and Validation

The AnyMOD and REMix benchmark models are designed to represent the same underlying energy system. Both implementations use equivalent time series, technology parameters, costs, efficiencies, capacity limits, transmission parameters, and emissions assumptions. They also include the same principal investment and operational decisions and minimize the same categories of system costs. For the cross-framework comparison, both models use the uniform hourly temporal resolution described above.

Nevertheless, complete mathematical identity between the implementations cannot be achieved because AnyMOD and REMix differ in their internal formulations, parameter structures, and input-data requirements. Parameter names and the organization of the input files differ, and certain restrictions cannot be expressed in exactly the same manner in both frameworks. For example, AnyMOD permits the inflow of natural gas to be bounded at the level of individual time steps, whereas REMix also allows an upper limit to be imposed on the annual sum of inflows. Where such differences occur, the parameterization is adapted to reproduce the same underlying modeling assumption as closely as possible.

Despite these implementation-specific differences, the principal technologies, carrier balances, capacity restrictions, conversion relationships, storage representations, and objective-function components are formulated according to the same underlying logic. The models are therefore treated as functionally equivalent rather than mathematically identical.

Before conducting the full benchmark, the two implementations were validated using smaller model instances. Their objective values and principal optimization results, including technology investments and operational quantities, were compared under identical input assumptions and a uniform hourly resolution. The close agreement observed in these test cases was used to establish that the implementations represent the same benchmark system with sufficient consistency for the subsequent cross-framework comparison.

4.6 Computational Environment

All benchmark experiments were conducted on the CARA high-performance computing cluster. The cluster comprises 740 compute nodes, each equipped with two AMD EPYC 9555 processors and 384 GB of main memory. Each processor provides 64 physical cores, resulting in 128 physical cores per node and, with simultaneous multithreading enabled, 256 logical processors. The compute nodes use a 64-bit Linux operating system. For

each benchmark run, one compute node was allocated exclusively, preventing interference from other workloads on the same node. The optimization solver was restricted to 16 threads for all configurations.

The software environments used for the AnyMOD and REMix implementations are summarized in Table 4.3. The principal difference lies in model construction: AnyMOD uses JuMP, whereas REMix relies on GAMS. Both frameworks use the Gurobi Optimizer, although with slightly different versions. Owing to compatibility issues between AnyMOD and more recent Julia releases, the AnyMOD implementation was executed using the comparatively old Julia version 1.7.3 and may therefore not benefit from performance improvements introduced in later versions.

Table 4.3: Software environments used for the benchmark implementations

Software	AnyMOD	REMix
Modeling framework	AnyMOD 0.1.12	REMix 0.14.0
Programming language	Julia 1.7.3	Python 3.13
Algebraic modeling interface	JuMP 1.29.4	GAMS 53.2.0
Gurobi interface	Gurobi_jll 13.0.2	GAMS–Gurobi link
Gurobi Optimizer	13.0.2	13.0.1
Operating system	64-bit Linux	64-bit Linux

The same solver settings were applied to the AnyMOD and REMix configurations to ensure that observed differences in computational performance were not attributable to differing solver parameterizations. Gurobi was restricted to 16 threads, and the barrier algorithm was selected by setting Method to 2. Each benchmark configuration was evaluated both with presolve enabled and with presolve disabled, allowing the effect of presolve on computational performance to be assessed separately. The complete solver configuration is summarized in Table 4.4. All parameters not listed in the table retained their default Gurobi values.

Table 4.4: Non-default Gurobi settings used for benchmark configurations

Parameter	Value
Threads	16
Method	2
Presolve	0/1
Crossover	0
BarConvTol	10^{-5}
FeasibilityTol	10^{-5}
OptimalityTol	10^{-5}
BarOrder	1
PreDual	0

Although the solver settings were identical, the two framework implementations used slightly different Gurobi versions and solver interfaces. The benchmark therefore compares the complete framework-specific model-generation and optimization workflows under otherwise equivalent hardware allocation and solver parameterization.

4.6.1 Gurobi Presolve

Gurobi presolve applies a collection of preprocessing techniques before the main optimization process to simplify and strengthen the mathematical formulation. These techniques can remove redundant constraints, fix or substitute variables, tighten variable bounds, and strengthen coefficients. The resulting presolved model may therefore contain fewer rows, columns, and non-zero coefficients than the original formulation, thereby potentially reducing the computational effort required by the subsequent solution algorithm [1].

Since the objective of this work is to isolate the computational effect of carrier-specific temporal resolution, presolve represents an additional source of model reduction that can influence the observed differences between the temporal-resolution configurations. The benchmark experiments were therefore performed with presolve disabled to assess the effect of variable temporal resolution on the original formulations. However, presolve is commonly used in practical optimization workflows and disabling it may not reflect typical model applications. To ensure that the observed benefits are also representative under realistic solver settings, all experiments were additionally performed with presolve enabled. The comparison of both settings allows the effect of carrier-specific temporal resolution to be assessed both independently of solver preprocessing and in combination with it.

4.7 Benchmark Execution Procedure

The benchmark runs were submitted to the CARA cluster using the Slurm workload manager. Separate submission scripts were used for the REMix and AnyMOD implementations. In both cases, the submission script defined the model datasets to be evaluated and submitted the corresponding jobs as a Slurm job array using `sbatch`. Each array entry represented one model instance and benchmark configuration. Runs with presolve enabled and disabled were submitted separately so that the effect of presolve could be evaluated under otherwise identical conditions.

For REMix, each job was executed through the framework’s command-line interface and launched on the allocated compute node using `srun`. Wall-clock time was measured externally using timestamps recorded immediately before and after the model execution, in addition to the timing information reported internally by the solver. Peak memory consumption was given by the process’s maximum resident set size reached during

execution, as tracked by the Linux kernel and returned via `wait4()` upon process termination. After completion, the job script extracted the relevant performance indicators, problem statistics, and solution information from the Gurobi log files. These values were combined with the externally measured timing information and written to a CSV file containing one observation for each benchmark run.

Table 4.5: Snippet of a CSV containing measurements from a REMix run

Dataset	Rows	Columns	NNZ	Elapsed (s)	Solve (s)	Objective	Max. RSS (kB)
2_Regions_TREM	508120	525645	1531446	20.56	10.43	1171.279	1114944

The AnyMOD workflow followed the same general submission structure. A submission script defined the datasets and created a Slurm job array, while the corresponding job script specified the working directories and execution parameters, like the number of measured runs, and the number of allocated threads. The job script then launched the Julia benchmark program using the project environment associated with the AnyMOD installation.

Within the Julia program, the dataset and run parameters supplied by the job script were parsed before initializing the model and solver. The required Gurobi settings were then applied, and timing measures and memory usage were recorded using Julia’s `@timed` macro and `Sys.maxrss()`. Solver-specific information, including solution and presolve times, barrier iterations, and matrix statistics, was extracted from the Gurobi output and log files. All measurements were written to a CSV summary file following the same general structure as the REMix benchmark results.

4.7.1 Replication and Result Aggregation

AnyMOD was executed three times independently for each model instance and configuration, as preliminary testing revealed noticeable variation in runtime between repeated executions. The reported AnyMOD performance metrics therefore use the median of the three runs, while the minimum–maximum range is used to indicate run-to-run variation. In contrast, repeated REMix executions showed only minor runtime variation during preliminary testing. REMix was therefore executed once for each model instance and configuration.

To analyze how computational performance develops with increasing model size, selected time-based metrics were additionally evaluated as a function of the number of regions. Where appropriate, the observed relationship was approximated using a power-law function,

$$y = aN^b, \tag{4.2}$$

where N denotes the number of regions, y the respective performance metric, a the fitted prefactor, and b the fitted scaling exponent. The exponent b provides a measure for comparing the rate at which the investigated metric increases with model size across

benchmark configurations, while the coefficient of determination R^2 is used to assess the goodness of fit. The fitted relationships are interpreted as empirical descriptions of the observed scaling behavior over the investigated range of 2–40 regions and not as estimates of asymptotic computational complexity.

The exit code of each job was retained to identify unsuccessful executions and runs terminated because of an error. Measurements from unsuccessful runs were recorded as **NaN** and excluded from the calculation of summary statistics and relative performance measures. Where no successful run was available for a particular model instance and configuration, no corresponding result was included in the analysis.

The generated solver log files and CSV summaries were stored separately for each model instance and execution. This allows all reported measurements to be traced back to the corresponding benchmark instance, run, and solver output.

4.8 Evaluation Metrics

The benchmark evaluation considers computational performance, mathematical problem size, and objective-value consistency. The corresponding metrics were obtained from a combination of external process measurements, framework-specific timing functions, and the Gurobi solver logs.

4.8.1 Computational Performance

Computational performance was primarily evaluated using wall-clock time, solver time, model-build time, and memory consumption. The externally measured wall-clock time, reported as **elapsed_sec**, represents the elapsed time between the recorded start and completion timestamps of the model execution. The solution time reported by Gurobi was stored separately as **solve_sec**, while **presolve_sec** records the time spent in the presolve phase when presolve was enabled.

For both AnyMOD and REMix, model-build time was measured directly within the respective benchmark workflows and represents the time required to construct the mathematical optimization model before it is passed to Gurobi. This allows model-construction performance to be compared directly between the two frameworks, separately from the time required by the optimization solver.

4.8.2 Mathematical Problem Size

The mathematical size and structure of each optimization problem were assessed using both framework-level and solver-level statistics. For AnyMOD, the number of variables and constraints generated by the framework were recorded as **n_variables** and **n_constraints**. A problem fingerprint was additionally recorded for each AnyMOD run

Table 4.6: Computational-performance metrics recorded for each benchmark run

Metric	Description
<code>start_time</code>	Recorded start timestamp
<code>end_time</code>	Recorded completion timestamp
<code>elapsed_sec</code>	Externally measured wall-clock time
<code>solve_sec</code>	Solution time reported by Gurobi
<code>presolve_sec</code>	Time spent in Gurobi presolve
<code>build_sec</code>	Model-construction time
<code>max_rss_kb</code>	Maximum resident set size in kilobytes

to detect differences in the generated solver model. Subsequent inspection showed that differing fingerprints between repeated runs resulted from variations in model-element ordering rather than changes in the underlying mathematical problem.

Gurobi reports the dimensions of the optimization matrix through the number of rows, columns, and non-zero coefficients. These values were recorded before presolve as `rows`, `cols`, and `nnz`. The corresponding dimensions after presolve were stored as `presolved_rows`, `presolved_cols`, and `presolved_nnz`.

Table 4.7: Problem-size and matrix-structure metrics recorded for the benchmark models

Metric	Description
<code>problem_hash</code>	Identifier of the generated AnyMOD problem
<code>rows</code>	Number of rows before presolve
<code>cols</code>	Number of columns before presolve
<code>nnz</code>	Number of non-zero coefficients before presolve
<code>presolved_rows</code>	Number of rows after presolve
<code>presolved_cols</code>	Number of columns after presolve
<code>presolved_nnz</code>	Number of non-zero coefficients after presolve

4.8.3 Objective-Value Consistency

Solution consistency was primarily evaluated using the objective value reported for each completed optimization run. The recorded value, `obj_value`, was compared across the equivalent AnyMOD and REMix implementations and across the different temporal-resolution configurations. For the cross-framework comparison, similar objective values indicate that the two implementations represent the underlying benchmark system consistently. For the variable-resolution configurations, deviations from the uniform AnyMOD reference indicate the extent to which temporal aggregation affects the optimized system result.

