Stochastic Programming for Energy Models: A Blended Cross-Scenario Representative Periods Approach

L.A.A. (Lotte) Kremer

Technische Universiteit Delft

TU Delft

Stochastic Programming for Energy Models: A Blended Cross-Scenario Representative Periods Approach

by

L.A.A. (Lotte) Kremer

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Wednesday the 25th of June, 2025

Student number: 4861957
Supervised by: Prof. dr. M.M. de Weerdt
Dr. G.A. Morales-España
Dr. G. Neustroev

TU Delft Delft University of Technology

Abstract

In this thesis, we investigate how representative periods can be used as a temporal reduction technique for stochastic programming formulations of large-scale energy models. We specifically apply this to generation expansion planning. The focus is on cost-efficient decisions as well as on robust decisions, especially since the latter is why uncertainty is incorporated in most energy models. For this, we build on recent work that uses instead of traditional clustering methods like k-means or k-medoids, a method to obtain a convex or bounded conical hull over the periods. We prove that when inter-period constraints are ignored and a perfect hull is found, we can ensure that a feasible solution for the original model exists, obtained from the optimal decision variables of the reduced model. The proof is suitable for both the traditional way of using representative periods in stochastic programming, as for an approach that shares representative periods across scenarios instead of making them scenario-dependent. We show that a greedy implementation of the hull approach can outperform standard clustering methods in terms of costs in small case studies when a valid hull is found. The near-optimal costs can be obtained with a low number of representatives, largely due to the absence of loss of load due to the hull methods. However, in more complex cases, such as a European-scale model with high renewable shares, finding such a hull with the greedy algorithm proves difficult. To address this, we propose adding extreme periods either beforehand or afterwards, and applying a blended weights approach to reduce conservativeness while maintaining feasibility. Our results on small case studies demonstrate that extreme representatives can significantly reduce loss of load, although not always at lower cost when applied to the large European case study. These findings suggest that targeted selection of extremes, a right weighting approach and improved hull approximations offer a promising direction for enabling scalable, robust planning under uncertainty.

Contents

1

Introduction

In the energy sector, it is crucial to ensure that current investments in energy production, transmission, and storage are sufficient to meet future demand. Optimization models play an important role in supporting such decisions by identifying cost-effective investment strategies and dispatch solutions that satisfy future energy needs while accounting for constraints related to transmission capacity, storage limitations, and availability of generation technologies. A prominent example is the "Ten-Year Network Development Plan (TYNDP)", published every two years by the European Network of Transmission System Operators for Electricity (ENTSO-E) [8]. Using optimization models and multiple future scenarios, the TYNDP identifies system needs and infrastructure requirements across Europe. However, while these models are powerful in finding optimal solutions, their scale and complexity make them computationally demanding: the most recent TYNDP scenarios report shows they had to limit the analysis to just three out of thirty-five scenarios for computational reasons [8].

One of the key challenges in these energy models is that, while decisions need to be made for the coming decades, they must satisfy constraints defined on a very fine temporal and spatial scale, often with strict technical requirements. For example, the system should be able to meet electricity demand at every hour over several years. This becomes even more computationally demanding with the growing share of renewable energy sources [27], whose availability fluctuates depending on weather conditions. As a result, we are dealing not only with demand data at high temporal resolution, but also with data on the availability of renewable generation technologies and related operational constraints at that same fine scale. Making optimal decisions while incorporating this level of temporal detail is not feasible with standard computing resources. To make the problem tractable, models often introduce reductions, lowering technical detail in the formulation, spatial resolution, or the temporal dimensions [21].

While energy system optimization problems are already large on their own, realistic solutions must also account for the fact that future conditions are inherently uncertain. This need gives rise to the field of optimization under uncertainty in energy systems [29]. The uncertainty shows up in many parts of the model, but here we focus on two types of it. First, the demand at each location is based on forecasts, which can vary depending on weather patterns or broader trends in electricity use. Even more uncertain, however, are renewable energy sources. Their availability depends heavily on weather conditions, and predicting future weather patterns in advance is difficult. Still, a system that remains reliable under a range of possible futures is highly desirable.

There are multiple ways to include uncertainty in demand or the availability of renewable generation technologies within an optimization model, as described by Roald et al. [29]. In this thesis, we focus specifically on the use of stochastic programming. Stochastic programming extends the deterministic version of the problem by optimizing the expected value of the objective across multiple scenarios, rather than just one. In the context of energy models, this often translates to minimizing expected system costs under uncertainty. Important for two-stage stochastic programming is the separation between two types of decision variables. First-stage decisions are made before the uncertainty is revealed, these typically correspond to long-term investment choices. Second-stage decisions represent uncertain short-term operational actions, such as dispatching generation units or managing storage. These are decided upon after the scenario has unfolded. This structure makes stochastic programming particularly suitable for energy system planning, where large upfront investments must be made without knowing future demand and availability values.

Incorporating multiple scenarios can improve solution quality under uncertainty, but stochastic programming also adds constraints and decision variables, making models more computationally intensive to solve. As mentioned earlier in the context of the TYNDP, ENTSO-E ran only three out of thirty-five scenarios. Although they had more scenarios available, they had to make a selection due to computational availability [8]. In such a case, scenario reduction techniques are used to limit the number of scenarios while preserving key system characteristics. While a lot of research exists in this field [9], even with a reduced set of scenarios, model size remains large. This reinforces the need for additional techniques, beyond scenario reduction, that can decrease computational effort while maintaining the robustness of the system.

Micheli et al. [20] combines two such techniques to ease the computational burden in stochastic energy models. The first, Benders decomposition, is a solving technique that decomposes the problem to solve it iteratively, which eventually converges to the optimum. The second technique is temporal reduction through representative periods. These and related temporal reduction techniques are well-studied in deterministic energy modeling. These approaches aim to identify a small set of time slices, such as days or weeks, that capture the essential temporal variability of the full dataset. Clustering algorithms like k-means or k-medoids are often used to group similar periods and to select a small number of representative periods as input for a reduced model [33]. However, as Pfenninger [26] highlights, the performance of clustering methods depends heavily on input data, and there is no universally optimal technique. One other limitation is that clustering tends to center around averaged values, systematically excluding outliers. As a result, models based on such representative periods may fail to capture rare but critical events, leading to solutions that are cost-efficient on average but vulnerable under extreme circumstances. This is undesirable, as reliability is critical in energy networks and power outages must be avoided. To address this issue, some approaches introduce extreme, worst-case periods after clustering [18], while others, such as in the package TulipaClustering.jl [23], focus on selecting representative periods that span the extremes of the dataset. They use two greedy hull methods for this.

This thesis builds on insights from deterministic modeling and extends them to the stochastic setting. The goal is to develop an approach that is both robust to uncertainty and computationally tractable. While stochastic programming offers a structured way to handle uncertainty, additional temporal reduction is needed to keep model complexity manageable. Although representative periods have been used in stochastic models, they are typically applied separately within each scenario. This thesis instead proposes a cross-scenario approach, selecting periods across all scenarios jointly. In addition to using representative periods more efficiently, we aim to improve robustness by focusing on methods that explicitly maintain feasibility. To this end, we apply the greedy hull algorithms proposed in TulipaClustering.jl [23] and provide a theoretical justification for its use in a stochastic programming context: if the selected representatives are a convex or bounded conical hull of the initial period set, then any solution feasible for the representatives should also lead to a feasible solution for the full dataset. We address some of the practical limitations of the greedy hull methods by adding artificially created representatives containing worst-case values. The methods developed are tested on a European case study with a high share of renewable generation, where uncertainty plays a central role and the benefits of stochastic programming are especially relevant.

The work in this thesis addresses the following research questions:

1. How can representative periods be shared across scenarios?
2. How can we ensure that solutions obtained from a reduced model using representative periods lead to a solution that remains feasible in the full, unreduced model?
3. Under which conditions are k-means, k-medoids, and the greedy hull methods preferable in terms of feasibility and optimality?
4. Can we improve the greedy hull methods by constructing artificial representative periods, especially when a true convex or bounded conical hull cannot be formed with the available number of representatives?
5. What are the advantages and limitations of using these proposed methods in realistic, large-scale case studies?

This thesis is structured as follows. Chapter 2 reviews related work, focusing on both the incorporation of uncertainty in energy models and the use of reduction techniques to manage model size in deterministic settings. Chapter 3 introduces the contributions of this thesis, where temporal reduction techniques are applied in a stochastic programming context using the proposed cross-scenario method. This chapter also presents the full mathematical formulation. This is done specifically for generation expansion planning (GEP), which is an example of an energy model, but all methods mentioned in this thesis can be applied to more general settings as well. Chapter 4 provides a theoretical foundation for maintaining feasibility in the original model while using the reduced model. It proves that, under the condition that the representative periods are a convex or bounded conical hull of the full set, any solution to the

reduced model can be transformed into a feasible solution for the original model. Chapter 5 compares the greedy bounded conical or convex hull method to traditional clustering techniques through selected case studies, evaluating their feasibility and optimality under an increasing variability. Practical limitations observed in this approach are further explored in Chapter 6, which proposes ways to maintain feasibility by generating artificial representative periods. These methods are first introduced in a two-dimensional setting, and then extended to higher-dimensional spaces. Chapter 7 presents a European case study with a high share of renewables and many scenarios, serving as a realistic example on which we can test the proposed methods. Finally, Chapter 8 summarizes the conclusions and outlines directions for future research.

2

Related work

This chapter reviews related work on the two main themes of this thesis: temporal reduction techniques and uncertainty in large-scale energy models. We first define what is considered an energy model to set the scope of this research, in Section 2.1. Then we look at potential reduction techniques in the deterministic setting focusing on temporal reduction methods which are widely researched in the literature. Section 2.2 discusses the main approaches. Next, we look at uncertainty. While recognized as a critical factor in energy planning, it is often omitted in practice due to computational challenges. Still, many studies have shown that incorporating uncertainty can lead to more robust and cost-effective decisions than models based solely on average scenarios. In Section 2.3, we review existing work on modeling uncertainty in energy systems, with a focus on how to deal with computational complexity.

2.1. Energy models

Energy models refer to any form of optimization or simulation that represents the flow and transformation of energy, from extraction to end use, as described by Pfenninger et al. [27]. While their work outlines a broad spectrum of existing models, for the purpose of this thesis, we focus specifically on what they describe as energy system optimization models and energy system simulation models. These are particularly relevant because they are affected by the key challenges this thesis addresses: how to balance temporal resolution with other details and how to deal with uncertainty in a computationally tractable way.

Within this category, models vary in focus and formulation, but generally share a common structure. At their core, they must satisfy demand at specific locations while adhering to technical and operational constraints, for example, limits on ramping, transmission, and production. In addition to these short-term operational decisions, many models include long-term investment decisions in generation, storage, or transmission infrastructure. These decisions must be made in advance and must be integrated with the detailed operational constraints, thereby adding to the model's size and complexity. Since these models aim to satisfy future operational constraints, accounting for uncertainty is essential, but doing so further increases computational demands. This makes the research in this thesis especially relevant for large-scale energy models.

While the methods proposed in this thesis are broadly applicable across different types of energy models, the experiments focus on generation expansion planning (GEP). The goal in this type of model is to determine which generation assets to invest in, and where. This focus allows for a consistent case study framework, but the techniques themselves are not tied to GEP and could be transferred to other model types with minimal adjustments. As such, the related work reviewed below covers a broad range of energy models, not only GEP. This helps place the proposed methods within the wider context of research aimed at simplifying large-scale energy models while maintaining their ability to handle uncertainty and system complexity. For a detailed review of GEP we refer to Koltsaklis and Dagoumas [13].

2.2. Reduction techniques in a deterministic setting

Energy models typically have high spatial and temporal resolution and can include many technical details. A common approach for reduction is to aggregate spatial regions and include only a few time slices per year. However, more detail has become increasingly necessary due to two major changes in energy systems: higher fluctuations in demand

and the growing share of renewables, which adds variability to supply [27]. In this section, we look at reduction techniques that aim to maintain accuracy while reducing model size, with a focus on temporal reduction. This can free up space to include more detail in other aspects of the model, such as spatial details or technology-specific constraints. Depending on the case, however, it may also be necessary to simplify those aspects. In such cases, it can be useful to explore more efficient model formulations [36], use relaxations on integer decision variables, or apply spatial aggregation techniques [25, 32].