The recorded `exit_code` was used to distinguish successfully completed runs from executions that failed or terminated abnormally. Only runs with a successful exit code and a valid objective value were included in comparisons of consistency and computational performance.

5 Results

This chapter presents the results of the benchmark experiments. The presentation focuses primarily on describing the observed results, with only limited interpretation where necessary for clarity. A more detailed interpretation of the findings and their implications is provided in Chapter 6. Unless stated otherwise, AnyMOD results are reported as the median of the three measured runs, with the minimum–maximum range indicating run-to-run variation. REMix was executed once for each benchmark configuration and model size and therefore does not have a corresponding range.

5.1 Objective-Value Consistency

5.1.1 Cross-Framework Equivalence

Figure 5.1 compares the objective values obtained with the uniform-resolution configurations of AnyMOD (AU) and REMix (RU) across the investigated model sizes.

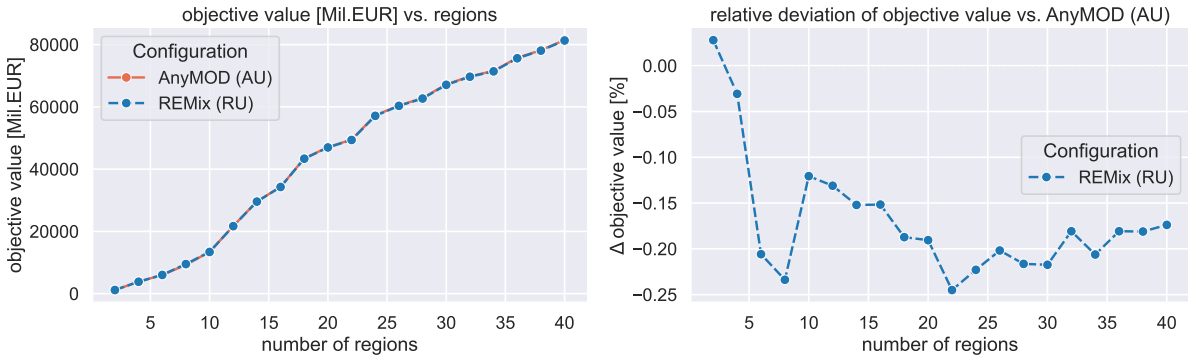


Figure 5.1: Cross-framework comparison of objective values.

The left panel shows the absolute objective value as a function of the number of regions. The objective increases monotonically from approximately 1,170 million EUR for the smallest model to approximately 81,500 million EUR for the largest model, although the increase is not perfectly linear. This reflects the model-expansion procedure, in which newly added regions are heterogeneous with respect to demand and time-dependent input data. Consequently, the additional system cost associated with each newly introduced region varies rather than increasing by a constant amount. The

objective values obtained with AU and RU are visually almost indistinguishable across the complete range of model sizes. The repeated AU runs resulted in identical objective values, such that no variation between runs is visible. This is expected, as repeated executions solve the same optimization problem with the same objective function and input data. Deviations in the resulting objective value would therefore indicate either differences in the generated problem or that the same optimum was not consistently reached.

The right panel shows the relative deviation of the RU objective value from the corresponding AU value. The deviation varies between model sizes but remains below 0.25% in magnitude for all investigated instances. No systematic increase or decrease in the relative deviation is observed with increasing model size. Activating presolve did not affect the obtained objective values for any of the benchmark configurations. As described in Section 4.6.1, presolve is intended to simplify the mathematical formulation without changing its optimal solution, so the absence of objective-value changes between the two presolve settings is consistent with the expected behavior.

5.1.2 Effect of Temporal Resolution

Figure 5.2 compares the objective values obtained with the three AnyMOD temporal-resolution configurations.

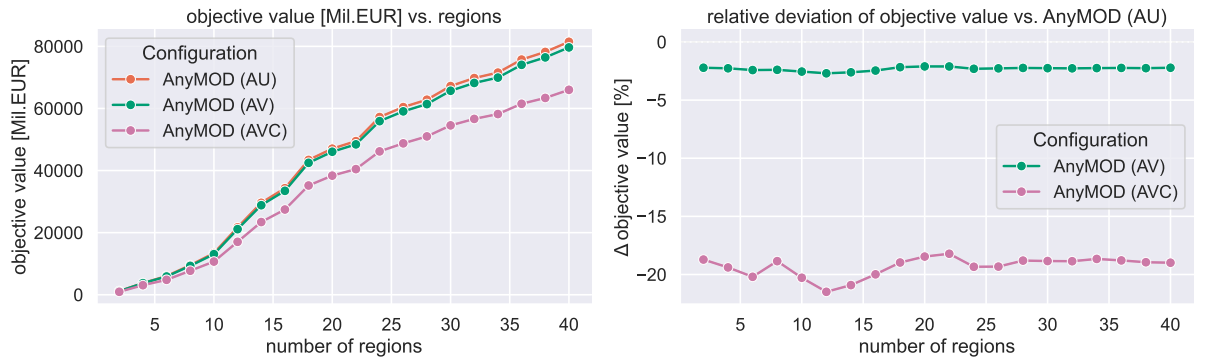


Figure 5.2: Effect of the temporal-resolution configuration on the objective value.

The absolute objective values shown in the left panel increase with the number of regions for all three configurations. The three configurations exhibit a similar overall progression with increasing model size, while systematic differences in the resulting system cost are visible. AV consistently produces slightly lower objective values than AU. The difference is substantially larger for AVC, with the objective value of the largest model being approximately 19% lower than that of AU. As for the uniform configuration, the repeated AnyMOD runs resulted in identical objective values, and no run-to-run variation is therefore visible.

The right panel shows the relative deviation of AV and AVC from the corresponding AU objective value. The relative deviations remain comparatively stable across the

investigated model sizes. AV produces objective values approximately 3% below those of AU, whereas AVC exhibits deviations of approximately 18–22%. No pronounced dependency of the relative deviation on the number of regions is apparent for either configuration.

5.2 Computational Performance

5.2.1 Execution and Solution Time

Total Runtime

Figure 5.3 shows the total runtime of all benchmark configurations as a function of the number of regions, separately for runs with presolve disabled and enabled.

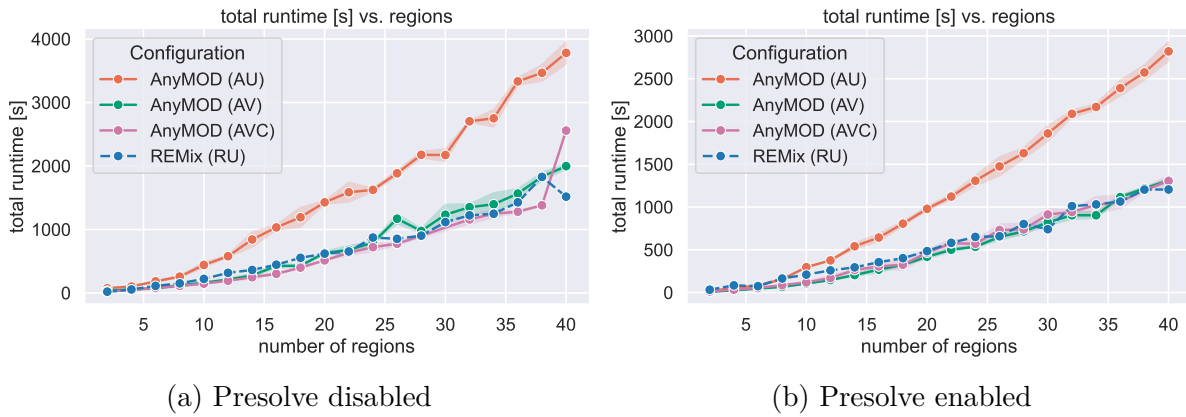


Figure 5.3: Total runtime across configurations and model sizes.

Without presolve, AU exhibits the highest total runtime, reaching approximately 3,900 s for the largest model instances. In comparison, the runtime of AV reaches approximately 2,000 s. RU and AV exhibit similar total runtimes over much of the investigated range, with RU generally being slightly faster, while AVC has the lowest runtime for most model sizes. The measured AnyMOD runtimes show some variation between repeated runs.

For AVC, no successful result was obtained for the 30-region instance without presolve because the solver encountered numerical difficulties. For the 40-region instance, only one of the three runs completed successfully, and this successful run exhibited a pronounced increase in runtime.

With presolve enabled, the overall pattern changes only partly. AU continues to exhibit the highest total runtime, while RU, AV, and AVC remain substantially closer to one another. However, AVC is no longer consistently faster than RU and AV. In addition, all AVC model instances are solved successfully, and the pronounced runtime increase observed for the largest instances without presolve is no longer present.

Solver Time

Figure 5.4 shows the optimization time reported by the solver for all configurations. In contrast to total runtime, this measure excludes framework-level model construction and other execution overhead outside the optimization process.

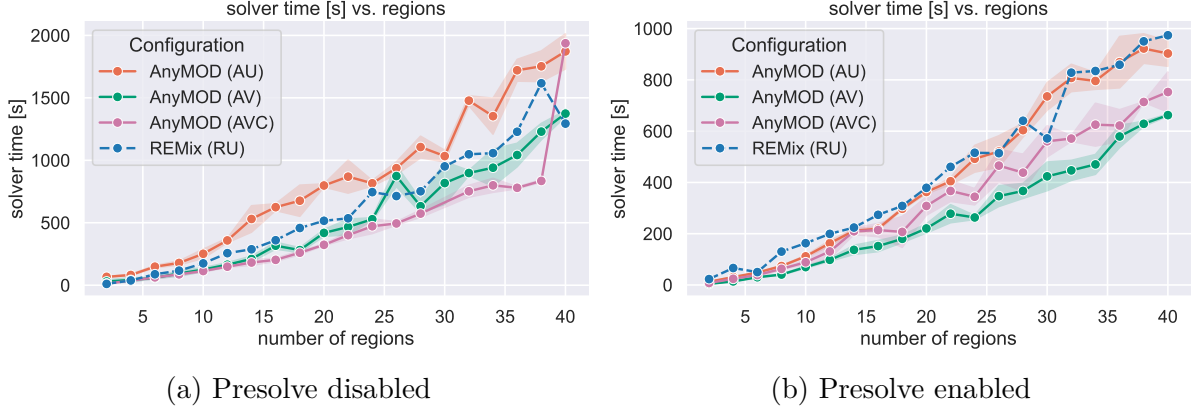


Figure 5.4: Solver time across configurations and model sizes.

Without presolve, solver times reach approximately 2,000 s for the largest model instances. AU remains the slowest configuration, although the differences between AU and the other configurations are smaller than for total runtime. AV exhibits lower solver times than RU for almost all model sizes, despite their comparatively similar total runtimes. AVC is solved faster than AV for most available model sizes. As observed for total runtime, however, no successful result is available for the 30-region AVC instance, while the 40-region model exhibits a pronounced increase in solver time.

Enabling presolve substantially reduces the solver time of AU and AV. For many model instances, the solver times are approximately halved relative to the corresponding runs without presolve. The reduction is less pronounced for AVC and RU. With presolve enabled, AV consistently exhibits lower solver times than AVC. RU exhibits higher solver times than both AV and AU over much of the investigated range.