Load levels Before the rise of renewables, a common approach in temporal reduction was to use load levels. Load levels are obtained by the load duration curve. The demand values are ordered by value and frequency, creating blocks that represent how often each level of demand occurs. For each range of demand values load blocks are obtained and then used to replace the hourly demand data. As described by Wogrin et al. [37], the downside is that the chronology is lost. Some operational constraints, such as ramping limits or storage behavior, depend on the chronological order of time steps. Removing the order and the associated transitions means the model cannot account for the variability. The problem becomes more pronounced with renewable energy sources, whose output depends heavily on weather and varies across hours and seasons. Losing the chronological structure can remove patterns in the data, such as correlations between demand and availability in renewable generation technologies, or the timing of extreme events. Moreover, the high variability causes extra needs in ramping, not captured when using load levels [18, 21, 27, 37].

System states To address this, several new approaches have emerged that include chronological order and the availability of renewable generation technologies. One such approach is proposed by Wogrin et al. [37], who introduce the concept of system states. The idea is to cluster the input data into a limited number of representative states, each defined by a typical combination of demand and renewable availability. These states are then used to replace the full set of hourly observations in the optimization model. To retain some chronology, a transition matrix is added that captures the frequency of moving from one state to another. This allows the model to include constraints that depend on chronology, even though the full temporal sequence is no longer explicitly modeled. The system is then solved over these representative states and their transitions, rather than over the original time series. While this preserves some form of chronology, it can still be too limited for models that rely more heavily on sequential constraints, for example when including storage. In those cases, more detailed chronological information is needed, but can increase runtime. Tejada-Arango et al. [35] propose an extension to apply the system states in models with storage.

Representative periods Another approach that is currently widely applied is the use of representative periods [14, 18, 21, 26, 28, 30, 33–35]. The general idea consists of a few steps, with different possible directions in each. We first outline the basic process before diving into the variations. Just as with system states, the hourly input data is clustered, but the clustering is based on groups of time steps (periods) instead of individual time steps. A period can be a day or a week, for example. The original input data is thus first divided into these periods. A few representative periods are then created or selected to best represent the original ones. Each representative is assigned a weight based on how many of the original periods it represents. The model is then only solved for these selected periods.

The main difference between methods lies in how the representative periods are selected. This is mostly done through clustering algorithms. Here, typical approaches include k-means [14, 18, 26, 33, 35], k-medoids [14, 18, 30, 33], and hierarchical clustering [14, 21, 26, 28]. In studies, the distinction is often made between centroid-based and medoid-based methods. The centroid of a cluster is the mean position of all points within the cluster, representing an average location that may not correspond to an actual data point. In contrast, the medoid is the data point within the cluster that is closest to the center, ensuring that the representative is an actual sample from the dataset. Some studies prefer centroid-based methods like k-means, as they result in underestimation of the objective, which is even proven to be a lower bound for the optimal solution of the original problem [18, 33]. Others argue that medoid-based clustering captures the actual variability better, since it does not average out time series as much [14, 33]. But in general, there is no consistent winner among the different methods, the best algorithm depends heavily on the specific case [34]. This is agreed upon by another large comparison carried out by Pfenninger [26]. Since there is a clear distinction in behavior between centroid-based and medoid-based methods, these two are both used in our experiments.

It is also highlighted in many papers that the clustering algorithm itself is only one part of the process [26, 30, 34]. Other choices can have just as much influence on results. We will discuss some of these choices here. First, a decision must be made about what data to cluster on. Most studies use historical time series, but variations exist. For example, some use net-demand profiles that incorporate ramping effects [30], or even cluster based on investment decisions instead of input data [18]. Second, normalization is a critical step before clustering, especially when demand and availability are on different scales. Techniques vary, with the most common being z-normalization or scaling by the

maximum value. Normalization can be applied to the full dataset or element-wise, such as per location or per time step [33, 34]. Third, the distance metric used in clustering can also impact the result. Most papers rely on the squared euclidean distance. Teichgraeber and Brandt [34] also mention that alternatives like the manhattan distance do not result in better performance. More advanced metrics like dynamic time warping or shape-based distance have also been tested, especially since they are better suited to compare time series. They are implemented with clustering by using DTW-Barycenter Averaging or k-shape clustering, but none show a consistent advantage. Finally, the weighting step can differ as well. While some approaches assign weights based on cluster sizes, others, like Scott et al. [30], adjust weights using net-demand to better match the original objective function.

In addition to the previously mentioned model variations, there are also differences in the period length and the coupling between periods. The motivation for using representative periods over other methods is to better capture the chronological information present in the input data. If no coupling constraints are present, a period of just one time step could be used and there would be no advantage in using more time steps per period. However, when coupling constraints like ramping or storage are included, most papers opt for a 24-hour period, while some suggest using one week to be able to capture even more information about temporal patterns. Using periods of one week increases computational time, therefore, longer periods are generally avoided [34]. One important limitation of the implementation of representative periods is that inter-period constraints are typically ignored. Some argue this is acceptable, as it is the most commonly used approach in the literature [18], while others explore methods to add coupling constraints back into the reduced model [35]. For now, we will focus in this research on finding and proposing the best technique when ignoring inter-period constraints, but future work could explore combining the methods developed here with techniques that reintroduce inter-period coupling.

In summary, there is a wide range of variation across studies using clustering techniques, and most methods still show significant error in some settings. One common addition to improve accuracy is the inclusion of extreme days. In a broad experimental study, Pfenninger [26] shows that including extremes can substantially improve model performance. Teichgraeber and Brandt [34] also note that when variables like renewable availability are likely to be binding in the constraints, it becomes even more important to include extreme conditions. There are different ways to do this. Some studies identify extremes beforehand, using simple heuristics like maximum demand or minimum availability [14, 26, 28, 30]. Others add them after clustering by checking which days result in the most energy not served under the chosen investment decisions [18]. Extreme days can be appended to the representative set [18], replace cluster centers [14], or even be forced into the clustering process [30]. A technique that emphasizes extreme cases involves replacing the clustering process entirely by selecting periods located on the boundary of the dataset, rather than identifying central cluster points. In the TulipaClustering.jl package, this is currently implemented by finding a convex or bounded conical hull for the set of periods and selecting the corner points of this hull as representative periods [23]. Since computing such a hull with a limited number of corner points can be computationally intensive, or even infeasible depending on the samples, an approximate greedy algorithm is used. The core idea is that when the selected representatives form a convex or bounded conical hull over the full period set, any original period can be expressed as a convex or bounded conical combination of the representatives. The resulting weights can be used to refine the estimation of objective functions. The rationale behind choosing boundary points is that they collectively span the feasible space, ensuring coverage of all periods within the hull. This thesis formalizes that intuition with a theoretical proof, and further compares the algorithmic approach proposed by Neustroev et al. [23] against more traditional clustering techniques for selecting representative periods.

To conclude, there is a wide range of options when it comes to selecting and using representative periods. Many different clustering methods, normalization approaches, distance metrics, and weighting schemes have been proposed, and none clearly outperform the others in all cases. In this thesis, we focus on the bounded conical and convex hull method, that aim to provide stronger guarantees by incorporating the idea of extremes more explicitly. We compare it to centroid- and medoid-based approaches (k-means and k-medoids). To make the approach practical, we make specific choices for normalization, distance metric, and period length, which are discussed in more detail in the methodology chapter. For now, we explore how the previously discussed methods can be incorporated into a setting with uncertainty, which we will dive into in the next section.

2.3. Uncertainty in energy models

Uncertainty in energy models can come from many sources and therefore different models are designed to incorporate different types of uncertainty. Here, we follow the overview of methods presented in the review by Roald et al. [29], along with insights from individual papers. The most commonly mentioned source of uncertainty is electricity demand [6, 9, 24, 31, 32, 38, 39], followed by the uncertainty in the availability of renewable generation technologies

[12, 24, 32, 38, 39], and in some cases, electricity prices or investment and operational costs [6, 20, 31]. While these sources differ, all papers agree that incorporating uncertainty leads to better decisions compared to relying on an average deterministic scenario. However, it also introduces significant computational challenges. In this thesis, we focus on uncertainty in demand and the availability of renewable generation technologies, as these types of models are widely studied, have accessible scenario data, and can benefit most from temporal reduction due to their high hourly variability.

Although there are many ways to model uncertainty, most approaches found in the literature fall into one of two main categories: robust optimization and stochastic programming. Robust optimization focuses on finding the best decision variables under worst-case realizations within a defined uncertainty set. This uncertainty set typically defines an interval of possible values that the uncertain parameters can take. While this makes the model more conservative, the degree of conservatism can be adjusted by controlling the size or shape of the uncertainty set [6, 29]. Dehghan et al. [6] argue that this approach is especially useful when it is difficult to construct scenarios, or when the uncertainty space is too large to sample properly. In stochastic programming, the uncertainty is represented by a finite set of scenarios with associated probabilities. These can be independent and identically distributed samples from a known distribution, or discrete samples with known probabilities. The goal is usually to minimize the expected value of the objective, taking into account both first-stage (investment) decisions and second-stage (operational) decisions, which adapt once the scenario is revealed. The main drawback is that the size of the model grows linearly with the number of scenarios. Moreover, it can be difficult to ensure that a limited set of scenarios captures the full probability space.

Both method categories tend to result in large-scale models that require simplification. For both robust optimization and stochastic programming, decomposition techniques such as Benders decomposition can be used to solve the problem iteratively by decomposing the problem into a master problem and many independent subproblems related to the second-stage variables [6, 9, 20, 38]. In stochastic programming, it is also common to reduce the number of scenarios. Scenario reduction techniques, such as clustering or filtering, are used to select a smaller number of representative scenarios that best capture the characteristics of the full set [9, 12, 24, 31, 32]. Although a combination of decomposition and scenario reduction techniques can help reduce runtime, solving the full model often remains challenging. We argue that the goal should be to retain as much relevant information from the scenarios as possible. In particular, when uncertainty lies in the same variables as the hourly time series, such as availability of generation technologies and demand, it is natural to extend temporal reduction techniques to the stochastic setting.

While Micheli et al. [20] explores the use of representative periods in stochastic settings, their approach applies them independently to each scenario. That is, representative periods are selected separately within each scenario, and no information is shared across scenarios. We argue that periods can be selected more efficiently when scenarios are treated jointly, especially since the variation across periods and scenarios is driven by demand and availability. We therefore propose a cross-scenario selection method, outlined in Chapter 3. Our aim is to develop reduction techniques that preserve as much relevant information as possible across all scenarios. The goal is to produce a reduced model that yields solutions close to the original optimum while maintaining feasibility for all full-scenario constraints. To this end, we propose and implement the convex and bounded conical hull methods discussed earlier, alongside more traditional clustering-based approaches.

3

Representative periods in stochastic programming for generation expansion planning

In Chapter 2, we identified temporal reduction, particularly through representative periods, as a promising approach to reduce model size. This chapter presents the methods used to implement this approach in a stochastic programming context. Some of the technical details that were previously introduced at a high level are now described more formally.

We begin in Section 3.1 by introducing the generation expansion planning (GEP) formulation used throughout this thesis. This serves as a running example to illustrate the implementation of the proposed methods. Section 3.2 explains how representative periods are integrated into the stochastic GEP formulation, both in the standard per-scenario approach and in the proposed cross-scenario method. In Section 3.3, we present the mathematical and practical details of the techniques used to construct representative periods, including normalization, selection, and calculating weights. Finally, Section 3.4 describes the evaluation metrics and experimental setup used to assess trade-offs between accuracy, computational efficiency, and robustness.

3.1. Stochastic programming formulation for generation expansion planning

As an example of a large-scale energy model, we begin our research with GEP. This involves selecting investments in generation technologies at various locations to meet demand. Uncertainty arises both in the demand and in the availability of renewable energy sources. Both parameters are often represented with hourly input data spanning a long time horizon. This complexity can make the model intractable, particularly when stochastic programming is applied. As noted by Gandulfo et al. [9], runtime issues often lead to selecting only a subset of the scenarios. This makes GEP a valuable case study for this thesis.

The formulation is presented in Section 3.1.2, including the main constraints typically included in this type of problem [9]. For an overview of possible additions and variations, the review by Koltsaklis and Dagoumas [13] can be helpful.