Relative Solver-Time Reduction

Figure 5.5 shows the relative deviation in solver time of AV and AVC from the corresponding AU configuration. Negative values indicate a reduction in solver time relative to AU.

Without presolve, the relative solver-time reduction achieved by AV varies considerably between model sizes. For some instances, the reduction relative to AU is only approximately 10%, whereas for others it exceeds 60%. No systematic relationship between the magnitude of the reduction and the number of regions is apparent. AVC also exhibits lower solver times than AU for almost all successfully solved model instances, but similarly shows no clear trend with increasing model size.

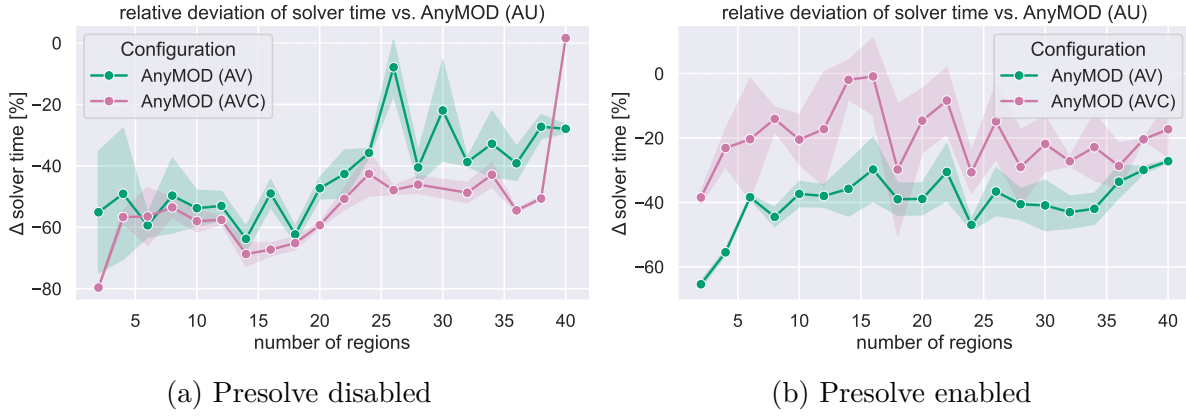


Figure 5.5: Relative solver-time deviation of AV and AVC compared with AU across model sizes.

With presolve enabled, the variation in the relative solver-time reduction of AV decreases substantially. AV reduces solver time by approximately 40% relative to AU for most model instances. No clear overall dependency on model size is observed, although the reduction becomes smaller for the largest instances containing 36–40 regions. AVC exhibits greater variation and consistently requires more solver time than AV under this setting.

Model-Build Time

Figure 5.6a shows the model-build time of all benchmark configurations as a function of the number of regions, while Figure 5.6b presents the corresponding relative deviation from AU. Presolve does not affect model-build time because it is applied only after the mathematical model has been constructed and passed to the solver.

Figure 5.6a shows clear differences in both the absolute model-build times and their development with increasing model size. RU exhibits the lowest model-build times across the entire range and the weakest increase with the number of regions, reaching approximately 80 s for the largest instance. Within AnyMOD, AU shows the strongest growth, increasing to approximately 1,380 s at 40 regions, while AV and AVC reach approximately 445 s and 390 s, respectively. The difference between the AnyMOD configurations becomes increasingly pronounced as model size grows. A comparatively abrupt increase in the model-build time of AU is visible between the 8- and 10-region instances, which is not accompanied by a corresponding increase for the remaining configurations.

The relative deviations shown in Figure 5.6b further illustrate these differences. RU requires less model-build time than AU across all investigated model sizes, with a reduction of approximately 54% for the smallest instance. Its relative advantage increases steadily with model size and reaches approximately 93% for the largest instance, the largest reduction among the investigated configurations. The reductions achieved by

5 Results

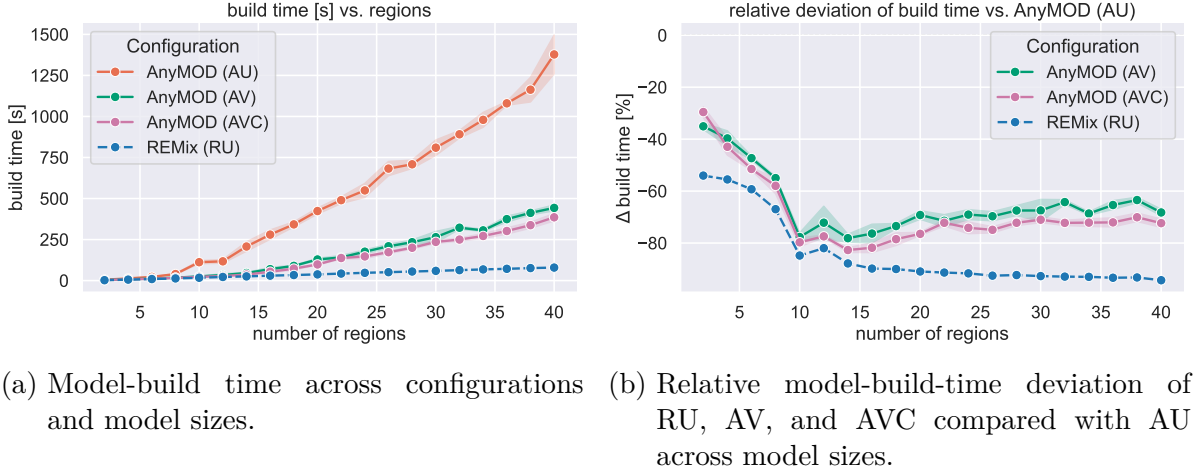


Figure 5.6: Model-build time across configurations and model sizes, in absolute terms and relative to AU.

AV and AVC follow a different pattern. They deepen sharply up to the 10-region instance, reflecting the increase in AU build time described above, and are largest for instances containing approximately 10 to 16 regions, where both configurations reduce model-build time by roughly 78–82% relative to AU. For larger instances, the relative reductions become gradually smaller, amounting to approximately 65–68% for AV and approximately 71–75% for AVC. AVC achieves a slightly larger reduction than AV over most of the range, whereas the two configurations are comparable for the smallest instances. Averaged across all model sizes, the reductions amount to 65.0% for AV and 69.8% for AVC, as reported in Table 5.1.

Average Deviations

Table 5.1 summarizes the average relative deviations in model-build time, solver time, and total runtime from AU. For each model size, the median value of the respective AnyMOD configuration is compared with the median value of AU. The resulting relative deviations are subsequently averaged across the available model instances. Model sizes for which no successful result was obtained are excluded from the corresponding average.

The largest average reductions are observed for model-build time. Relative to AU, AV reduces model-build time by 65.0% on average, while AVC achieves a slightly larger reduction of 69.8%. These reductions are independent of the presolve setting because model construction is completed before Gurobi presolve is applied.

For solver time, the relative performance depends more strongly on the presolve setting. Without presolve, AV reduces solver time by 42.8% on average, while AVC achieves a larger reduction of 54.6%. With presolve enabled, the average solver-time reduction of AV remains comparatively similar at 39.7%, whereas the reduction achieved by AVC decreases substantially to 19.8%.

A similar change is visible for total runtime. Without presolve, AV reduces total runtime by 54.3% relative to AU, while AVC achieves the largest reduction at 61.3%. With presolve enabled, AV provides the larger improvement, reducing total runtime by 56.6%, compared with a reduction of 51.0% for AVC.

Table 5.1: Average relative deviation in model-build time, solver time, and total runtime from AU.

Configuration	Build time	Solver time		Total runtime	
	–	No presolve	Presolve	No presolve	Presolve
AV	-65.0%	-42.8%	-39.7%	-54.3%	-56.6%
AVC	-69.8%	-54.6%	-19.8%	-61.3%	-51.0%

5.2.2 Memory Requirements

Figure 5.7 shows the peak memory requirement of the four benchmark configurations as a function of model size, separately for runs with presolve disabled and enabled.

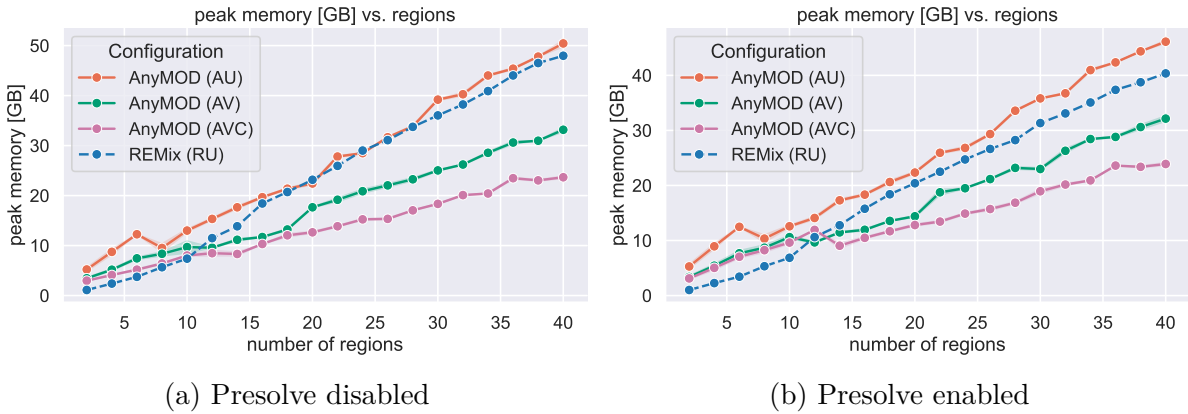


Figure 5.7: Peak memory requirement across configurations and model sizes.

The peak memory requirement increases approximately linearly with the number of regions for all configurations. AU and RU exhibit similar memory requirements over the investigated model sizes. Compared with AU, AV reduces peak memory consumption by approximately 33% on average, while AVC achieves the largest reduction at approximately 48%.

Enabling presolve reduces the peak memory requirement of RU, whereas only a small change is observed for AU. The memory requirements of AV and AVC remain largely unchanged between the two presolve settings.

5.3 Mathematical Problem Size

5.3.1 Rows, Columns, and Non-Zero Coefficients

Figure 5.8 shows the number of non-zero coefficients in the constraint matrix as a function of the number of regions. Without presolve, AU and RU exhibit very similar numbers of non-zero coefficients, with an average deviation of approximately 1%. In comparison with AU, the number of non-zero coefficients is reduced by approximately 50% for AV and 55% for AVC. For all configurations, the number of non-zero coefficients increases approximately linearly with the number of regions.

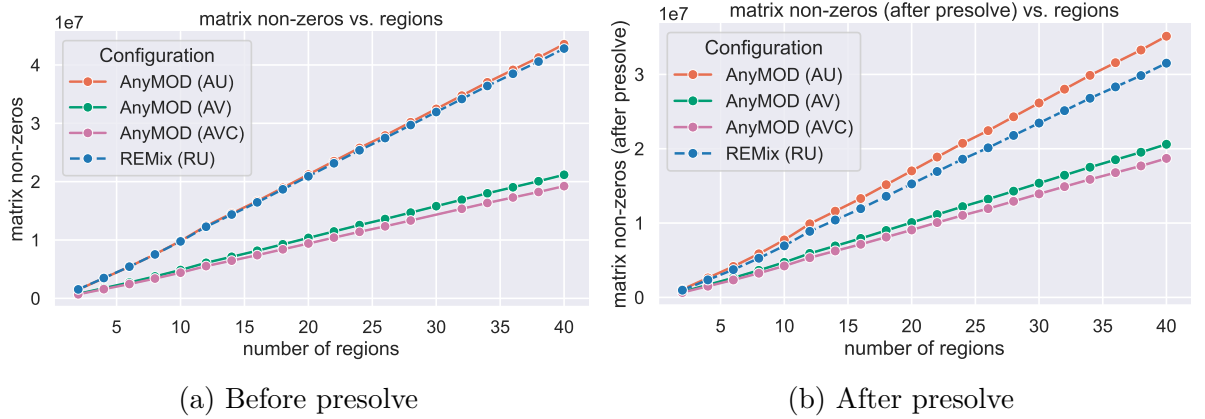


Figure 5.8: Number of non-zero matrix coefficients across configurations and model sizes.