3.1.1. Model components

This subsection further explains the formulation of GEP. Before focusing on the model formulation in Subsection 3.1.2, we first define the underlying model components: the sets, parameters, and decision variables.

Sets

Table 3.1 shows the sets used in GEP, along with their descriptions and associated indices. Since we will later reduce the model with representative periods, it is helpful to present it in a period-based formulation from the beginning. The duration of one period is defined as $|\mathcal{H}|$, where $\mathcal{H}$ represents the set of time steps within a given period. One

period typically consists of 24 hours, but we will further discuss this when explaining the use of representative periods in Section 3.2. Each variable and parameter that is time step-based will be indexed by the scenario, the period within that scenario and the hour within that period. This means that the combination (s, p, h) will be reused often. To simplify, we have introduced an additional set $\mathcal{I}$. This set acts as an abstract index for the time and allows for more flexible representations of the indices in different contexts. This is especially important later on when using representative periods in the formulations. A similar approach is applied for $e = (n, n')$, representing transmission lines.

Set	Description	Index
$\mathcal{N}$	Locations	n
$\mathcal{G}$	Generation technologies	g
$\mathcal{P}$	Periods	p
$\mathcal{H}$	Time steps within one period	h
$\mathcal{S}$	Scenarios	s
$\mathcal{NG} \subseteq \mathcal{N} \times \mathcal{G}$	Generation technologies available per location	(n, g)
$\mathcal{I} = \mathcal{S} \times \mathcal{H} \times \mathcal{P}$	Moment in time defined by the period, time step and scenario	$i = (s, p, h)$
$\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$	Transmission lines between different locations	$e = (n, n')$

Table 3.1: Sets and their indices used in the GEP Model

Parameters

The parameters used in the model are summarized in Table 3.2. The first two rows describe time-dependent parameters: demand and availability. The availability for non-renewable generation technologies is always equal to 1.0. Following these are three parameters related to the costs considered in the model, including investment and operational costs. The next four parameters define the technical constraints, ramping limits, and transmission capacities. We consider a model in which import and export capacities can differ. The import capacity of a transmission line $e = (n, n')$ gives a lower bound to the flow, as import is negative flow, while the export capacity is an upper bound and refers to the flow going from n to n'. An additional parameter used is the annualization factor $AF = \frac{8760}{|\mathcal{P} \times \mathcal{H}|}$. When the input data does not cover a full year, this factor is applied to the operational costs to ensure comparability with the annualized investment costs. Finally, the parameter π_s represents the probability of scenario s, which is incorporated into the objective function to calculate the expected operating costs. Those parameters satisfy $\sum_{s \in \mathcal{S}} \pi_s = 1$.

Parameter	Description	Unit	Domain
$D_{n,i}$	Demand for location n, at time i	MW	$\mathbb{R}_{\geq 0}$
$A_{n,g,i}$	Availability as a fraction of the maximum possible capacity for generation technology g at location n and time i	1/unit	$[0, 1]$
$I_{n,g}$	Annualized investment costs for generation technology g at location n	EUR/MW	$\mathbb{R}_{\geq 0}$
$V_{n,g}$	Operational cost for each MWh produced by generation technology g at location n	EUR/MWh	$\mathbb{R}_{\geq 0}$
V^{loss}	Value of lost load	EUR/MWh	$\mathbb{R}_{\geq 0}$
$U_{n,g}$	Maximum capacity per unit of generation technology g at location n	MW/unit	$\mathbb{R}_{\geq 0}$
$R_{n,g}$	Ramping rate for generation technology g at location n	1/unit	$[0, 1]$
L_e^{exp}	Export capacity of the transmission line e	MW	$\mathbb{R}_{\geq 0}$
L_e^{imp}	Import capacity of the transmission line e	MW	$\mathbb{R}_{\geq 0}$
π_s	Probability of occurrence of scenario s	1	$[0, 1]$
AF	Annualization factor for when the periods do not cover a full year	1	$\mathbb{R}_{\geq 0}$

Table 3.2: Parameters, their units, and domains used in the GEP Model

Decision variables

The decision variables, summarized in Table 3.3, are divided into first-stage and second-stage variables, reflecting the structure of a stochastic programming model. The first-stage variables capture decisions made before the realization of any scenario, specifically investment choices and their associated costs. The second-stage variables are scenario-dependent and include production levels, loss of load, transmission flows, and the resulting operational costs. The loss of load variables act as slack variables. They are penalized in the objective function and indicate the amount of unmet demand under the current investment decisions. Throughout this thesis, we will often refer to a solution as leading to an infeasible outcome, by which we mean that demand cannot be fully met.

Although the decision variables for the investments originally are limited to have an integer value, we relax them to lower computational complexity. This is the common practice [18]. For the theoretical results in Chapter 4, it does not matter for the proof whether this relaxation is used.

Variable	Description	Unit	Domain
c^{inv}	Total investment costs	EUR	$\mathbb{R}_{\geq 0}$
c_s^{op}	Operational costs for scenario s, annualized	EUR	$\mathbb{R}_{\geq 0}$
$i_{n,g}$	Amount of units invested in generation technology g at location n	Units	$\mathbb{Z}_{\geq 0}$
$prod_{n,g,i}$	Production of generation technology g at location n and time i	MW	$\mathbb{R}_{\geq 0}$
$f_{e,i}$	Flow in the transmission line e at time i	MW	$\mathbb{R}$
$l_{n,i}$	Loss of load in MW at location n and time i	MW	$\mathbb{R}_{\geq 0}$

Table 3.3: Decision variables, their units, and domains used in the GEP Model

3.1.2. Full model formulation

The mathematical formulation of the optimization problem is given in (3.1). To better understand its meaning, we will break it down and discuss each part separately. First, we look at the objective function, and then we go over the constraints. The model includes four types of constraints: balance, transmission, capacity, and ramping constraints.

$$\min \quad c^{\text{inv}} + AF\left(\sum_{s\in\mathcal{S}} \pi_s \cdot c_s^{\text{op}}\right) \tag{3.1a}$$

$$\text{s.t.} \quad c^{\text{inv}} = \sum_{(n,g)\in\mathcal{NG}} I_{n,g} \cdot U_{n,g} \cdot i_{n,g} \tag{3.1b}$$

$$c_s^{\text{op}} = \sum_{\substack{p\in\mathcal{P}\\ h\in\mathcal{H}}} \left(\sum_{\substack{(n,g)\\ \in\mathcal{NG}}} V_{n,g} \cdot prod_{n,g,(s,p,h)} + \sum_{n\in\mathcal{N}} V^{\text{loss}} \cdot l_{n,(s,p,h)} \right) \qquad \forall s \in \mathcal{S} \tag{3.1c}$$

$$\sum_{g|(n,g)\in\mathcal{NG}} prod_{n,g,i} + \sum_{n'|(n',n)\in\mathcal{L}} f_{(n',n),i} - \sum_{n'|(n,n')\in\mathcal{L}} f_{(n,n'),i} + l_{n,i}$$

$$= D_{n,i} \qquad \forall n \in \mathcal{N}, i \in \mathcal{I} \tag{3.1d}$$

$$f_{e,i} \geq -L_e^{\text{imp}} \qquad \forall e \in \mathcal{L}, i \in \mathcal{I} \tag{3.1e}$$

$$f_{e,i} \leq L_e^{\text{exp}} \qquad \forall e \in \mathcal{L}, i \in \mathcal{I} \tag{3.1f}$$

$$prod_{n,g,i} \leq A_{n,g,i} \cdot i_{n,g} \cdot U_{n,g} \qquad \forall (n,g) \in \mathcal{NG}, i \in \mathcal{I} \tag{3.1g}$$

$$prod_{n,g,(s,p,h)} - prod_{n,g,(s,p,h-1)} \leq R_{n,g} \cdot U_{n,g} \cdot i_{n,g} \qquad \forall (n,g) \in \mathcal{NG}, p \in \mathcal{P}, h \in \mathcal{H}\backslash\{1\}, s \in \mathcal{S} \tag{3.1h}$$

$$prod_{n,g,(s,p,h)} - prod_{n,g,(sp,h-1)} \geq -R_{n,g} \cdot U_{n,g} \cdot i_{n,g} \qquad \forall (n,g) \in \mathcal{NG}, p \in \mathcal{P}, h \in \mathcal{H}\backslash\{1\}, s \in \mathcal{S} \tag{3.1i}$$

$$prod_{n,g,(s,p-1,|\mathcal{H}|)} - prod_{n,g,(s,p,1)} \leq R_{n,g} \cdot U_{n,g} \cdot i_{n,g} \qquad \forall (n,g) \in \mathcal{NG}, p \in \mathcal{P}\backslash\{1\}, s \in \mathcal{S} \tag{3.1j}$$

$$prod_{n,g,(s,p-1,|\mathcal{H}|)} - prod_{n,g,(s,p,1)} \geq -R_{n,g} \cdot U_{n,g} \cdot i_{n,g} \qquad \forall (n,g) \in \mathcal{NG}, p \in \mathcal{P}\backslash\{1\}, s \in \mathcal{S} \tag{3.1k}$$

Objective function (3.1a)

The objective for GEP is to minimize the total cost, which consists of the investment costs (first-stage variables) and the expected operational costs (second-stage variables). The expected operational costs are calculated by multiplying the operational cost of each scenario by its probability and summing these values. The operational costs are annualized using the factor AF.

The investment costs, further defined in (3.1b), are computed by multiplying the number of invested units by the capacity of each unit and its cost per MW. The operational costs, shown in (3.1c), include two components: the variable costs of production and the cost of lost load, both of which depend on the scenario.

Balance constraint (3.1d)

The balance constraint ensures that, at every moment in time, the demand at a given location matches the energy available at that location. The available energy is calculated as the sum of the production from generation technologies at that location and the energy imported, minus the energy exported. This constraint is softened by allowing a loss of load, which acts as a slack variable. This comes with an extra cost.

Transmission constraints (3.1f) - (3.1e)

The flow on a transmission line is limited by both its export and import capacities. Export capacity for a transmission line $f_{=}(n,n')$ provides an upper bound for positive flow (energy flowing from n to n'), while import capacity sets a lower bound on the negative flow (energy flowing from n' to n).

Capacity constraint (3.1g)

The production of each generation technology is constrained by its availability, the number of units invested at a specific location, and the capacity of each unit. This ensures that production does not exceed the installed capacity.

Ramping constraints (3.1h) - (3.1k)

Lastly, we consider ramping constraints. For each unit of generation technology, there is a limit on how much its production can increase or decrease between consecutive time steps, depending on the specific technology and location. In the previous constraints, we indexed the time using $i \in \mathcal{I}$. For the ramping constraints, we shift to the fully written-out indices (s,p,h). This change allows for a clearer distinction between intra-period constraints and inter-period constraints. The intra-period constraints are the ramping constraint within a period, the ramping up constraint is shown in (3.1h) and the ramping down constraint is shown in (3.1i). The inter-period ramping constraints relate to the ramping between consecutive periods. These are shown in (3.1j) and (3.1k).

3.2. Applying representative periods in stochastic programming

Since representative periods are not yet widely used in stochastic programming formulations, we first explain how they can be applied once identified. This section focuses on the reduced model formulation. The details of the selection process will follow in Section 3.3.

In this thesis, we compare two approaches for applying representative periods: the per-scenario method, as used by Micheli et al. [20], where representative periods are selected independently for each scenario, and the cross-scenario method we propose, where periods are shared across scenarios. For both approaches, we first describe the general idea and then present the mathematical formulation in the context of the GEP model, so the implementation becomes immediately clear. To support this, we also define the input data on which clustering is performed.

General formulation of representative periods in GEP

As described in Chapter 2, our goal is to identify a small set of representative periods that can replace the original, much larger set of periods. This original set is composed of multiple periods per scenario. To abstract from the scenario notation, we define an index set $\mathcal{J}$, which can be any subset of $\mathcal{S} \times \mathcal{P}$. Later, we clarify how $\mathcal{J}$ is defined in the per-scenario and cross-scenario methods.