The effect of presolve on the number of non-zero coefficients differs between configurations. A noticeable reduction is observed for RU, whereas the reduction for AU is comparatively small. For AV and AVC, the number of non-zero coefficients remains nearly unchanged. Consequently, the relative reduction in mathematical problem size achieved by AV and AVC compared with AU becomes smaller when the presolved matrices are considered.

Figure 5.9 summarizes the average relative reductions in rows, columns, and non-zero coefficients of AV and AVC compared with AU. Both variable-resolution configurations reduce all three measures of mathematical problem size. The reductions are consistently larger for AVC than for AV, while the relative advantage over AU is smaller after presolve.

5.3.2 Relationship Between Problem Size and Computational Performance

To further examine the relationship between mathematical problem size and computational performance, Figure 5.10 relates the measured performance metrics to the number of non-zero coefficients in the respective matrices. While AV and AVC generally require

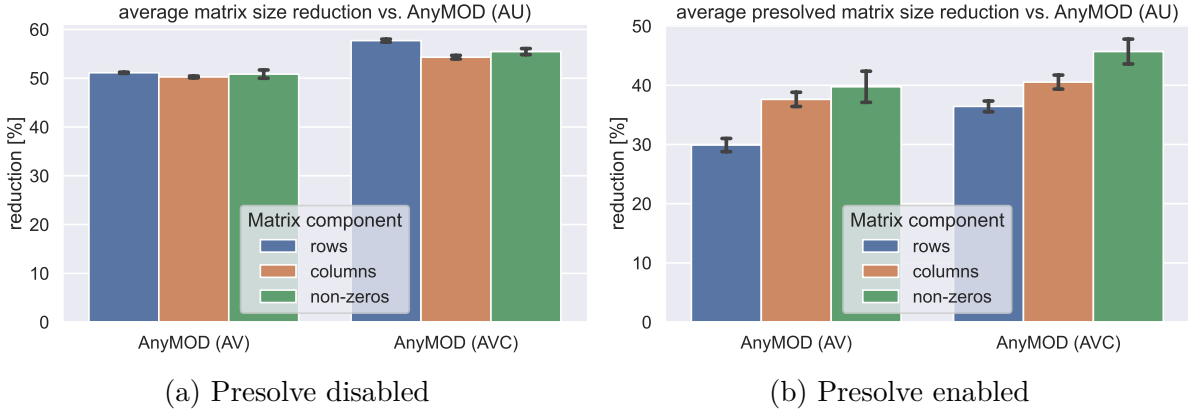


Figure 5.9: Average relative reduction in rows, columns, and non-zero coefficients of AV and AVC compared with AU.

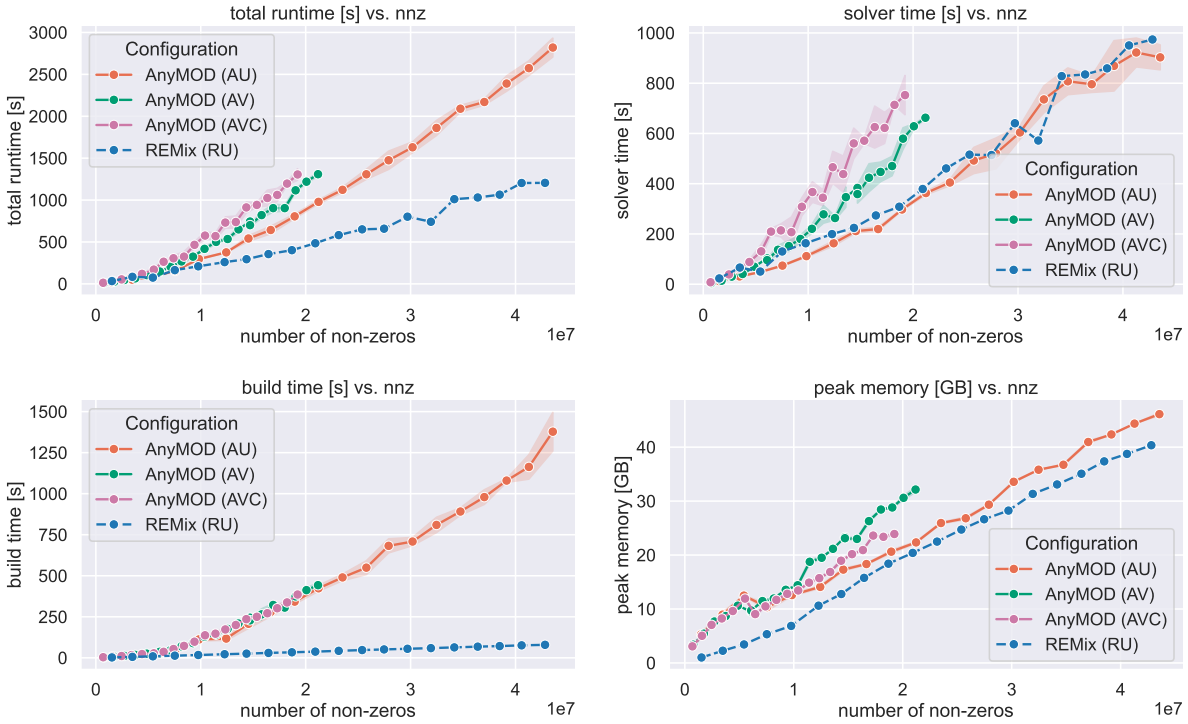


Figure 5.10: Computational performance metrics across configurations as a function of the number of non-zero matrix coefficients with presolve activated.

less computational effort than AU for a given number of regions, a different pattern emerges when configurations with comparable numbers of non-zero coefficients are considered. For comparable numbers of non-zero matrix entries, AV and particularly AVC generally exhibit higher solver and total runtimes than AU. Similar tendencies are visible for model-build time and peak memory requirements.

The reduction in computational requirements observed for AV and AVC with increasing model size therefore does not correspond directly to the reduction in the number of non-zero coefficients. Instead, the lower absolute runtimes of the variable-resolution configurations coincide with their substantially smaller matrices, while differences in computational effort remain between formulations of comparable nominal size.

5.4 Scaling Behavior

To characterize how computational performance develops with increasing spatial model size, power-law relationships of the form $y = aN^b$ were fitted to model-build time, solver time, and total runtime as described in Section 4.7.1, where N denotes the number of regions, b the fitted scaling exponent, and a the fitted prefactor. The fits are based on the runs with presolve enabled and use the median of the repeated executions for each model instance. The resulting parameters are summarized in Table 5.2. The coefficients of determination range from $R^2 = 0.969$ to $R^2 = 0.999$ across all configurations and metrics.

Table 5.2: Power-law fit parameters for model-build time, solver time, and total runtime as a function of the number of regions, for runs with presolve enabled.

Configuration	Model-build time			Solver time			Total runtime		
	b	a	R^2	b	a	R^2	b	a	R^2
REMIX (RU)	1.154	1.184	0.999	1.283	8.370	0.979	1.240	12.138	0.985
AnyMOD (AU)	1.979	0.985	0.989	1.522	3.668	0.995	1.731	5.079	0.994
AnyMOD (AV)	1.748	0.622	0.974	1.665	1.454	0.998	1.716	2.287	0.996
AnyMOD (AVC)	1.666	0.674	0.969	1.542	2.708	0.993	1.596	3.578	0.993

For model-build time, the fitted exponent decreases with the degree of temporal aggregation within AnyMOD, from $b = 1.979$ for AU to $b = 1.748$ for AV and $b = 1.666$ for AVC. The corresponding prefactors are $a = 0.985$ for AU, $a = 0.622$ for AV, and $a = 0.674$ for AVC. RU exhibits the lowest model-build-time exponent at $b = 1.154$, with a prefactor of $a = 1.184$.

For solver time, the fitted exponents do not follow the degree of temporal aggregation. AU has an exponent of $b = 1.522$, compared with $b = 1.665$ for AV and $b = 1.542$ for AVC. The prefactors differ more markedly, with $a = 3.668$ for AU, $a = 1.454$ for AV, and

$a = 2.708$ for AVC. RU exhibits the lowest solver-time exponent at $b = 1.283$, combined with the highest prefactor at $a = 8.370$.

For total runtime, AU and AV exhibit similar exponents of $b = 1.731$ and $b = 1.716$, while their prefactors differ by more than a factor of two, at $a = 5.079$ and $a = 2.287$, respectively. AVC has the lowest total-runtime exponent among the AnyMOD configurations at $b = 1.596$, with a prefactor of $a = 3.578$. RU exhibits the lowest total-runtime exponent overall, with $b = 1.240$ and $a = 12.138$.

Two restrictions apply to the reported fits. They are based on median values without accompanying confidence intervals, so small differences between exponents cannot be resolved. Furthermore, all fits cover model instances containing between 2 and 40 regions and are interpreted as empirical descriptions of the observed behavior over this range rather than as estimates of asymptotic computational complexity.

6 Discussion

6.1 Validity of the Benchmark Comparison

The comparison between the uniform-resolution implementations provides a basis for interpreting the subsequent benchmark results. Across all investigated model sizes, the objective values obtained with AnyMOD (AU) and REMix (RU) differ by less than 0.25%, without a systematic increase in the deviation as the number of regions grows. The mathematical problem sizes of the two uniform configurations are also similar before presolve, with an average deviation of approximately 1% in the number of non-zero coefficients. These results indicate that the benchmark system is represented sufficiently consistently in both frameworks for the purpose of the computational comparison.

Nevertheless, the close agreement between AU and RU should not be interpreted as evidence that the underlying mathematical programs are identical. Differences in framework structure and available modeling constructs required individual aspects of the benchmark model to be represented differently. The comparison therefore concerns functionally equivalent implementations of the same energy-system model rather than mathematically identical formulations. The comparatively small differences between AU and RU provide an indication of the magnitude of these implementation-related effects and are substantially smaller than the differences introduced by changing the temporal resolution.

6.2 Computational Benefits of Variable Temporal Resolution

The benchmark results demonstrate substantial computational benefits from assigning different temporal resolutions to individual energy carriers. Compared with the uniform AnyMOD configuration, AV reduces the number of non-zero matrix coefficients by approximately 50% and peak memory consumption by approximately 33%. These reductions are accompanied by considerable improvements in execution time. Without presolve, AV reduces total runtime by 54.3% and solver time by 42.8% on average. With presolve enabled, the corresponding reductions amount to 56.6% and 39.7%, respectively.

The larger reduction in total runtime compared with solver time shows that the computational benefits are not limited to the optimization process itself. Reducing the

number of temporally indexed model elements also lowers the effort required to construct the mathematical model within AnyMOD. This becomes increasingly relevant for larger instances, for which model construction represents up to 50% of total runtime (AU, 40 regions, presolve on). Evaluating carrier-specific temporal resolution solely on the basis of solver time would therefore underestimate its computational benefit in the investigated implementation.

Variable temporal resolution also substantially reduces peak memory requirements. This represents a relevant benefit independently of execution time, since memory availability can constitute a limiting factor for large energy-system models. A reduction in memory requirements may therefore allow larger or more detailed models to be solved within a given computational environment.

The reduction in mathematical problem size is consistent with the results originally reported by Göke [15]. Using a similar carrier-specific temporal structure, with hourly electricity, four-hourly heat, daily gaseous carriers, and annual coal, Göke likewise obtained a reduction of approximately 50% in mathematical problem size compared with a uniformly hourly formulation. The present benchmark therefore reproduces the principal problem-size benefit of the approach for a different multi-regional model structure. The reduction in solver time is less pronounced in the present experiments, however. Göke reported reductions in barrier solution time of approximately 64–75%, depending on the matrix-ordering method, whereas AV reduces solver time by approximately 40–43% on average in the present benchmark. The qualitative computational benefit of carrier-specific temporal resolution is therefore reproduced, but its magnitude is not directly transferable between model applications. This difference indicates that the solver-time improvement depends not only on the amount of temporal aggregation but also on the structure of the resulting optimization problem and its interaction with the solver.

At the same time, the reductions in mathematical problem size and computational effort are not proportional. This is particularly apparent when the performance metrics are considered as a function of the number of non-zero matrix coefficients rather than the number of regions. For a given spatial model size, AV and AVC generally require less computational effort than AU because their mathematical formulations contain substantially fewer non-zero coefficients. However, when configurations with comparable numbers of non-zero coefficients are considered, AV and particularly AVC generally exhibit higher solver and total runtimes than AU. Similar tendencies are observed for model-build time and peak memory requirements. The results therefore suggest that the substantial reduction in mathematical problem size is an important contributor to the computational advantage of the variable-resolution configurations.

This comparison further demonstrates that the number of non-zero coefficients alone is insufficient to predict computational performance. The same nominal matrix size can be associated with different computational requirements depending on the structure of the underlying formulation. Carrier-specific temporal resolution not only removes variables and constraints but also changes how the remaining model elements are connected, particularly where technologies couple carriers represented at different temporal resolu-

tions. Consequently, a reduction in rows, columns, or non-zero coefficients cannot be expected to translate directly into an equivalent reduction in model-build time, memory requirements, or solver time.

The comparison at similar numbers of non-zero coefficients should nevertheless be interpreted with some caution. Models with comparable matrix sizes across the different configurations do not generally represent the same number of regions and therefore do not constitute otherwise identical optimization problems. The observed relationships consequently do not establish a computational cost per non-zero coefficient or demonstrate that variable temporal resolution is intrinsically more difficult to solve at a fixed problem size. They do, however, provide further evidence that nominal problem size alone cannot explain the observed performance differences and that the structure of the resulting formulation also contributes to computational behavior. This becomes particularly apparent when comparing the AV and AVC configurations, as discussed in the following section.

6.3 Importance of Temporal-Resolution Selection

The comparison between AV and AVC highlights the importance of selecting appropriate carrier-specific temporal resolutions. AV was constructed using resolutions motivated by differences in the temporal characteristics of the represented energy carriers, retaining hourly resolution for electricity while using progressively coarser resolutions for heat, natural gas, and coal. AVC deliberately applies substantially coarser temporal resolutions, with all carriers except electricity represented at annual resolution.

The additional aggregation introduced in AVC does not result in a proportional reduction in mathematical problem size. While AV reduces the number of non-zero coefficients by approximately 50% relative to AU, the corresponding reduction for AVC is only moderately larger at approximately 55%. This limited additional reduction can partly be related to the structure of the model. Electricity remains represented hourly in both AV and AVC and is involved in a large share of generation, storage, transmission, and conversion processes. In addition, investment variables and other model components that are independent of the carrier-specific temporal resolution remain unaffected by further temporal aggregation. Coarsening the remaining carriers therefore cannot eliminate the mathematical structure associated with the hourly representation of electricity.

The comparison of computational performance at comparable matrix sizes further supports this interpretation. Despite its smaller mathematical formulation, AVC generally exhibits higher solver times than AV when configurations with similar numbers of non-zero coefficients are considered. More generally, the variable-resolution formulations tend to require greater computational effort than AU for a comparable nominal matrix size. This is consistent with the observation that solver performance depends on more than the number of non-zero coefficients alone. In the case of AVC, the strongly aggregated coupling between annually represented carriers and hourly electricity variables may con-

tribute to a less favorable mathematical structure, such that the additional reduction in problem size does not necessarily translate into improved solver performance.

The results are consequently consistent with the premise that the value of carrier-specific temporal resolution does not arise from maximizing temporal aggregation. Instead, temporal detail should be retained where it is required to represent the relevant temporal characteristics of a carrier or subsystem, while coarser resolutions may be sufficient where short-term detail is of lesser importance. The comparison between AV and AVC does not by itself prove that the physically motivated resolutions selected for AV are generally preferable, but it demonstrates that substantially increasing the degree of aggregation can provide diminishing computational benefits and simultaneously have a much stronger effect on the resulting solution.

6.3.1 Performance-Solution Trade-Off

The computational benefits of variable temporal resolution must be considered together with the changes introduced in the optimization results. Temporal aggregation does not simply accelerate the solution of an otherwise identical mathematical problem. By changing the temporal representation, it alters the constraints under which investment and operational decisions are optimized and therefore changes the resulting solution.

This trade-off is clearly visible in the comparison between AV and AVC. AV produces objective values approximately 3% below those of AU across the investigated model sizes, whereas the deviation of AVC increases to approximately 18–22%. At the same time, the substantially coarser temporal representation of AVC does not provide a corresponding improvement in computational performance. With presolve enabled, AV reduces total runtime by 56.6% compared with AU, while AVC achieves a slightly smaller reduction of 51.0%. The difference is more pronounced for solver time, for which AV reduces the average runtime by 39.7%, compared with only 19.8% for AVC.

Consequently, increasingly coarse temporal resolution cannot generally be expected to produce an increasingly favorable computational trade-off. In the investigated benchmark, AV combines a substantially smaller deviation from the uniform-resolution solution with better computational performance than AVC when presolve is enabled. The choice of temporal resolution is therefore an integral part of the model-reduction approach rather than a purely technical parameter for minimizing model size.

It is important, however, not to interpret the deviation from AU as a direct measure of model accuracy. AU provides a consistent hourly reference against which the effects of temporal aggregation can be evaluated, but it does not constitute a representation of the true physical energy system. The comparison therefore measures changes relative to the selected full-resolution formulation rather than errors relative to an independently known correct solution.

6.3.2 Interpretation and Acceptability of Deviations

Whether deviations introduced by temporal aggregation are acceptable depends on the purpose and scope of the analysis. The required temporal detail is influenced by the represented technologies and carriers, their characteristic time scales, and the quantities that are relevant to the research question. A deviation that may be acceptable in an exploratory analysis of broad system-development pathways may be unsuitable for a study concerned with storage requirements, peak demand, short-term flexibility, or individual operational decisions.

The approximately 3% deviation observed for AV should therefore not be interpreted as a generally acceptable threshold. It describes the behavior of the investigated benchmark model under one particular set of carrier-specific temporal resolutions. In another model or for another research question, a smaller deviation could already be problematic, while in other applications larger differences may be acceptable if the relevant system characteristics are preserved.

Furthermore, the objective value represents only one characteristic of the optimization result. Similar total system costs do not necessarily imply similar technology portfolios, storage capacities, transmission expansion, or dispatch decisions. Different system configurations can potentially produce similar objective values. Since the primary objective of this work is to assess the computational effects of carrier-specific temporal resolution rather than to conduct a comprehensive analysis of solution accuracy, the results do not establish how closely all individual system decisions are preserved by temporal aggregation.

The interpretation of the results must also reflect the temporal resolution at which they were obtained. A carrier represented at four-hour intervals cannot resolve hourly behavior, while an annually represented carrier contains no information on its intra-annual operation. Results from a temporally aggregated model should therefore only be interpreted at a level of temporal detail supported by the respective representation. In this sense, changing temporal resolution affects not only the numerical values of the results but also the types of conclusions that can meaningfully be derived from them.

6.4 Interaction with Solver

The computational effect of carrier-specific temporal resolution also depends on the interaction between the resulting mathematical formulation and the optimization solver. Presolve substantially reduces solver time for AU and AV, whereas its effect is less pronounced for AVC. Consequently, the relative performance of the temporal-resolution configurations changes when presolve is enabled. Without presolve, AVC achieves the largest average solver-time reduction relative to AU, which is consistent with its smaller mathematical formulation. More notably, however, this ordering reverses once presolve is enabled: AV consistently outperforms AVC despite retaining more temporal detail.

The matrix-size results provide additional context for this behavior. Presolve reduces the mathematical problem size of the uniform configurations more strongly than that of AV and AVC, for which comparatively little additional reduction is observed. This suggests that temporal aggregation and solver presolve provide partly overlapping forms of mathematical problem reduction. Importantly, however, AV retains a substantial computational advantage over AU after presolve, indicating that standard solver pre-processing does not eliminate the benefit of carrier-specific temporal resolution.

The behavior of AVC without presolve further demonstrates that a smaller mathematical problem is not necessarily easier to solve. No successful result was obtained for the 30-region AVC instance, while only one of the three 40-region runs completed successfully and exhibited a pronounced increase in solver time. With presolve enabled, all AVC instances were solved successfully and the runtime increase observed for the largest instances disappeared. This indicates that the extremely coarse temporal representation affects not only the size but also the numerical properties of the optimization problem.

One possible explanation lies in the structure introduced by coupling carriers represented at substantially different temporal resolutions. In AVC, annual carrier quantities can be linked through conversion constraints to large numbers of hourly electricity variables. The resulting constraints therefore aggregate considerably more fine-resolution variables than in AV. At the same time, removing intra-annual restrictions may increase the number of feasible operating patterns that differ only slightly in their objective value. These effects could result in a mathematical structure that is less favorable for the barrier algorithm despite containing fewer variables and constraints.

The available solver diagnostics, however, do not allow the underlying numerical mechanism to be identified conclusively. In particular, the benchmark does not establish whether the observed behavior results from degeneracy, conditioning, the structure of the coupling constraints, or a combination of these factors. The results should therefore primarily be interpreted as evidence that mathematical problem size alone is insufficient to predict solver performance. Extremely coarse temporal aggregation may reduce the nominal size of a model while simultaneously producing less favorable numerical behavior.

6.5 Scaling and Framework-Level Performance

6.5.1 Scaling

The scaling analysis shows that the effect of carrier-specific temporal resolution differs between model construction and optimization. Interpreting the fits in Table 5.2 requires distinguishing the two fitted parameters. Carrier-specific temporal resolution removes an approximately constant share of the mathematical problem across model sizes, as the reductions in rows, columns, and non-zero coefficients reported in Section 5.3.1 are largely independent of the number of regions. A reduction of this kind is expected to

lower the prefactor a while leaving the exponent b unchanged. A change in b would instead indicate that the growth rate of the respective metric is affected, which requires an explanation beyond the proportional reduction in problem size.