In the GEP model, uncertainty and hourly variability arise both in the demand parameters, $D_{n,i}$, and in the availability parameters, $A_{n,g,i}$. We want to find representative periods that capture the variability in both of these inputs. Let $\mathcal{C}$ be the set of vectors we use for clustering, defined as:

$$\mathcal{C} = \left\{ \mathbf{q}_j \mid \mathbf{q} \in \mathbb{R}^{NG\times H + N\times H}, j \in J \right\}. \tag{3.2}$$

Here, $NG = |\mathcal{NG}|$ is the number of generator-location combinations, $H = |\mathcal{H}|$ is the number of hours per period, and $N = |\mathcal{N}|$ is the number of locations. Each vector $\mathbf{q}_j$ includes both availability and demand, defined as:

$$\mathbf{q}_j = \begin{bmatrix} \mathbf{a}_j & \mathbf{d}_j \end{bmatrix}^\intercal. \tag{3.3}$$

These vectors are flattened versions of the availability and demand across time and locations:

$$\mathbf{a}_j = \begin{pmatrix} A_{n_1,g_1,(j,h_1)} \\ A_{n_1,g_1,(j,h_2)} \\ \vdots \\ A_{n_{NG},g_G,(j,h_H)} \end{pmatrix} \in \mathbb{R}^{NG\times H} \quad \text{and} \quad \mathbf{d}_j = \begin{pmatrix} D_{n_1,(s,p,h_1)} \\ D_{n_1,(s,p,h_2)} \\ \vdots \\ D_{n_N,(s,p,h_H)} \end{pmatrix} \in \mathbb{R}^{N\times H}. \tag{3.4}$$

The representative periods are then used to replace the full set of original periods in the reduced model. We define a new set $\mathcal{R}$ that substitutes for $\mathcal{J}$ in the model formulation. For each $r \in \mathcal{R}$, we determine the associated demand and availability vectors, $\mathbf{a}_r$ and $\mathbf{d}_r$.

We also redefine the time-specific index set $\mathcal{I}$ as follows:

$$\mathcal{I} = \mathcal{R} \times \mathcal{H}. \tag{3.5}$$

This means that representative periods replace the full scenario-period product. The reduced model, using representative periods, is formulated below.

$$\min \quad c^{\text{inv}} + AF \cdot c^{\text{op}} \tag{3.6a}$$

$$\text{s.t.} \quad c^{\text{inv}} = \sum_{(n,g)\in\mathcal{NG}} I_{n,g} \cdot U_{n,g} \cdot i_{n,g} \tag{3.6b}$$

$$c^{\text{op}} = \sum_{\substack{i=(r,h)\\ \in\mathcal{R}\times\mathcal{H}}} \sum_{\substack{j=(s,p)\\ \in\mathcal{J}}} W_{r,j} \cdot \pi_s \left(\sum_{\substack{(n,g)\\ \in\mathcal{NG}}} V_{n,g} \cdot prod_{n,g,i} + \sum_{n\in\mathcal{N}} V^{\text{loss}} \cdot l_{n,i} \right) \tag{3.6c}$$

$$\sum_{g|(n,g)\in\mathcal{NG}} prod_{n,g,i} + \sum_{n'|(n',n)\in\mathcal{L}} f_{(n',n),i} - \sum_{n'|(n,n')\in\mathcal{L}} f_{(n,n'),i} + l_{n,i}$$

$$= D_{n,i} \qquad \forall n \in \mathcal{N}, i \in \mathcal{R}\times\mathcal{H} \tag{3.6d}$$

$$f_{e,i} \geq -L_e^{\text{imp}} \qquad \forall e \in \mathcal{L}, i \in \mathcal{R}\times\mathcal{H} \tag{3.6e}$$

$$f_{e,i} \leq L_e^{\text{exp}} \qquad \forall e \in \mathcal{L}, i \in \mathcal{R}\times\mathcal{H} \tag{3.6f}$$

$$prod_{n,g,i} \leq A_{n,g,i} \cdot i_{n,g} \cdot U_{n,g} \qquad \forall (n,g) \in \mathcal{NG}, i \in \mathcal{R}\times\mathcal{H} \tag{3.6g}$$

$$prod_{n,g,(r,h)} - prod_{n,g,(r,h-1)} \leq R_{n,g} \cdot U_{n,g} \cdot i_{n,g} \qquad \forall (n,g) \in \mathcal{NG}, r \in \mathcal{R}, h \in \mathcal{H}\backslash\{1\} \tag{3.6h}$$

$$prod_{n,g,(r,h)} - prod_{n,g,(r,h-1)} \geq -R_{n,g} \cdot U_{n,g} \cdot i_{n,g} \qquad \forall (n,g) \in \mathcal{NG}, r \in \mathcal{R}, h \in \mathcal{H}\backslash\{1\} \tag{3.6i}$$

The main change is in the operational costs. We do not separately calculate this per scenario, but have one total operational cost. This is a design choice so that the same model can be used for both per-scenario and cross-scenario approaches. The number $W_{r,j}$ symbolizes the weight contributed by period j to representative $\mathcal{R}$. This is multiplied by the probability of the scenario the period appears in. The total weight of each representative, where the scenario probability is taken into account, can be denoted as $W_r = \sum_{j=(s,p)\in\mathcal{J}} W_{r,j} \cdot \pi_s$. These weights are calculated based on how many original periods are represented by each r, incorporating both frequency and scenario probability. We explain the chosen approach for the weights in more detail in Section 3.3.4.

Per-scenario

An example of the per-scenario use of representative periods is provided by Micheli et al. [20] in the context of a generation and transmission expansion planning model. In this approach, representative periods are selected separately for each scenario. This involves applying the steps of normalization, selection, and calculation of weights to each scenario individually. This results in a separate representative set $\mathcal{R}_s$ for each scenario, representing all periods in that scenario, $\mathcal{J}_s$.

To calculate the weights in this case, we multiply the clustering weights by the scenario probability π_s for each $r \in \mathcal{R}_s$. One advantage of this approach is that the clustering task is performed on smaller, more homogeneous sets. However, this separation also means that similar patterns across different scenarios may be ignored. As a result, the union of all representatives, $\mathcal{R} = \bigcup_{s \in \mathcal{S}} \mathcal{R}_s$, may include redundant or highly similar periods that increase the model size without adding new information.

Cross-scenario

To better capture the overall structure of the full scenario space, we propose the cross-scenario method. Here, representative periods are selected from the combined set of periods across all scenarios, resulting in a single, scenario-independent set $\mathcal{R}$. This approach allows us to identify a smaller set of representative periods that generalizes well across different scenarios, reducing redundancy and potentially improving model compactness.

The key idea is to perform clustering on the full joint set $\mathcal{J} = \mathcal{S} \times \mathcal{P}$. The selection process thus considers the joint variability in demand and availability across all scenarios. This enables the model to reuse the same representative periods in multiple scenarios, rather than duplicating similar patterns. In this approach, the weights W_r reflect how many original periods across all scenarios are best represented by each r, taking into account the probability of those periods occurring based on the scenario probability. Although the calculation of weights becomes slightly more involved, it ensures that the reduced model still reflects the original probability structure. We further elaborate on this in Section 3.3.4.

An advantage of the cross-scenario method is its ability to minimize redundancy by recognizing similarities between periods in different scenarios. This can lead to a more efficient representation of uncertainty, especially when common temporal patterns exist across the scenario set. However, it also requires more computational effort during the selection phase.

3.3. Selecting representative periods

As we discussed in Chapter 2, there is a wide variety of approaches for selecting representative periods for the reduced model formulation. These approaches range from clustering-based methods to the earlier-mentioned hull methods or handpicking days with certain characteristics. Within these categories, further variations arise from choices in normalization, distance metrics, and weight strategies. The main aim of this thesis is to compare two classical clustering methods, k-means and k-medoids, with newer selection methods based on greedy algorithms that construct a convex or bounded conical hull, as implemented in the TulipaClustering.jl framework [23]. To ensure a fair comparison, we apply the same normalization and distance metrics across all methods. These steps are explained first, followed by a description of the selection methods. We conclude the section by discussing two approaches for the weights.

3.3.1. Normalization

Normalization is necessary for any selection method that uses a distance metric, as the availability of generation technologies ranges from 0 to 1, while demand values are in megawatts (MW) and can range up to high values, depending on the location and spatial scale. Without normalization, the scale difference would cause demand values to dominate the distance calculation.

As discussed in Chapter 2, two common normalization methods are z-normalization and normalization by the maximum value. We opt for the latter, as it requires less computation and is easier to interpret. Before clustering, we divide each demand value by a maximum value. However, demand levels vary significantly between locations. In our European case study, smaller countries may consume up to 50 times less electricity than larger ones. If we normalize by a global maximum across all locations, differences in smaller countries may have too little influence on the clustering process. While one could argue that larger countries are also responsible for more cross-border exports and thus deserve more weight, this risks neglecting important patterns at smaller locations. No country is entirely dependent on its neighbors. Therefore, to ensure that relative differences are treated equally across locations, we normalize by the maximum demand per location rather than using a global maximum. A similar approach, but then focused on the maximum demand per time step, was mentioned by Teichgraeber and Brandt [33].

In addition to scaling the demand, we also adapt the structure of the vectors used for clustering. For GEP models, we found that the ratio of availability to demand is more informative than the absolute availability itself. This is because increased availability alongside proportionally increased demand often results in similar system requirements. We are going into more detail on that later in Chapter 6. As such, we use the normalized demand in combination with the availability-to-demand ratio as the basis for clustering.

The modified clustering vectors then contain the following values per location:

$$\mathbf{a}_{j,n} = \begin{pmatrix} A_{n,g_1,(j,h_1)} \Big/ D_{n,(j,h_1)} \\ A_{n,g_1,(j,h_2)} \Big/ D_{n,(j,h_2)} \\ \vdots \\ A_{n,g_G,(j,h_H)} \Big/ D_{n,(j,h_H)} \end{pmatrix} \quad \text{and} \quad \mathbf{d}_{j,n} = \begin{pmatrix} D_{n,(s,p,h_1)} \Big/ D_n^{\texttt{max}} \\ D_{n,(s,p,h_2)} \Big/ D_n^{\texttt{max}} \\ \vdots \\ D_{n,(s,p,h_H)} \Big/ D_n^{\texttt{max}} \end{pmatrix}. \tag{3.7}$$

Here $D_n^{\texttt{max}}$ denotes the maximum demand at each location over the full dataset (typically taken over all scenarios and periods), used to normalize the demand. The availability is normalized by the demand at that same location and time step. We assume that if a location has a demand at some point in time higher than zero, it will have a demand higher than 0 at all times. If there are locations that only produce energy and do not have a demand, no normalization by demand is applied.

3.3.2. Distance metrics

Normalization is a necessary preprocessing step when we intend to compute distances between periods. It plays a key role in clustering-based approaches as well as in the hull-based algorithms we explore in this thesis. After normalization, the next step is to define how similarity or dissimilarity between periods is measured, which directly affects the outcome of the process.

There are several possible distance metrics. The most common one is the squared euclidean distance, which is widely used in clustering algorithms such as k-means and k-medoids. Alternative metrics like the manhattan distance have also been studied and, according to Mallapragada et al. [19], tend to yield comparable results in many energy system applications. For simplicity and consistency with prior work, we primarily use the squared euclidean distance in this thesis.

However, the TulipaClustering.jl package also includes an option for cosine distance, particularly for use with the greedy convex hull method [23]. While cosine distance is less commonly applied in energy system clustering, it can be especially useful if we want to find vectors that have a large angle between them, and are thus not proportional to each other. Due to its novelty in this context, we include cosine distance in our experiments to assess its performance.

Squared euclidean distance

The squared euclidean distance between two vectors $\mathbf{q}_i$ and $\mathbf{q}_j$ is defined as:

$$d_{\text{SE}}(i,j) = \sum_{k=1}^{D} \left(\mathbf{q}_{i,k} - \mathbf{q}_{j,k}\right)^2, \tag{3.8}$$

where D is the dimensionality of the clustering vectors. This metric penalizes large differences heavily in selection methods, thus tends to favor centroids that minimize the total variance within clusters. In other fields, it is shown that this metric is not always the most suitable option in higher dimensions and that the manhattan distance would be preferable [1]. However, it is still the most common choice for k-means and k-medoids.

Cosine distance

The cosine distance between two vectors measures the angle between them, ignoring differences in magnitude. It is defined as:

$$d_{\cos}(i,j) = 1 - \frac{\mathbf{q}_i \cdot \mathbf{q}_j}{|\mathbf{q}_i|, |\mathbf{q}_j|}. \tag{3.9}$$

Cosine distance ranges from 0 (vectors point in the same direction) to 2 (opposite direction). It emphasizes similarity in shape or profile rather than absolute scale, which could result in the selection of representative periods that are more distinct in those attributes rather than having similar patterns. This could lead to more variability in the set of representatives.