Solver time follows the expected pattern. The fitted prefactor decreases from $a = 3.668$ for AU to $a = 1.454$ for AV, a reduction of approximately 60%, whereas the exponents of $b = 1.522$ and $b = 1.665$ remain of comparable magnitude. The solver-time benefit of carrier-specific temporal resolution therefore manifests primarily as a downward shift in level rather than as a change in growth rate, which is precisely what a proportional reduction in problem size would produce. The slightly higher fitted exponent of AV implies that its relative advantage narrows as the model grows, consistent with the observation in Section 5.2.1 that the relative solver-time reduction becomes smaller for the largest instances containing 36–40 regions. The absence of a systematic solver-time scaling improvement should therefore not be read as an unexpected shortcoming of the approach, but as the behavior that a constant-factor reduction in problem size would be expected to produce.

Model-build time, in contrast, does show a systematic reduction in the fitted exponent, from $b = 1.979$ for AU to $b = 1.748$ for AV and $b = 1.666$ for AVC. This cannot be explained by the reduction in problem size alone. A proportional reduction applied to a cost that grows as a power of problem size would lower the prefactor while leaving the exponent unchanged, whereas the relative model-build-time reductions reported in Section 5.2.1 vary considerably across model sizes rather than remaining constant. Model construction in AnyMOD therefore responds disproportionately to the number of temporally indexed model elements, beyond what the accompanying change in overall problem size would imply. The comparison with mathematical problem size makes the underlying behavior more apparent: as reported in Section 5.3.1, the number of non-zero coefficients increases approximately linearly with the number of regions for all configurations, so that the model-build time of AU grows approximately quadratically with the number of regions while the generated optimization problem grows approximately linearly. Model construction in AnyMOD is consequently superlinear in the size of the problem being constructed. This points to the framework-level model-generation procedure rather than to the temporal-resolution concept itself as the source of the observed scaling behavior, an interpretation supported by the comparison with REMix discussed in Section 6.5.2.

The total-runtime fits combine these two patterns. AU and AV exhibit almost identical exponents of $b = 1.731$ and $b = 1.716$, while their prefactors differ by more than a factor of two at $a = 5.079$ and $a = 2.287$. The fitted relationship for AV therefore corresponds to an approximately constant relative reduction across the investigated range, consistent with the average total-runtime reduction reported in Table 5.1. AVC combines the lowest total-runtime exponent among the AnyMOD configurations, $b = 1.596$, with a higher prefactor than AV at $a = 3.578$, so that its fitted advantage over AV increases with model size while remaining smaller in absolute level at the lower end of the range. The weaker growth of model construction is thus only partly transferred to total runtime, because no comparable change in growth rate occurs during optimization and because

model construction accounts for a declining share of total runtime as temporal resolution is reduced.

These findings clarify the distinction between absolute computational savings and scaling behavior. AV substantially reduces model-build time, solver time, and total runtime relative to AU across the investigated model sizes, even though its solver-time exponent is higher and its total-runtime exponent is nearly identical. Carrier-specific temporal resolution can therefore provide considerable computational benefits without changing how computational effort grows with spatial model size, and the two properties should not be conflated when assessing the approach. Two restrictions apply to this interpretation. The fits are based on median values without confidence intervals, so only the larger differences between exponents, such as that between AU and AVC for model-build time, can be regarded as meaningful; differences such as $b = 1.522$ for AU and $b = 1.542$ for AVC cannot be resolved on the available data. In addition, the fits describe the runs with presolve enabled and over the investigated range of 2–40 regions, and the interaction between temporal resolution and solver preprocessing discussed in Section 6.4 means that the fitted relationships should not be assumed to hold under different solver settings.

6.5.2 Framework-Level Performance

The comparison with REMix indicates that the scaling of model construction discussed in Section 6.5.1 is a property of the framework rather than of the modeled problem. RU exhibits a model-build-time exponent of $b = 1.154$, which is close to the approximately linear growth of the mathematical problem size with the number of regions reported in Section 5.3.1. The model-generation effort of REMix therefore grows at approximately the same rate as the problem it generates, whereas the corresponding exponent of $b = 1.979$ for AU indicates that model construction in AnyMOD grows considerably faster than the size of the resulting optimization problem. This is particularly notable because RU and AU have comparable mathematical problem sizes before presolve, with an average deviation of approximately 1% in the number of non-zero coefficients. The difference in model-construction behavior consequently cannot be attributed to differences in the size of the generated problem and instead reflects how the respective frameworks assemble it. In this respect, the near-linear behavior of REMix is what an efficient model generator would be expected to produce, and the superlinear growth observed for AnyMOD constitutes the result requiring explanation.

The low fitted exponents of RU should not, however, be read as indicating uniformly faster execution. RU exhibits the lowest exponent for all three timing measures but also the highest prefactor, with $a = 8.370$ for solver time and $a = 12.138$ for total runtime. For solver time in particular, the fitted relationships correspond to RU requiring more time than AU over much of the investigated range, which is consistent with the measurements reported in Section 5.2.1. The framework-level advantage of REMix in the present benchmark is therefore concentrated in model generation rather than in

optimization, and the differences in fitted exponents describe growth rates rather than absolute performance.

Several factors limit how far these differences can be attributed to the frameworks themselves. AnyMOD and REMix differ in their data-processing and model-generation procedures, programming languages, software architectures, and solver interfaces. Owing to compatibility constraints, the AnyMOD implementation was executed using Julia 1.7.3 and may therefore not benefit from performance improvements introduced in later releases, including improvements relevant to compilation latency. The two implementations additionally used slightly different Gurobi versions and different solver interfaces. The measured differences consequently reflect the complete framework-specific workflows in the software environments available for this benchmark rather than intrinsic properties of the underlying modeling concepts.

The comparison is also restricted in scope. The graph-based structure used by AnyMOD is not limited to temporal resolution but forms part of a more general representation that can support other forms of aggregation and substitution, including varying spatial resolution and carrier substitution. These capabilities are not exercised in the present benchmark. The additional model-construction effort associated with this general structure can therefore not be assessed solely on the basis of the carrier-specific temporal-resolution application considered here, and in applications making use of these capabilities the greater framework-level effort may be accompanied by modeling benefits outside the scope of this work. For the same reason, the comparison between AU, AV, and AVC provides the more direct assessment of the effect of carrier-specific temporal resolution, since these configurations are generated within the same framework, software environment, and computational setup.

6.6 Limitations and Generalizability

Several measures were taken to make the benchmark representative of relevant characteristics of applied multi-energy system optimization models. The model includes multiple interacting energy carriers, a range of conversion, generation, storage, and transmission technologies, and regional heterogeneity in electricity and heat demand and renewable generation profiles. Interregional networks are generated using Delaunay triangulation, while a system-wide CO₂ emission limit introduces a global constraint coupling decisions across regions, carriers, and time steps. These features were intended to avoid evaluating variable temporal resolution on an overly simplified or homogeneous model. Nevertheless, complete generalization from a single benchmark is not possible. Multi-energy systems differ substantially in carrier composition, technological scope, spatial structure, temporal characteristics, and purpose. The results should therefore be interpreted as evidence that the computational benefits observed by Göke [15] can be reproduced for a different model structure, rather than as universal performance estimates for MES optimization models.

A related limitation is that all experiments use one underlying carrier and technology configuration, with only the number of regions varied between 2 and 40. The observed benefits therefore depend on the specific distribution of variables and constraints across carriers, technologies, storage options, and conversion processes. Only two variable-resolution configurations were investigated: AV, with physically motivated carrier-specific resolutions, and AVC, with substantially coarser resolutions. The experiments consequently do not constitute a systematic sensitivity analysis and cannot identify generally optimal temporal resolutions for individual carriers. Instead, they demonstrate the computational and solution effects of the selected configurations within the investigated benchmark.

Temporal aggregation also introduces an implicit representation of flexibility. Increasing the duration of a time step removes constraints on intra-period variations and therefore implicitly assumes that the corresponding carrier or subsystem can accommodate these variations. The amount of flexibility introduced in this way is difficult to quantify or control directly. Where flexibility arises from identifiable physical mechanisms, such as storage, thermal inertia, gas-network linepack, demand response, or conversion-technology operation, explicit representation through model parameters and constraints provides greater transparency. Temporal resolution should therefore be understood as a modeling abstraction rather than as a direct representation or substitute for physical flexibility. Its appropriateness depends on the temporal characteristics relevant to the particular research question.

The assessment of changes in the optimization outcome is another limitation. The present analysis primarily uses the objective value to quantify deviations from the uniform-resolution AU formulation. The objective value does not, however, characterize the complete system solution: installed capacities, technology selection, storage deployment, transmission expansion, and dispatch may differ even when total system costs remain similar. AU should therefore be regarded as a reference formulation rather than ground truth, and the results should be interpreted as an assessment of computational performance and objective-value deviations rather than a comprehensive evaluation of solution accuracy.

The computational results are additionally dependent on the solver and implementation used. All experiments employ the Gurobi barrier method with a common parameterization, and different solvers, algorithms, or settings may respond differently to the structural changes introduced by variable temporal resolution. Furthermore, only the graph-based implementation available in AnyMOD was evaluated. Alternative implementations could represent different temporal resolutions differently and may therefore exhibit different model-construction times, memory requirements, or solver behavior. The measured computational effects consequently reflect both the underlying temporal-resolution approach and its specific implementation.

Repeated AnyMOD executions of nominally identical configurations produced different problem fingerprints and noticeable runtime variation, although the generated models were found to differ only in the ordering of model elements rather than in their math-

emational content. The objective values, model dimensions, coefficients, and underlying problem data remained unchanged. The differing fingerprints therefore do not indicate that different optimization problems were generated between runs. Nevertheless, differences in model ordering may influence the internal matrix representation and subsequent solver behavior, and thus provide a possible source of the observed runtime variation. Reporting the median and minimum–maximum range of three independent executions reduces the influence of individual runtime fluctuations on the presented results. More generally, the absolute runtime and memory results depend on the software versions, solver, solver settings, and computing environment used in the experiments. The fitted scaling relationships should likewise be interpreted only over the investigated range of 2–40 regions and not as estimates of asymptotic computational complexity. Broader generalization would require benchmarks with different carrier and technology compositions, temporal configurations, solver algorithms, and implementations.

7 Conclusion

This work investigated the computational effects of carrier-specific temporal resolution in multi-energy system optimization models. Building on the graph-based formulation introduced by Göke [15], the approach was evaluated using a synthetic multi-regional benchmark model implemented in AnyMOD. Model size was systematically increased from 2 to 40 regions, and two carrier-specific temporal-resolution configurations were compared with a uniform hourly reference. A uniform REMix implementation was additionally used to assess the consistency of the benchmark formulation across modeling frameworks.

7.1 Research Questions and Main Findings

The first research question asked:

How does carrier-specific temporal resolution affect computational performance and mathematical problem size compared with uniform temporal resolution?

The results demonstrate substantial computational benefits from carrier-specific temporal resolution. Compared with the uniform AnyMOD configuration, AV reduces the number of non-zero matrix coefficients by approximately 50% and peak memory consumption by approximately 33%. Model-build time decreases by approximately 65% on average. Depending on the presolve setting, total runtime is reduced by approximately 54–57%, while solver time decreases by approximately 40–43%.