3.3.3. Selection methods

Now that the normalization and distance metric are defined, we turn to the selection of representative periods. Selection methods either choose periods directly from the original dataset (as in k-medoids or greedy hull methods), or generate artificial periods as centroids (as in k-means). In this thesis, we compare four methods: k-means, k-medoids, and two greedy algorithms that construct a bounded conical hull and a convex hull.

In what follows, we describe each method in more detail, including how representatives are selected in the algorithmic implementations that we use for each of the methods.

Greedy convex hull

We begin with the convex hull approach, which is not yet commonly used for selecting representative periods but was introduced in the TulipaClustering.jl package [23]. The convex hull of a set of points $S \subseteq \mathbb{R}^n$ is the minimal convex set containing S. This can formally be written as:

$$\text{conv}(S) = \left\{ \mathbf{x} \in \mathbb{R}^n \;\middle|\; \mathbf{x} = \sum_{i=1}^{k} \lambda_i \mathbf{s}_i,\ \mathbf{s}_i \in S,\ \lambda_i \geq 0,\ \sum_{i=1}^{k} \lambda_i = 1,\ k \in \mathbb{N} \right\}. \tag{3.10}$$

The objective is to identify a subset of representative periods, $\mathcal{R} \subseteq \mathcal{P}$, such that the set of representatives $\mathcal{R}$ are part of the corner points of the convex hull of the period set. This means that $\text{conv}(\mathcal{P}) = \text{conv}(\mathcal{R})$. All periods can be expressed as the convex combination of the representatives, guaranteeing that there is a feasible solution for all periods in the initial dataset when those representatives are used in the reduced model formulation. The theoretical justification for this feasibility is provided in Chapter 4.

However, computing the full convex hull in high-dimensional spaces is computationally expensive. The worst-case complexity of established algorithms, such as the method by Clarkson [4], is $O\left(N^{\lfloor \frac{n}{2} \rfloor}\right)$, where N is the number of input points and n the dimensionality. While algorithms such as Quickhull [2] often perform better in practice, they share the same exponential upper bound. Moreover, it could be possible that for a predefined number of representatives, we are sure that they will not form a convex hull over the initial set since the number of representatives is lower than the number of corner points of the hull.

To address this, Neustroev et al. [23] use a greedy approximation algorithm that incrementally builds a set of extreme points. Starting with an empty representative set, the algorithm first computes the mean of all periods. It then iteratively selects the point from the input set that is furthest from the convex hull of the current representatives, measured by the chosen distance metric (e.g., squared euclidean or cosine distance). This process continues until a desired number of representatives is reached.

Greedy bounded conical hull

The bounded conical hull method is closely related to the convex hull but covers a larger area. The idea is to find a conical hull, which is the minimal set of conical combinations of a set S. A conical combination is different from a convex combination in the sense that the weights do not have to sum up to exactly 1. For the bounded conical hull used, the weights are constrained to have a maximum sum of 1. This can formally be written as:

$$\text{conical}(S) = \left\{ \mathbf{x} \in \mathbb{R}^n \;\middle|\; \mathbf{x} = \sum_{i=1}^{k} \lambda_i \mathbf{s}_i,\ \mathbf{s}_i \in S,\ \lambda_i \geq 0,\ \sum_{i=1}^{k} \lambda_i \leq 1,\ k \in \mathbb{N} \right\}. \tag{3.11}$$

As with the convex hull method, a similar greedy algorithm is used to try to find representatives that form a bounded conical hull. To reform the algorithm from finding a convex hull to finding a bounded conical hull, the zero vector is added to the set of representatives. Therefore, this approach can only work when the distance to zero can be computed, which is not the case for the cosine distance metric. This means that this method will only be evaluated in combination with the squared euclidean distance.

***k*-means**

We follow with an explanation of k-means as previously given by Li et al. [18] in a similar context of representatives. With k-means, the goal is to find clusters such that the summed distance of each cluster member to the cluster mean (centroid) is minimized. Given a set S, the goal is to find a partition into k clusters, so that $S = \cup_{i=1}^{k} S_i$. The k-means clustering algorithm then solves the following optimization problem to find the optimal partition $\mathcal{S}^*$:

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \sum_{i=1}^{k} \sum_{s \in S_i} d(s, \mu_i), \tag{3.12}$$

where μ_i is the mean of the points in cluster S_i. Calculating this exactly is generally not feasible, so in practice iterative algorithms are used. In our case, we use the implementation provided in the Clustering.jl package in Julia. It starts with a random initialization of cluster centers, then partitions the input data based on these centers. The next iteration of cluster centers is calculated by taking the mean value of the points assigned to each cluster. Since random initialization is used, the outcome varies with each run. In this thesis, we always use multiple random seeds to account for this variability, and present the median and quartile values.

The k-means method is widely used due to its simplicity and efficiency, especially when using the squared euclidean distance. However, the resulting centroids may not correspond to actual periods from the dataset, and might have values that are overly averaged out. Li et al. [18] shows that if k-means is used, the reduced model solution will provide a lower bound. This can be an advantage when bounds are desired, but a consistent underestimation can also be seen as a drawback.

k-medoids

The k-medoids algorithm is similar to k-means but selects actual data points (medoids) as the cluster centers. The process is nearly the same, but each time that k-means computes the mean of a cluster, k-medoids finds the sample closest to this mean from the points in that cluster. A similar random initialization is done based on the input data, so also for k-medoids, we propose always using multiple seeds in our experiments.

3.3.4. Weight calculation

When the representative periods are found, we can calculate their weights to have an accurate representation in the objective function. The common approach is that each period gets assigned to exactly one representative, resulting in Dirac weights. We first explain this process. Afterwards, we also go more into detail in a new method of computing weights that is proposed by Neustroev et al. [23].

Dirac weights

While various methods for weight calculation exist, the most common is to assign each period to exactly one representative, typically the one to which it is closest. For k-means and k-medoids, this means that each representative receives a weight corresponding to the size of the cluster for which it serves as centroid or medoid. In the case of the greedy convex or bounded conical hull method, this assignment is less intuitive, since those approaches are not based on clusters. However, the same weights method of assigning each period to the closest representative can still be used.

In practice, this means that each entry of the weight matrix, $W_{r,j}$ is either equal to 0 or to 1. Also, for all j there is exactly one r such that $W_{r,j} = 1$. This weights method ensures that the annualization factor in the objective of the reduced model remains consistent with that of the full model. While this way of assigning weights aligns naturally with clustering-based methods like k-means and k-medoids, its interpretation is less straightforward for the hull-based approaches. For this reason, TulipaClustering.jl also implements an alternative weighting scheme, which we describe in the next part.

Blended representative periods

The core idea behind the hull methods is to identify representatives that form the corner points of the convex or bounded conical hull of the original set of periods. If this is achieved, each original period can be expressed as a convex or conical combination of the representatives, as defined in (3.10) and (3.11). The concept of blended weights builds on this observation.

If a period lies within the convex or bounded conical hull formed by the selected representatives, then we can compute the associated weights, denoted as the λ coefficients in the earlier definitions of the hulls. These express the period as a convex or conical combination of the representatives. If a period does not lie within the hull, we approximate it by projecting it onto the hull and use the associated weights from the projection. This approach assigns each period to multiple representatives, with different weights, rather than to just one. These weights are then summed across all periods to determine the final weight for each representative, resulting in what is called the blended weights method in the TulipaClustering.jl [23]. This approach is applicable to both convex and conical combinations.

In the implementation, the implementation begins with a projection onto the unit simplex, using the algorithm proposed by Condat [5], which finds the point within the unit simplex closest to the original period vector. This step ensures that the projected weights are non-negative and sum to one. Subsequently, a subgradient descent algorithm is used to iteratively minimize the error between the reconstructed period (using the weighted combination of representatives) and the original period. Again, for a bounded conical combination a similar approach is used, but including the zero vector and removing its weight later on.

In the weight matrix, the entries are no longer restricted to be either 0 or 1, but can take any value in between. For the convex approach, the total weight distributed over all representatives for one period j should sum exactly to 1. For conical combinations, individual periods may contribute total weights of less than one (since we are using a bounded conical hull). In such cases, a rescaling step is required to ensure that the total sum of all entries of the weight matrix matches the total number of original periods in the full model.

3.4. Evaluation techniques

Finally, we also discuss the evaluation techniques used in this thesis. While some elements are theoretical, the focus lies also on conducting experiments to highlight the advantages and disadvantages of each method. To evaluate these methods properly, we must first clarify the purpose of the reduction.

The reduced model yields optimal decision variables for a smaller, representative set of periods. While the operational decision variables are not always directly meaningful, since the original periods are not exactly the same as the representatives, the investment decisions are immediately relevant, as they are carried over into the full model. Some papers evaluate the quality of the reduced model by comparing its objective value to that of the full model, using the reduced model's investment decisions as input [18]. Others use the installed capacity as a performance metric [26]. In this thesis, we follow a third approach: we fix the investment decisions obtained from the reduced model, and then run the full model using these fixed decisions, similar to the method in Li et al. [18]. We compare this result to a solution obtained without any reduction at all.

There are several ways to compare these solutions. One metric we use is to assess feasibility: by counting the number of time steps with loss of load under the full model when using reduced-model investment decisions. This gives an impression of the robustness of the model. We also evaluate the total cost, and from that compute the relative regret, defined as the percentage increase in the objective function due to using a reduced model. This gives an insight in how close to the optimal solution the reduction is.

In the context of stochastic programming, it is especially important to define what we consider the original model. As mentioned earlier, one could compare the stochastic model to a deterministic model using average input values, which gives rise to the value of the stochastic solution (VSS) [29], a common argument for using stochastic optimization. However, since the benefit of stochastic programming is well established in prior work, our focus is on comparing reduction methods within the stochastic framework. The challenge, especially in the large-scale case study presented at the end of this thesis, is that solving the full stochastic model is computationally infeasible. In such cases, we compare the reduced models against one another based on their objective values, but we cannot benchmark them against the full optimal solution.

Finally, since we are working with scenarios, it is important to decide on the evaluation perspective: do we want a method that performs best on average, or one that performs best in the worst-case scenario. Different methods may excel under different criteria, but we assess the mean value in this thesis. Additionally, evaluating performance on the same scenarios used for representative selection can introduce bias. It is therefore preferable to test on scenarios unseen by the model, a practice known as out-of-sample evaluation [29].

4

Feasibility in the hull

The ultimate goal of solving the reduced model is to construct a feasible solution for the original complete model, using the optimal solution of the reduced model. In this chapter, we prove that whenever an exact version of the bounded conical hull or convex hull method is applied to identify representative periods, we can always construct such a feasible solution for the initial model from the reduced model. We focus on the stochastic programming formulation for generation expansion planning (GEP), but the results can be extended to more general formulations. Specifically for GEP, we demonstrate that if the optimal solution of the reduced model has no loss of load, then the provided solution for the original model will also satisfy demand at all times. We distinguish between GEP models without (Section 4.1) and with (Section 4.2) inter-period ramping constraints. For the latter, we need additional constraints in the reduced model to ensure that we can construct a feasible solution for the initial model. Using the bounded conical or convex hull method combined with the previously mentioned blended weights, we can also prove that the optimal solution of the reduced model provides an upper bound on the costs for the original model. We do so in Section 4.3.

For the proofs, we rely on the critical assumption that representative periods are identified using the bounded conical hull or convex hull method. Consequently, these results do not apply to other clustering methods, such as k-means or k-medoids. Moreover, we assume that the hull perfectly represents the periods, which may not always hold for approximate algorithms. This highlights not only the advantages of our method over existing clustering techniques, but also the need to further refine approaches to approximate the bounded conical hull or convex hull.

4.1. Feasible solution for generation expansion planning without inter-period constraints

In this section, we present the theorem on feasibility for the GEP model without inter-period ramping constraints. We start by clarifying the definition and usage of representative periods, as previously discussed in Section 3.2. We explain when representative periods perfectly represent the original set of periods in Subsection 4.1.1, and then we introduce the theorem without inter-period constraints in Subsection 4.1.2.