These findings reproduce the principal computational benefit reported by Göke [15] for a different energy-system model. In both studies, carrier-specific temporal resolution reduces mathematical problem size by approximately 50%. The solver-time reduction is less pronounced in the present benchmark, however, with approximately 40–43% compared with the 64–75% reduction in barrier solution time reported by Göke. The present results therefore confirm that substantial problem-size reductions can be achieved beyond the original application while also showing that the resulting solver-time improvement depends on the structure of the generated optimization problem.

The relationship between mathematical problem size and computational performance further supports this distinction. At a given number of regions, AV and AVC generally require less computational effort than AU while also generating substantially smaller matrices. When configurations with comparable numbers of non-zero coefficients are considered, however, the variable-resolution formulations generally require more solver

and total runtime than AU. Reducing mathematical problem size is therefore an important source of the observed computational benefit, but the number of non-zero coefficients alone is insufficient to predict execution time. The structure and coupling of the resulting formulation also influence computational performance.

The second research question asked:

How do the computational benefits of carrier-specific temporal resolution change with increasing model size?

The computational benefits persist as the spatial size of the benchmark increases, but the different components of execution time exhibit distinct scaling behavior. The clearest scaling improvement is observed for model construction. Within AnyMOD, the fitted power-law exponent for model-build time decreases from $b = 1.979$ for AU to $b = 1.748$ for AV and $b = 1.666$ for AVC, indicating weaker growth in model-construction effort as temporal resolution is reduced.

No corresponding systematic improvement is observed for solver time. AV has a fitted solver-time exponent of $b = 1.665$, compared with $b = 1.522$ for AU, and the relative solver-time reduction shows no clear trend with increasing number of regions. Carrier-specific temporal resolution therefore continues to reduce absolute solver time as the model grows, but its relative solver-time advantage does not systematically increase with spatial model size.

The total-runtime scaling reflects the combination of these effects. AU and AV exhibit similar fitted exponents of $b = 1.731$ and $b = 1.716$, respectively, while AVC has a lower exponent of $b = 1.596$. The stronger scaling improvement in model construction is therefore only partly transferred to total runtime because no comparable improvement occurs during optimization. This distinction shows that substantial absolute computational savings can be achieved even when the overall scaling behavior changes only moderately.

The third research question asked:

How does the degree of carrier-specific temporal aggregation affect the trade-off between computational performance and deviations from the uniform-resolution solution?

The comparison between AV and AVC demonstrates that increasingly coarse temporal aggregation does not necessarily produce an increasingly favorable trade-off. AV produces objective values approximately 3% below those of AU, whereas the substantially coarser AVC configuration results in deviations of approximately 18–22%. At the same time, the additional reduction in mathematical problem size is comparatively small: the reduction in non-zero coefficients increases only from approximately 50% for AV to approximately 55% for AVC.

The additional aggregation introduced in AVC also does not translate into a consistent improvement in runtime. Without presolve, AVC achieves larger average reductions in solver time and total runtime than AV. Once presolve is enabled, however, this ordering reverses: AV reduces total runtime by 56.6% and solver time by 39.7% relative to AU,

compared with reductions of 51.0% and 19.8% for AVC. The largest AVC instances additionally exhibit numerical difficulties when presolve is disabled. The comparison therefore shows that further temporal aggregation may reduce mathematical problem size without producing a corresponding improvement in computational performance.

Taken together, AVC provides only a modest additional reduction in mathematical problem size relative to AV, does not provide a consistent runtime advantage, and introduces substantially larger deviations from the uniform-resolution solution. The results therefore emphasize that the purpose of carrier-specific temporal resolution is not to maximize temporal aggregation. Instead, temporal detail should be allocated selectively, retaining fine resolution where it is important for representing the relevant system behavior and applying coarser resolution where additional temporal detail contributes comparatively little to the intended analysis.

7.2 Implications

The results demonstrate the potential of carrier-specific temporal resolution as a model-reduction approach for multi-energy system optimization. In the investigated benchmark, substantial reductions in mathematical problem size, memory requirements, model-build time, solver time, and total runtime are achieved while retaining hourly resolution for electricity and applying coarser resolutions to carriers with different temporal characteristics.

Such reductions can be valuable beyond the direct decrease in computation time. Computational resources saved through temporal model reduction may instead be used to increase other dimensions of model detail, including spatial resolution, technological scope, scenario coverage, or the modeled time horizon. This becomes particularly relevant as additional energy carriers and sector-coupling technologies are integrated into MES, since applying the finest required temporal resolution uniformly across all carriers can introduce substantial computational effort.

At the same time, the results show that the computational benefit of carrier-specific temporal resolution depends on the structure of the underlying model. The share of the formulation that remains at fine temporal resolution, the technologies coupling carriers represented at different resolutions, the representation of storage and transmission, and the resulting matrix structure all influence the realized performance improvement. The comparison between AV and AVC further demonstrates that additional temporal aggregation may reduce mathematical problem size without producing a corresponding improvement in runtime. Carrier-specific temporal resolution should therefore be understood as a model-design choice in which computational savings depend on how temporal detail is distributed across the system.

The present analysis also distinguishes between the general concept of carrier-specific temporal resolution and its implementation in AnyMOD. AnyMOD realizes variable

temporal and spatial resolution through a graph-based representation that provides modeling capabilities extending beyond the temporal-resolution application investigated in this work. At the same time, the benchmark shows that framework-level model construction can represent a substantial share of total execution time and that its scaling differs considerably between AnyMOD and REMix. This indicates that the computational benefit of temporal model reduction is influenced not only by the reduction in mathematical problem size but also by how the modeling framework constructs the resulting optimization problem. Alternative implementations of carrier-specific temporal resolution may therefore exhibit different model-construction and runtime behavior.

7.3 Future Work

A central direction for future work is the systematic selection of carrier-specific temporal resolutions. The present work compares one configuration selected based on carrier-specific temporal characteristics with one deliberately coarse configuration, but does not determine an optimal resolution for any carrier. Future research could investigate methods for identifying the coarsest temporal representation that preserves the model outputs relevant to a particular application within a predefined tolerance.

Such an analysis should also extend the comparison beyond objective values. Installed capacities, technology selection, storage requirements, transmission expansion, renewable curtailment, and dispatch patterns could be compared across temporal resolutions. This would provide a stronger basis for determining whether reductions in temporal detail preserve the conclusions relevant to a particular modeling application.

Further benchmarks could systematically vary carrier and technology composition, the degree of sector coupling, storage characteristics, and spatial network structure. In particular, varying network density independently of the number of regions would help distinguish the effects of increasing spatial size from those of increasing spatial coupling. These experiments could identify which model characteristics most strongly influence the computational benefits of carrier-specific temporal resolution.

The relationship between temporal aggregation and explicitly modeled physical flexibility also warrants further investigation. Coarser temporal resolution removes intra-period restrictions implicitly, whereas mechanisms such as storage, thermal inertia, gas-network linepack, demand response, or operational flexibility can be represented explicitly. Comparing these approaches could help determine when coarse temporal resolution provides an adequate abstraction and when the underlying flexibility should instead be modeled directly.

Finally, the computational analysis could be extended to alternative optimization algorithms and implementations of carrier-specific temporal resolution. The present results show that solver behavior depends on the structure produced by temporal aggregation and on its interaction with presolve. Benchmarking alternative solvers and formulations would therefore help distinguish computational effects arising from the general

concept of carrier-specific temporal resolution from those associated with a particular implementation or solution method.

List of Figures

3.1	Reduced rooted tree of time steps	17
4.1	Interactions of carriers and technologies.	24
4.2	Delaunay graphs for 4, 12, and 40 regions.	26
5.1	Cross-framework comparison of objective values.	37
5.2	Effect of the temporal-resolution configuration on the objective value. . .	38
5.3	Total runtime across configurations and model sizes.	39
5.4	Solver time across configurations and model sizes.	40
5.5	Relative solver-time deviation of AV and AVC compared with AU across model sizes.	41
5.6	Model-build time across configurations and model sizes, in absolute terms and relative to AU.	42
5.7	Peak memory requirement across configurations and model sizes.	43
5.8	Number of non-zero matrix coefficients across configurations and model sizes.	44
5.9	Average relative reduction in rows, columns, and non-zero coefficients of AV and AVC compared with AU.	45
5.10	Computational performance metrics across configurations as a function of the number of non-zero matrix coefficients with presolve activated. . .	45

List of Tables

4.1	Overview of the benchmark configurations	22
4.2	Overview of the temporal-resolution configurations	28
4.3	Software environments used for the benchmark implementations	30
4.4	Non-default Gurobi settings used for benchmark configurations	30
4.5	Snippet of a CSV containing measurements from a REMix run	32
4.6	Computational-performance metrics recorded for each benchmark run . .	34
4.7	Problem-size and matrix-structure metrics recorded for the benchmark models	34
5.1	Average relative deviation in model-build time, solver time, and total runtime from AU.	43
5.2	Power-law fit parameters for model-build time, solver time, and total runtime as a function of the number of regions, for runs with presolve enabled.	46

Bibliography

- [1] T. Achterberg, R. E. Bixby, Z. Gu, E. Rothberg, and D. Weninger. Presolve Reductions in Mixed Integer Programming. *INFORMS Journal on Computing*, 32(2): 473–506, Apr. 2020. ISSN 1091-9856, 1526-5528. doi: 10.1287/ijoc.2018.0857. URL <http://pubsonline.informs.org/doi/10.1287/ijoc.2018.0857>.
- [2] A. A. M. Aljabery, H. Mehrjerdi, S. Mahdavi, and R. Hemmati. Multi carrier energy systems and energy hubs: Comprehensive review, survey and recommendations. *International Journal of Hydrogen Energy*, 46(46):23795–23814, July 2021. ISSN 0360-3199. doi: 10.1016/j.ijhydene.2021.04.178. URL <https://www.sciencedirect.com/science/article/pii/S0360319921016256>.
- [3] R. Amin, D. Mathur, D. Ompong, and K. K. Zander. Integrating Social Aspects into Energy System Modelling Through the Lens of Public Perspectives: A Review. *Energies*, 17(23):5880, Jan. 2024. ISSN 1996-1073. doi: 10.3390/en17235880. URL <https://www.mdpi.com/1996-1073/17/23/5880>.
- [4] O. Arvesen, V. Medbo, S. E. Fleten, A. Tomasgard, and S. Westgaard. Linepack storage valuation under price uncertainty. *Energy*, 52:155–164, Apr. 2013. ISSN 0360-5442. doi: 10.1016/j.energy.2012.12.031. URL <https://www.sciencedirect.com/science/article/pii/S0360544212009589>.
- [5] F. J. Baader, S. Moret, W. Wiesemann, I. Staffell, and A. Bardow. Streamlining Energy Transition Scenarios to Key Policy Decisions, Nov. 2023. URL <http://arxiv.org/abs/2311.06625>. arXiv:2311.06625 [cs.LG].
- [6] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye. Grid Structural Characteristics as Validation Criteria for Synthetic Networks. *IEEE Transactions on Power Systems*, 32(4):3258–3265, July 2017. ISSN 1558-0679. doi: 10.1109/TPWRS.2016.2616385. URL <https://ieeexplore.ieee.org/document/7725528>.
- [7] T. Blanke, K. S. Schmidt, J. Götttsche, B. Döring, J. Frisch, and C. van Treeck. Time series aggregation for energy system design: review and extension of modelling seasonal storages. *Energy Informatics*, 5(1):17, Sept. 2022. ISSN 2520-8942. doi: 10.1186/s42162-022-00208-5. URL <https://doi.org/10.1186/s42162-022-00208-5>.
- [8] K.-K. Cao, K. von Krbek, M. Wetzel, F. Cebulla, and S. Schreck. Classification and Evaluation of Concepts for Improving the Performance of Applied Energy System Optimization Models. *Energies*, 12(24):4656, Jan. 2019. ISSN 1996-1073. doi: 10.3390/en12244656. URL <https://www.mdpi.com/1996-1073/12/24/4656>.