4.1.1. Mathematical notation

As defined earlier in (3.2), we call $\mathcal{C}$ the set of vectors for which we want to find representatives. In the case of the GEP model, each period j has an associated vector $\mathbf{q}_j$, containing the values for availability and demand during that period, as given in (3.3) and (3.4). Our goal is to find a set of representative periods $\mathcal{R}$ such that their vectors perfectly represent the corresponding values in $\mathcal{C}$. Since this definition is central to the proofs in this chapter, we formalize it as follows:

Definition 4.1. We say that the set $\mathcal{R}$ represents the set $\mathcal{J}$ perfectly if every element $\mathbf{q}_j$ in $\mathcal{C}$ can be expressed as the sum of weighted elements of the set of representative values indexed by $\mathcal{R}$. We denote this set of representative values by $\mathcal{C}_\mathcal{R}$:

$$\mathcal{C}_\mathcal{R} = \{\mathbf{q}_r \mid \mathbf{q} \in \mathbb{R}^{G\times H+N\times H}, r \in \mathcal{R}\}. \tag{4.1}$$

Thus, $\mathcal{R}$ represents the set $\mathcal{J}$ perfectly if for all $j \in \mathcal{J}$ there exist scalars $\lambda_{r,j} \in \mathbb{R}$ such that the following holds:

$$\mathbf{q}_j = \sum_{r \in \mathcal{R}} \lambda_{r,j} \cdot \mathbf{q}_r. \tag{4.2}$$

This completes the definition.

The representatives are indexed by the set $\mathcal{R}$, which can either be a subset of $\mathcal{J}$ when actual data points are selected as representatives, or a set of new points when artificial representatives are created. This distinction does not affect the proof, however, adding artificial representatives can affect the costs and thus the optimality by making the model too conservative. Whether it might be beneficial to add some artificial representatives will be explained later on when discussing them in 6.

Next, we present four conditions for the weights $\lambda_{r,j}$. Conditions 1 and 2 define a bounded conical hull and are essential for all theorems in this chapter. However, if the representatives form a convex hull, as described in Condition 3, we may be able to obtain more precise results. The appearance of Condition 4 is highly unlikely in practice, but it is included to provide additional theoretical insight.

Condition 1. *The weights are non-negative, i.e., $\lambda_{r,j} \in \mathbb{R}_{\geq 0}$ for all $r \in \mathcal{R}$ and $j \in \mathcal{J}$. This implies that $\mathcal{C} \subseteq coni(\mathcal{C}_{\mathcal{R}})$, the conical hull of $\mathcal{C}_{\mathcal{R}}$.*

Condition 2. *The sum of the weights for each period j is bounded above by 1, i.e., $\sum_{r \in \mathcal{R}} \lambda_{r,j} \leq 1$ for all $j \in J$.*

Condition 3. *The weights sum to 1, i.e., $\sum_{r \in \mathcal{R}} \lambda_{r,j} = 1$ for all $j \in J$. Together with Condition 1, this implies that $\mathcal{C} \subseteq conv(\mathcal{C}_{\mathcal{R}})$, the convex hull of $\mathcal{C}_{\mathcal{R}}$.*

Condition 4. *Every period is perfectly represented by exactly one of the representative periods, i.e., $\lambda_{r,j} \in \{0,1\}$ for all $r \in \mathcal{R}$ and $j \in \mathcal{J}$ and $\sum_{r \in \mathcal{R}} \lambda_{r,j} = 1$ for all $j \in \mathcal{J}$. This implies that $\mathcal{C} = \mathcal{C}_{\mathcal{R}}$.*

Lastly, we introduce a more compact notation for the maximum sum of the weights, as it will be used in the constraints of the reduced model:

$$\Lambda = \max_{j \in \mathcal{J}} \sum_{r \in \mathcal{R}} \lambda_{r,j} \tag{4.3}$$

Under Condition 3 or 4, this is equal to 1. In case of Condition 2, it can be smaller than 1, provided that the representatives selected are artificial representatives. However, when actual data points are chosen as representatives, those points can be exactly represented by one of their representatives, resulting again in $\Lambda = 1$.

Remark 1. *Before presenting the theorem, we note that in practice, clustering is performed not on the original vectors given in* (3.3) *and* (3.4)*, but on the normalized vectors defined in* (3.7)*. However, when the weights satisfy* (4.2) *for the normalized vectors, they also satisfy it for the denormalized vectors for both the representative periods and the original periods.*

4.1.2. Constructing a feasible solution

With the conditions stated and all notation defined, we now focus on the first theorem, in which we construct a feasible solution to the original model using the optimal solution of the reduced model. In this theorem, we ignore inter-period constraints. Since representative periods are not actual consecutive time steps, modeling inter-period constraints becomes challenging. As a result, a common assumption in the literature is to omit these constraints entirely [18, 34]. However, especially in settings that include long-term storage, including them could be beneficial for model outcomes [35]. For this reason, we later examine formulations that do include inter-period constraints.

In the cross-scenario method, we identify a set of representative periods $\mathcal{R}$ that represent the full set of initial periods across all scenarios, i.e., $\mathcal{J} = \mathcal{S} \times \mathcal{P}$. In the per-scenario method, we instead identify a separate set of representative periods $\mathcal{R}_s$ for each scenario, corresponding to $\mathcal{J}_s = (s,p) \mid p \in \mathcal{P}$. The complete set of representative periods is then $\mathcal{R} = \bigcup_s \mathcal{R}_s$, and $\mathcal{J} = \bigcup_s \mathcal{J}_s$. Based on these sets, we define the reduced model as follows:

$$\min \quad c^{\text{inv}} + AF \cdot c^{\text{op}} \tag{4.4a}$$

$$\text{s.t.} \quad c^{\text{inv}} = \sum_{(n,g)\in\mathcal{NG}} I_{n,g}\cdot U_{n,g}\cdot i_{n,g} \tag{4.4b}$$

$$c^{\text{op}} = \sum_{\substack{i=(r,h)\\ \in\mathcal{R}\times\mathcal{H}}} W_r \left(\sum_{\substack{(n,g)\\ \in\mathcal{NG}}} V_{n,g}\cdot prod_{n,g,i} + \sum_{n\in\mathcal{N}} V^{\text{loss}}\cdot l_{n,i} \right) \tag{4.4c}$$

$$\sum_{g|(n,g)\in\mathcal{NG}} prod_{n,g,i} + \sum_{n'|(n',n)\in\mathcal{L}} f_{(n',n),i} - \sum_{n'|(n,n')\in\mathcal{L}} f_{(n,n'),i} + l_{n,i}$$

$$= D_{n,i} \qquad \forall n\in\mathcal{N}, i\in\mathcal{R}\times\mathcal{H} \tag{4.4d}$$

$$f_{e,i} \geq -L_e^{\text{imp}}\cdot\frac{1}{\Lambda} \qquad \forall e\in\mathcal{L}, i\in\mathcal{R}\times\mathcal{H} \tag{4.4e}$$

$$f_{e,i} \leq L_e^{\text{exp}}\cdot\frac{1}{\Lambda} \qquad \forall e\in\mathcal{L}, i\in\mathcal{R}\times\mathcal{H} \tag{4.4f}$$

$$prod_{n,g,i} \leq A_{n,g,i}\cdot i_{n,g}\cdot U_{n,g} \qquad \forall (n,g)\in\mathcal{NG}, i\in\mathcal{R}\times\mathcal{H} \tag{4.4g}$$

$$prod_{n,g,(r,h)} - prod_{n,g,(r,h-1)} \leq R_{n,g}\cdot U_{n,g}\cdot i_{n,g}\cdot\frac{1}{\Lambda} \qquad \forall (n,g)\in\mathcal{NG}, r\in\mathcal{R}, h\in\mathcal{H}\backslash\{1\} \tag{4.4h}$$

$$prod_{n,g,(r,h)} - prod_{n,g,(r,h-1)} \geq -R_{n,g}\cdot U_{n,g}\cdot i_{n,g}\cdot\frac{1}{\Lambda} \qquad \forall (n,g)\in\mathcal{NG}, r\in\mathcal{R}, h\in\mathcal{H}\backslash\{1\} \tag{4.4i}$$

We chose to use the summed weights W_r in the objective function, which was earlier defined in Section 3.2 and is a product of the weight matrix and the scenario probabilities.

As mentioned earlier, feasibility in an energy model unofficially means that the demand equation (4.4d) can be met without loss of load. This is why loss of load is heavily penalized in the model's operational costs. However, because of this internal penalization, GEP can always obtain a feasible solution by adding sufficient loss of load. In the following theorem, we thus analyze the amount of the loss of load as well: Assuming that the optimal solution to the reduced model has no loss of load, the constructed solution for the original model will also contain no loss of load. If this is not possible, the theorem ensures that a solution exists in which the loss of load costs are at least bounded.

Theorem 4.1. *Consider an instance of the GEP model as described in Section 3.1, but without inter-period ramping constraints. We find representative periods $\mathcal{R}$ that perfectly represent the original period set $\mathcal{J}$. Solve the reduced model with this representative set to optimality, as formulated in* (4.4)*. An optimal solution to the reduced model always exists. Denote the optimal values of the decision variables as $i^\star_{n,g}$, $prod^\star_{n,g,(r,h)}$, $f^\star_{e,(r,h)}$, and $l^\star_{n,(r,h)}$.*

If the weights $\lambda_{r,j}$ in (4.2) *satisfy Condition 1 and 2 for every $j\in\mathcal{J}$, then the variables $prod_{n,g,(j,h)}$, $f_{e,(j,h)}$, and $l_{n,(j,h)}$, as defined in* (4.5) *through* (4.7)*, together with the investment decisions $i^\star_{n,g}$, will form a feasible solution to the original model:*

$$prod_{n,g,(j,h)} = \sum_{r\in\mathcal{R}} \lambda_{r,j}\cdot prod^\star_{n,g,(r,h)} \qquad \forall (n,g)\in\mathcal{NG}, h\in\mathcal{H}, j\in\mathcal{J}, \tag{4.5}$$

$$f_{e,(j,h)} = \sum_{r\in\mathcal{R}} \lambda_{r,j}\cdot f^\star_{e,(r,h)} \qquad \forall e\in\mathcal{L}, h\in\mathcal{H}, j\in\mathcal{J}, \tag{4.6}$$

$$l_{n,(j,h)} = \sum_{r\in\mathcal{R}} \lambda_{r,j}\cdot l^\star_{n,(r,h)} \qquad \forall n\in\mathcal{N}, h\in\mathcal{H}, j\in\mathcal{J}. \tag{4.7}$$

Moreover, if there is no loss of load, e.g. if $l^\star_{n,(r,h)} = 0$ for all $n\in\mathcal{N}, h\in\mathcal{H}, j\in\mathcal{J}$, then the proposed solution will also not contain any loss of load.

If the optimal solution of the reduced model does contain loss of load, then in the proposed solution, the loss of load at each hour and location is bounded above by the maximum loss of load observed in the reduced model for that hour and location.

Proof. We aim to show that all constraints of the original model are satisfied by the proposed solution. The constraints are given in (3.1d) through (3.1j), and we address each set of constraints individually. Afterwards, we prove the statements about the loss of load.

Balance constraint (3.1d) Let $j \in \mathcal{J}$, $n \in \mathcal{N}$, and $h \in \mathcal{H}$. We first rewrite the balance constraint so that we only have the production on the left-hand side of the equality:

$$\sum_{g|(n,g)\in\mathcal{NG}} prod_{n,g,(j,h)} = D_{n,(j,h)} - \sum_{n'|(n',n)\in\mathcal{N}} f_{(n',n),(j,h)} + \sum_{n'|(n,n')\in\mathcal{L}} f_{(n,n'),(j,h)} - l_{n,(j,h)}. \tag{4.8}$$

We need to show that the balance constraint of the original model will hold under the proposed solution $\forall n \in \mathcal{N}, j \in \mathcal{J}, h \in \mathcal{H}$. Using the definitions in (4.5), we can rewrite the production term on the left-hand side of (4.8). Interchanging the sums and using that the balance constraint is satisfied for the representative periods in the reduced model, leads to the following set of equations:

$$\begin{aligned}
\sum_{g|(n,g)\in\mathcal{NG}} prod_{n,g,(j,h)} &= \sum_{g|(n,g)\in\mathcal{NG}} \left(\sum_{r\in\mathcal{R}} \lambda_{r,j} \cdot prod^{\star}_{n,g,(r,h)} \right) \\
&= \sum_{r\in\mathcal{R}} \lambda_{r,j} \cdot \left(\sum_{g|(n,g)\in\mathcal{NG}} prod^{\star}_{n,g,(r,h)} \right) \\
&= \sum_{r\in\mathcal{R}} \lambda_{r,j} \cdot \left(D_{n,(r,h)} - \sum_{n'|(n',n)\in\mathcal{N}} f^{\star}_{(n',n),(r,h)} + \sum_{n'|(n,n')\in\mathcal{L}} f^{\star}_{(n,n'),(r,h)} - l^{\star}_{n,(r,h)} \right).
\end{aligned} \tag{4.9}$$

We look at each component in (4.9) individually whether we can rewrite it. First of all the demand term, $\sum_{r\in\mathcal{R}} \lambda_{r,j} \cdot D_{n,(r,h)}$, equals $D_{n,(j,h)}$ because the weights $\lambda_{r,j}$ satisfy (4.2). Then for the flow and loss of load parts, we use the definitions in (4.6) and (4.7) to make a similar replacement. For the first flow component this looks as follows:

$$\begin{aligned}
\sum_{r\in\mathcal{R}} \lambda_{r,j} \sum_{n'|(n',n)\in\mathcal{N}} f^{\star}_{(n',n),(r,h)} &= \sum_{n'|(n',n)\in\mathcal{N}} \sum_{r\in\mathcal{R}} \lambda_{r,j} f^{\star}_{(n',n),(r,h)} \\
&= \sum_{n'|(n',n)\in\mathcal{N}} f_{(n',n),(j,h)}.
\end{aligned} \tag{4.10}$$

By repeating this process for the other terms in (4.9), we get the following equation, which is equal to the right-hand side of the balance equation in (4.8):

$$D_{n,(j,h)} - \sum_{n'|(n',n)\in\mathcal{N}} f_{(n',n),(j,h)} + \sum_{n'|(n,n')\in\mathcal{L}} f_{(n,n'),(j,h)} - l_{n,(j,h)}. \tag{4.11}$$

We can thus conclude that the constraint is met by the solution.

Transmission constraints (3.1e) - (3.1f) We will show that the import capacity constraint in (3.1e) is met for the original model using the proposed solution. Let $e \in \mathcal{L}$, $j \in \mathcal{J}$, and $h \in \mathcal{H}$. The flow $f^{\star}_{e,(r,h)}$ satisfies (3.1e) for each representative. Using the definition in (4.6) and the non-negativity of the weights in Condition 1, we can rewrite the flow:

$$\begin{aligned}
f_{e,(j,h)} &= \sum_{r\in\mathcal{R}} \lambda_{r,j} \cdot f^{\star}_{e,(r,h)} \\
&\geq \sum_{r\in\mathcal{R}} \lambda_{r,j} \cdot (-L^{\text{imp}}_{e} \cdot \frac{1}{\Lambda}) \\
&\geq -L^{\text{imp}}_{e}
\end{aligned} \tag{4.12}$$

The last inequality follows from Condition 2 and the definition of Λ as given in (4.3). We can conclude that the import constraint is satisfied. The argument for the export capacity constraint in (3.1f) is similar.

Capacity constraint (3.1g) Let $(n,g) \in \mathcal{NG}$, $j \in \mathcal{J}$ and $h \in \mathcal{H}$. We follow a similar argument as above to rewrite the left-hand side of the original constraint. We can then get an inequality, using the definition in (4.5) and Condition 1. The last equality in the equations below is derived from $\lambda_{r,j} \cdot A_{n,g,(r,h)} = A_{n,g,(j,h)}$, by the definition of the weights in (4.2).

$$\begin{aligned} prod_{n,g,(j,h)} &= \sum_{r\in\mathcal{R}} \lambda_{r,j} \cdot prod^{\star}_{n,g,(r,h)} \\ &\leq \sum_{r\in\mathcal{R}} \lambda_{r,j} \cdot A_{n,g,(r,h)} \cdot i^{\star}_{n,g} \cdot U_{n,g} \\ &= i^{\star}_{n,g} \cdot U_{n,g} \cdot A_{n,g,(j,h)}. \end{aligned} \tag{4.13}$$

We can conclude that the capacity constraint holds for the proposed solution for the original model.

Ramping constraint (3.1h) - (3.1i) Lastly, we need to check the ramping constraints. In this model, we are only considering intra-period ramping constraints. Below is shown how the ramping up constraint in (3.1h) is satisfied in the original model under the proposed solution. The argument for the ramping-down constraint follows similarly. Let $j \in \mathcal{J}$, $h \in \mathcal{H}\backslash\{1\}$, and $(n,g) \in \mathcal{NG}$. We follow a similar approach as before with the import capacity constraint:

$$\begin{aligned} prod_{n,g,(j,h)} - prod_{n,g,(j,h-1)} &= \sum_{r\in\mathcal{R}} \lambda_{r,j} \cdot prod^{\star}_{n,g,(r,h)} - \sum_{r\in\mathcal{R}} \lambda_{r,j} \cdot prod^{\star}_{n,g,(r,h-1)} \\ &= \sum_{r\in\mathcal{R}} \lambda_{r,j} \cdot \left(prod^{\star}_{n,g,(r,h)} - prod^{\star}_{n,g,(r,h-1)} \right) \\ &\leq \sum_{r\in\mathcal{R}} \lambda_{r,j} \cdot \left(R_{n,g} \cdot U_{n,g} \cdot i^{\star}_{n,g} \cdot \frac{1}{\Lambda} \right) \\ &\leq R_{n,g} \cdot U_{n,g} \cdot i^{\star}_{n,g}. \end{aligned} \tag{4.14}$$

We can conclude that the intra-period ramping-up constraint is met, and with a similar argument we can show the same for the ramping-down constraint.

By verifying the balance, transmission, capacity and ramping constraints, we have demonstrated that the proposed solution satisfies all constraints for the full input set indexed by $\mathcal{J}$, under the assumption that the representative set $\mathcal{C}_{\mathcal{R}}$ satisfies Condition 1 and Condition 2.

Loss of load To show that if there is no loss of load in the reduced model solution, then the proposed solution will also be free of loss of load, we use the defined values in (4.7). If the reduced model has no loss of load, this sum will always be zero, meaning that the proposed solution is indeed free of loss of load.

If there is loss of load in the reduced model, then for each location and hour, define the maximum loss of load in the reduced model as follows:

$$l^{\max}_{n,h} = \max_{r\in\mathcal{R}} l^{\star}_{n,(r,h)} \tag{4.15}$$

We then bound the loss of load in the original model at location n and hour h, for each $j \in \mathcal{J}$:

$$l_{n,(j,h)} = \sum_{r\in\mathcal{R}} \lambda_{r,j} \cdot l^{\star}_{n,(r,h)} \tag{4.16}$$

$$\leq \sum_{r\in\mathcal{R}} \lambda_{r,j} \cdot l^{\max}_{n,h} \tag{4.17}$$

$$\leq l^{\max}_{n,h} \tag{4.18}$$

The last step follows from the non-negativity of the weights and Condition 2, which ensures that the weighted sum does not exceed the maximum value. We have thus shown that the loss of load in the proposed solution is bounded at each time step and hour by the maximum loss of load in the reduced model for that location and hour. This completes the proof of the theorem.

□

A nice property of the proof is that it applies to any set $\mathcal{J}$ for which $\mathcal{R}$ is a perfectly representing set satisfying Condition 1 and 2. So for all periods with arbitrary values in the bounded conical hull of $\mathcal{C}_{\mathcal{R}}$ a feasible solution can be found with this theorem.

Also note that the proof does not rely on Condition 3. Therefore, using either the bounded conical hull or a convex hull to represent the values does not affect the existence of the proposed solution in the theorem. However, when

Condition 3 holds, the final step for the ramping and transmission constraints can be reformulated as an equality. This might have beneficial effects on optimality, and therefore we will continue to experiment with both the bounded conical hull and convex hull in later case studies.

Since this thesis focuses on Generation Expansion Planning (GEP) in particular, the theorem and proof were presented in the context of GEP. However, the results are extendable to more general formulations. To support this, we provide a version of the proof in a general setting in Appendix A, and encourage readers to apply the approach more broadly.

4.2. Including inter-period ramping constraints

In the previous section, we assumed that inter-period constraints could be ignored in the initial model. This simplification is common in the literature and will also be applied in subsequent chapters of this thesis. However, neglecting these constraints can lead to an unrealistic system, especially for models involving long-term storage or technologies with limited flexibility. To address this, we now present two formulations that incorporate a specific type of inter-period constraint: ramping constraints. These are intended as initial steps toward more comprehensive future work in this area.

In the first formulation of the reduced model, we include all inter-period ramping constraints from the original model directly. The feasibility of the resulting solution now depends on preserving these constraints in the reduced formulation. To achieve this without introducing additional decision variables, we reformulate the original ramping constraints (3.1j) and (3.1k) using the representative periods. Since the original production variables in the theorem's solution are defined as weighted sums of production values from the reduced model, we substitute $prod_{n,g,(j,h)}$ in the original ramping with their equivalent expressions in (4.5). This leads to the following two constraints, which exactly replicate the original inter-period ramping behavior:

$$\sum_{r\in\mathcal{R}} \lambda_{r,(s,p-1)} \cdot prod_{n,g,(r,H)} - \sum_{r\in\mathcal{R}} \lambda_{r,(s,p)} \cdot prod_{n,g,(r,1)} \leq R_{n,g} \cdot U_{n,g} \cdot i_{n,g} \qquad \forall (n,g) \in \mathcal{NG}, s \in \mathcal{S}, p \in \mathcal{P}\backslash\{1\}, \tag{4.19}$$

$$\sum_{r\in\mathcal{R}} \lambda_{r,(s,p-1)} \cdot prod_{n,g,(r,H)} - \sum_{r\in\mathcal{R}} \lambda_{r,(s,p)} \cdot prod_{n,g,(r,1)} \geq -R_{n,g} \cdot U_{n,g} \cdot i_{n,g} \qquad \forall (n,g) \in \mathcal{NG}, s \in \mathcal{S}, p \in \mathcal{P}\backslash\{1\}. \tag{4.20}$$

The theorem below follows directly from these constraints.

Theorem 4.2. *Consider an instance of the GEP model as described in Section 3.1. We find representative periods $\mathcal{R}$ that perfectly represent the original period set $\mathcal{J}$. Solve the reduced model, as formulated in* (4.4) *with the additional ramping constraints in* (4.20) *and* (4.19) *to optimality. An optimal solution to the reduced model always exists. Denote the optimal values of the decision variables as $i^\star_{n,g}$, $prod^\star_{n,g,(r,h)}$, $f^\star_{e,(r,h)}$, and $l^\star_{n,(r,h)}$.*

If the weights $\lambda_{r,j}$ in (4.2) *satisfy Conditions 1 and 2 for every $j \in \mathcal{J}$, then the variables $prod_{n,g,(j,h)}$, $f_{e,(j,h)}$, and $l_{n,(j,h)}$, as defined in* (4.5) *through* (4.7)*, together with the investment decisions $i^\star_{n,g}$, will form a feasible solution to the original model.*

Moreover, if there is no loss of load, e.g. if $l^\star_{n,(r,h)} = 0$ for all $n \in \mathcal{N}, h \in \mathcal{H}, j \in \mathcal{J}$, then the proposed solution will also not contain any loss of load.

If the optimal solution of the reduced model does contain loss of load, then in the proposed solution, the loss of load at each hour and location is bounded above by the maximum loss of load observed in the reduced model for that hour and location.

Proof. To show that all constraints of the original model are satisfied, we can reuse the proof of Theorem 4.1. This proof already covers all constraints except the inter-period ramping constraints. We only need to show that these inter-period ramping constraints hold for the original model under the proposed solution.

The reduced model contains the constraints defined in (4.19) and (4.20). These constraints are, in fact, identical to the inter-period ramping constraints in the original model, given in (3.1j) and (3.1k). To show this, we need to substitute $prod_{n,g,(s,p-1,H)}$ with its equivalent expression:

$$\sum_{r\in\mathcal{R}} \lambda_{r,(s,p-1)} \cdot prod_{n,g,(r,H)}. \tag{4.21}$$

Applying the same substitution to the other production variable gives the equations in the reduced model formulation. Since they are already enforced in the solution of the reduced model, it follows that the inter-period ramping constraints are automatically satisfied in the solution in the theorem. We can thus conclude that the proposed solution satisfies all constraints of the original model. This completes the proof.