- [9] J. DeCarolís, H. Daly, P. Dodds, I. Keppo, F. Li, W. McDowall, S. Pye, N. Strachan, E. Trutnevyte, W. Usher, M. Winning, S. Yeh, and M. Zeyringer. Formalizing best practice for energy system optimization modelling. *Applied Energy*, 194:184–198, May 2017. ISSN 0306-2619. doi: 10.1016/j.apenergy.2017.03.001. URL <https://www.sciencedirect.com/science/article/pii/S0306261917302192>.
- [10] M. K. Deshmukh and S. S. Deshmukh. Modeling of hybrid renewable energy systems. *Renewable and Sustainable Energy Reviews*, 12(1):235–249, Jan. 2008. ISSN 1364-0321. doi: 10.1016/j.rser.2006.07.011. URL <https://www.sciencedirect.com/science/article/pii/S1364032106001134>.
- [11] M. M. Frysztacki, J. Hörsch, V. Hagenmeyer, and T. Brown. The strong effect of network resolution on electricity system models with high shares of wind and solar. *Applied Energy*, 291:116726, June 2021. ISSN 03062619. doi: 10.1016/j.apenergy.2021.116726. URL <http://arxiv.org/abs/2101.10859>. arXiv:2101.10859 [physics.soc-ph].
- [12] Z. Gao, M. Gazzani, D. A. Tejada-Arango, A. Soares Siqueira, N. Wang, M. Gibescu, and G. Morales-España. Fully flexible temporal resolution for energy system optimization. *Applied Energy*, 396:126267, Oct. 2025. ISSN 03062619. doi: 10.1016/j.apenergy.2025.126267. URL <https://linkinghub.elsevier.com/retrieve/pii/S0306261925009973>.
- [13] M. Geidl. *Integrated Modeling and Optimization of Multi-Carrier Energy Systems*. PhD thesis, ETH Zurich, 2007. URL <http://hdl.handle.net/20.500.11850/123494>. Artwork Size: 143 p. Medium: application/pdf Pages: 143 p.
- [14] L. Göke. AnyMOD.jl: A Julia package for creating energy system models. *SoftwareX*, 16:100871, Dec. 2021. ISSN 23527110. doi: 10.1016/j.softx.2021.100871. URL <http://arxiv.org/abs/2011.00895>. arXiv:2011.00895 [physics].
- [15] L. Göke. A graph-based formulation for modeling macro-energy systems. *Applied Energy*, 301:117377, Nov. 2021. ISSN 03062619. doi: 10.1016/j.apenergy.2021.117377. URL <http://arxiv.org/abs/2004.10184>. arXiv:2004.10184 [physics].
- [16] Göke, Leonard. Anymod github repository. <https://github.com/leonardgoeke/AnyMOD.jl/tree/master/examples/demo>. Accessed: 2026-03-02.
- [17] G. Haydt, V. Leal, A. Pina, and C. A. Silva. The relevance of the energy resource dynamics in the mid/long-term energy planning models. *Renewable Energy*, 36(11):3068–3074, Nov. 2011. ISSN 0960-1481. doi: 10.1016/j.renene.2011.03.028. URL <https://www.sciencedirect.com/science/article/pii/S096014811100142X>.
- [18] M. Hoffmann, L. Kotzur, D. Stolten, and M. Robinius. A Review on Time Series Aggregation Methods for Energy System Models. *Energies*, 13(3):641, Jan. 2020.

ISSN 1996-1073. doi: 10.3390/en13030641. URL <https://www.mdpi.com/1996-1073/13/3/641>.

- [19] M. Hoffmann, J. Priesmann, L. Nolting, A. Praktiknjo, L. Kotzur, and D. Stolten. Typical periods or typical time steps? A multi-model analysis to determine the optimal temporal aggregation for energy system models. *Applied Energy*, 304:117825, Dec. 2021. ISSN 0306-2619. doi: 10.1016/j.apenergy.2021.117825. URL <https://www.sciencedirect.com/science/article/pii/S0306261921011545>.
- [20] M. Howells, H. Rogner, N. Strachan, C. Heaps, H. Huntington, S. Kypreos, A. Hughes, S. Silveira, J. DeCarolus, M. Bazillian, and A. Roehrl. OSeMOSYS: The Open Source Energy Modeling System: An introduction to its ethos, structure and development. *Energy Policy*, 39(10):5850–5870, Oct. 2011. ISSN 0301-4215. doi: 10.1016/j.enpol.2011.06.033. URL <https://www.sciencedirect.com/science/article/pii/S0301421511004897>.
- [21] M. Ihlemann, I. Kouveliotis-Lysikatos, J. Huang, J. Dillon, C. O’Dwyer, T. Rasku, M. Marin, K. Poncelet, and J. Kiviluoma. SpineOpt: A flexible open-source energy system modelling framework. *Energy Strategy Reviews*, 43:100902, Sept. 2022. ISSN 2211-467X. doi: 10.1016/j.esr.2022.100902. URL <https://www.sciencedirect.com/science/article/pii/S2211467X22000955>.
- [22] L. Kotzur, P. Markewitz, M. Robinius, and D. Stolten. Impact of different time series aggregation methods on optimal energy system design. *Renewable Energy*, 117:474–487, Mar. 2018. ISSN 0960-1481. doi: 10.1016/j.renene.2017.10.017. URL <https://www.sciencedirect.com/science/article/pii/S0960148117309783>.
- [23] L. Kotzur, L. Nolting, M. Hoffmann, T. Groß, A. Smolenko, J. Priesmann, H. Büsing, R. Beer, F. Kullmann, B. Singh, A. Praktiknjo, D. Stolten, and M. Robinius. A modeler’s guide to handle complexity in energy systems optimization. *Advances in Applied Energy*, 4:100063, Nov. 2021. ISSN 26667924. doi: 10.1016/j.adapen.2021.100063. URL <https://linkinghub.elsevier.com/retrieve/pii/S266679242100055X>.
- [24] S. Le Bihan, T. Naegler, S. J. Schlosser, K.-K. Cao, M. Schicho, and F. Dunkel. Energy System Pathway Optimization Considering Market-Based and Geopolitical Technology Criticality. In H. A. Le Thi, T. Pham Dinh, and H. M. Le, editors, *Modelling, Computation and Optimization in Information Systems and Management Sciences*, pages 284–295, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-08381-4. doi: 10.1007/978-3-032-08381-4_22.
- [25] P. Mancarella. MES (multi-energy systems): An overview of concepts and evaluation models. *Energy*, 65:1–17, Feb. 2014. ISSN 0360-5442. doi: 10.1016/j.energy.2013.10.041. URL <https://www.sciencedirect.com/science/article/pii/S0360544213008931>.

- [26] P. Mancarella, G. Andersson, J. Peças-Lopes, and K. Bell. Modelling of integrated multi-energy systems: Drivers, requirements, and opportunities. In *2016 Power Systems Computation Conference (PSCC)*, pages 1–22, June 2016. doi: 10.1109/PSCC.2016.7541031. URL <https://ieeexplore.ieee.org/document/7541031/>.
- [27] S. Miehling, A. Hanel, J. Lambert, S. Fendt, and H. Spliethoff. Energy System Optimisation using (Mixed Integer) Linear Programming, Aug. 2023. URL <http://arxiv.org/abs/2308.01882>. arXiv:2308.01882 [math.OC].
- [28] M. Mostafa, D. Wensser, P. T. Baboli, and C. Becker. Unified Graph-Theoretic Modeling of Multi-Energy Flows in Distribution Systems. In *2025 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*, pages 1–5, Oct. 2025. doi: 10.1109/isgteurope64741.2025.11305330. URL <http://arxiv.org/abs/2509.06447>. arXiv:2509.06447 [eess.SY].
- [29] S. Pfenninger, A. Hawkes, and J. Keirstead. Energy systems modeling for twenty-first century energy challenges. *Renewable and Sustainable Energy Reviews*, 33:74–86, May 2014. ISSN 1364-0321. doi: 10.1016/j.rser.2014.02.003. URL <https://www.sciencedirect.com/science/article/pii/S1364032114000872>.
- [30] J. S. Schwarz, M. C. Pham, Q. T. Tran, and K. Heussen. Scaling Analysis in a Multi-Energy System. In *2023 Asia Meeting on Environment and Electrical Engineering (EEE-AM)*, pages 01–06, Nov. 2023. doi: 10.1109/EEE-AM58328.2023.10395068. URL <http://arxiv.org/abs/2410.17603>. arXiv:2410.17603 [cs.CE].
- [31] SciPy. Scipy spatial package: Delaunay algorithm. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.Delaunay.html>. Accessed: 2026-07-22.
- [32] H. Teichgraeber and A. R. Brandt. Time-series aggregation for the optimization of energy systems: Goals, challenges, approaches, and opportunities. *Renewable and Sustainable Energy Reviews*, 157:111984, Apr. 2022. ISSN 1364-0321. doi: 10.1016/j.rser.2021.111984. URL <https://www.sciencedirect.com/science/article/pii/S1364032121012478>.
- [33] A. Vandermeulen, G. Reynders, R. Salenbien, D. Saelens, and L. Helsen. SOURCES OF ENERGY FLEXIBILITY IN DISTRICT HEATING NETWORKS: BUILDING THERMAL INERTIA VERSUS THERMAL ENERGY STORAGE IN THE NETWORK PIPES.
- [34] M. Wetzel, H. C. Gils, and V. Bertsch. Green energy carriers and energy sovereignty in a climate neutral European energy system. *Renewable Energy*, (210):591–603, July 2023. ISSN 0960-1481. URL <https://elib.dlr.de/194896/>.
- [35] M. Wetzel, E. S. A. Ruiz, F. Witte, J. Schmugge, S. Sasanpour, M. Yeligeti, F. Miorelli, J. Buschmann, K.-K. Cao, N. Wulff, H. Gardian, A. Rubbert, B. Fuchs, Y. Scholz, and H. C. Gils. REMix: A GAMS-based framework for optimizing

- energy system models. *Journal of Open Source Software*, 9(99):6330, July 2024. ISSN 2475-9066. doi: 10.21105/joss.06330. URL <https://joss.theoj.org/papers/10.21105/joss.06330>.
- [36] D. W. Wu and R. Z. Wang. Combined cooling, heating and power: A review. *Progress in Energy and Combustion Science*, 32(5):459–495, Sept. 2006. ISSN 0360-1285. doi: 10.1016/j.pecs.2006.02.001. URL <https://www.sciencedirect.com/science/article/pii/S0360128506000244>.
- [37] W. Xu, Y. Ruan, Y. Li, F. Xiao, T. Xu, Y. Yao, and H. Meng. Investigation and evaluation of strategic residential heat and transport electrification for supporting deep grid decarbonization. *Renewable Energy*, 256:124566, Jan. 2026. ISSN 0960-1481. doi: 10.1016/j.renene.2025.124566. URL <https://www.sciencedirect.com/science/article/pii/S096014812502230X>.