□

This proof relies on incorporating period-specific ramping constraints into the reduced model. As a result, it cannot be extended to an arbitrary set $\mathcal{J}$ where $\mathcal{C}_{\mathcal{R}}$ satisfies Conditions 1 and 2, while this was possible without inter-period constraints. This will likely increase the costs when the actual realizations do not perfectly match the input data. Moreover, the reduced model in this theorem will take more time to solve to optimality, as significantly more constraints are added.

These two downsides of the proposed formulation in the theorem motivate us to explore alternative methods for incorporating inter-period ramping constraints. In Corollary 4.3, we introduce an approach that will need fewer constraints. While this method will likely eliminate the optimal and near-optimal solutions of the original problem, it can provide a useful starting point for developing more efficient strategies for handling inter-period (ramping) constraints.

For the corollary, we add a set of constraints to the reduced model formulation to replace the constraints in (4.19) and (4.20). The replacement constraints make sure that the first hour of each representative period is within the ramping limits of the last hour of all representative periods. This can be summarized in the following two constraints:

$$prod_{n,g,(r_1,H)} - prod_{n,g,(r_2,1)} \leq R_{n,g} \cdot U_{n,g} \cdot i_{n,g} \qquad \forall (n,g) \in \mathcal{NG}, r_1, r_2 \in \mathcal{R}, \tag{4.22}$$

$$prod_{n,g,(r_1,H)} - prod_{n,g,(r_2,1)} \geq -R_{n,g} \cdot U_{n,g} \cdot i_{n,g} \qquad \forall (n,g) \in \mathcal{NG}, r_1, r_2 \in \mathcal{R}. \tag{4.23}$$

We do need the stronger assumption that the weights satisfy Condition 3. With a convex hull, the maximum production difference between representative periods provides an upper bound for the differences among the actual periods they represent. This bound does not hold for the bounded conical hull.

Corollary 4.3. *Consider an instance of the GEP model as described in Section 3.1. We find representative periods $\mathcal{R}$ that perfectly represent the original period set $\mathcal{J}$. Solve the reduced model, as formulated in* (4.4) *with the additional ramping constraints in* (4.22) *and* (4.23) *to optimality. An optimal solution to the reduced model always exists. Denote the optimal values of the decision variables as $i^{\star}_{n,g}$, $prod^{\star}_{n,g,(r,h)}$, $f^{\star}_{e,(r,h)}$, and $l^{\star}_{n,(r,h)}$.*

If the weights $\lambda_{r,j}$ in (4.2) *satisfy Conditions 1, 2 and 3 for every $j \in \mathcal{J}$, then the variables $prod_{n,g,(j,h)}$, $f_{e,(j,h)}$, and $l_{n,(j,h)}$, as defined in* (4.5) *through* (4.7)*, together with the investment decisions $i^{\star}_{n,g}$, will form a feasible solution to the original model.*

Moreover, if there is no loss of load, e.g. if $l^{\star}_{n,(r,h)} = 0$ for all $n \in \mathcal{N}, h \in \mathcal{H}, j \in \mathcal{J}$, then the proposed solution will also not contain any loss of load.

If the optimal solution of the reduced model does contain loss of load, then in the proposed solution, the loss of load at each hour and location is bounded above by the maximum loss of load observed in the reduced model for that hour and location.

Proof. Since this corollary replaces Theorem 4.2, we only need to verify that the inter-period ramping constraints remain satisfied under these replacements.

Let $(n,g) \in \mathcal{NG}$. For the pair (n,g), we define the maximum production value across all representative periods at the end of a period, and the minimum production value across all representative periods at the beginning of a period, as follows:

$$prod^{\max}_{n,g} = \max_{r \in \mathcal{R}} prod_{n,g,(r,H)}, \tag{4.24}$$

$$prod^{\min}_{n,g} = \min_{r \in \mathcal{R}} prod_{n,g,(r,1)}. \tag{4.25}$$

Let $s \in \mathcal{S}$ and $p \in \mathcal{P} \backslash \{1\}$. Define $\lambda_{r,j}$ as the weights satisfying (4.2) for $j = (p,s)$. Then, the following holds:

$$\begin{aligned}\sum_{r\in\mathcal{R}} \lambda_{r,(p-1)} \cdot prod_{n,g,(r,H)} - \sum_{r\in\mathcal{R}} \lambda_{r,p} \cdot prod_{n,g,(r,H)} &\leq \sum_{r\in\mathcal{R}} \lambda_{r,(p-1)} \cdot prod^{\max}_{n,g} - \sum_{r\in\mathcal{R}} \lambda_{r,p} \cdot prod^{\min}_{n,g} \\ &= prod^{\max}_{n,g} - prod^{\min}_{n,g} \\ &\leq R_{n,g} \cdot U_{n,g} \cdot i^{\star}_{n,g}.\end{aligned} \tag{4.26}$$

The weights can be removed in the second step since $prod^{\max}_{n,g}$ and $prod^{\min}_{n,g}$ are constant for a fixed combination of location and generation technology, and the weights satisfy Condition 3.

By confirming that the inter-period ramping constraints hold, we conclude that the defined solution is indeed feasible. This holds for any set $\mathcal{C}$ for which $\mathcal{C}_{\mathcal{R}}$ is representative.

□

From the proof, it can be seen why this might be too conservative. The minimum and maximum production occurring at the beginning or end of a period must be within the ramping limits of each other for each location and generation technology. In a realistic scenario, this would most likely not happen, since high demand and availability changes resulting in such a production shift are unlikely in consecutive hours. However, we might be able to relax these constraints if we can determine the maximum production change at a given location. We will not do so in this thesis, but motivate future work to focus on this.

4.3. Upper bound on the objective with blended weights

While this thesis primarily focuses on making robust decisions, another key aspect of using representative period reductions is ensuring that the objective function leads to a realistic estimate of system costs. This helps prevent outcomes where systems have unexpectedly high operational costs. By applying the bounded conical or convex hull method in combination with blended weights (as introduced in Section 3.3.4), we can derive an upper bound on the true system costs.

To achieve this, we have to use blended weights instead of the more commonly used Dirac weights. Under the Dirac approach, each original period $j \in \mathcal{J}$ was assigned to exactly one representative $r \in \mathcal{R}$ such that $W_{r,j} = 1$. With blended weights, we instead use the coefficients $\lambda_{r,j}$ for the weights. This modification allows us to prove that the optimal value of the reduced model provides an upper bound on the true objective value.

For completeness, we also show how the total weight per representative is computed under this approach:

$$W_r = \sum_{j=(s,p)\in\mathcal{J}} \pi_s \cdot \lambda_{r,j}. \tag{4.27}$$

In Theorem 4.1, we showed that the investment decisions from the reduced model yield a feasible solution to the original model. We now extend this result by demonstrating that, when using blended weights, the objective value of that constructed solution matches the objective value of the reduced model. Since the solution is feasible for the original model, this implies that the reduced model's optimal value serves as an upper bound on that of the original model.

Theorem 4.4. *Consider an instance of the GEP model as described in Section 3.1, but without inter-period ramping constraints. We find representative periods $\mathcal{R}$ that perfectly represent the original period set $\mathcal{J}$. Solve the reduced model to optimality, as formulated in* (4.4)*, where the weights are defined as in* (4.27)*. An optimal solution to the reduced model always exists. Denote the optimal values of the decision variables as $i^{\star}_{n,g}$, $prod^{\star}_{n,g,(r,h)}$, $f^{\star}_{e,(r,h)}$, and $l^{\star}_{n,(r,h)}$. Let $z^{\star}$ denote the corresponding objective value.*

If the weights $\lambda_{r,j}$ in (4.2) *satisfy Conditions 1 and 2 for every $j \in \mathcal{J}$, then the variables $prod_{n,g,(j,h)}$, $f_{e,(j,h)}$, and $l_{n,(j,h)}$, as defined in* (4.5) *through* (4.7)*, together with the investment decisions $i^{\star}_{n,g}$, will form a feasible solution to the original model. Moreover, its objective value in the original problem is equal to $z^{\star}$.*

Proof. In Theorem 4.1, we already showed that the proposed decision variables form a feasible solution to the original model. It remains to show that, using the defined weights, the cost of this solution equals the optimal objective value of the reduced model.

The objective function of the original model is given in (3.1). Substituting the proposed solution into this expression gives:

$$\sum_{(n,g)\in\mathcal{NG}} I_{n,g}\cdot U_{n,g}\cdot i^{\star}_{n,g} + \sum_{j=(s,p)\in\mathcal{J}} \pi_s \sum_{h\in\mathcal{H}} \left(\sum_{(n,g)\in\mathcal{NG}} V_{n,g} \sum_{r\in\mathcal{R}} \lambda_{r,j}\cdot prod^{\star}_{n,g,(r,h)} + \sum_{n\in\mathcal{N}} V^{\text{loss}} \sum_{r\in\mathcal{R}} \lambda_{r,j}\cdot l^{\star}_{n,(s,r,h)} \right). \tag{4.28}$$

We can reorder these sums to obtain the following:

$$\sum_{(n,g)\in\mathcal{NG}} I_{n,g}\cdot U_{n,g}\cdot i^{\star}_{n,g} + \sum_{r\in\mathcal{R}} \sum_{j=(s,p)\in\mathcal{J}} \pi_s\cdot\lambda_{r,j} \sum_{h\in\mathcal{H}} \left(\sum_{(n,g)\in\mathcal{NG}} V_{n,g}\cdot prod^{\star}_{n,g,(r,h)} + \sum_{n\in\mathcal{N}} V^{\text{loss}}\cdot l^{\star}_{n,(s,r,h)} \right). \tag{4.29}$$

Using (4.27) gives:

$$\sum_{(n,g)\in\mathcal{NG}} I_{n,g}\cdot U_{n,g}\cdot i^{\star}_{n,g} + \sum_{r\in\mathcal{R}} W_r \sum_{h\in\mathcal{H}} \left(\sum_{(n,g)\in\mathcal{NG}} V_{n,g}\cdot prod^{\star}_{n,g,(r,h)} + \sum_{n\in\mathcal{N}} V^{\text{loss}}\cdot l^{\star}_{n,(s,r,h)} \right). \tag{4.30}$$

This is exactly the objective function of the reduced model with blended weights. Since the decision variables are optimal for that model, the resulting cost equals $z^{\star}$. This completes the proof. □

This upper bound is particularly useful when solving the operational part of the original model under fixed investments is already computationally expensive. The result also extends to more general model formulations, as shown in Appendix A.

4.4. Conclusion

We now have two strong theoretical motivations for using the convex hull or bounded conical hull method when selecting representative periods. First, if such a hull is constructed and the reduced model has no loss of load in its optimal solution, we can guarantee a feasible solution without loss of load for the original model. Second, the optimal objective value of the reduced model when using blended weights provides an upper bound on the original model's costs. These results are derived under the assumption that the representative periods form a bounded conical or convex hull. Finding an exact solution can be computationally expensive. For this reason, we use the earlier proposed greedy algorithm in our experiments. In the next chapters, we examine the practical implications of the hull-based methods, both in cases where a perfect representative set is found and where it is potentially harder to find one. The theorems from this chapter provide a strong motivation for improving and optimizing hull-based methods in practice, so that their theoretical guarantees can be translated effectively into real-world applications.

5

Feasibility and optimality of the greedy hull methods compared to traditional methods

In Chapter 4, we proved that using representatives that form a bounded conical or convex hull, under the assumption that such a hull is found, ensures a feasible solution. Feasibility, better defined here as meeting all demand, is essential for energy systems, and commonly used selection methods do not guarantee this same level of feasibility. However, it is still possible that other methods, by coincidence, find a set of representative periods that also result in a feasible solution. Moreover, alternative methods could, in some cases, lead to lower costs for the full model.

To illustrate this, consider a simplified example with only three periods in total, with a single location, and one time step per period, all with the same demand. At this location, we can choose between investing in **gas** or **wind onshore**. Satisfying demand in any given period requires one fully operational unit of either technology. However, the availability of wind onshore varies:

Period	Wind Availability
1	0.0
2	0.6
3	1.0

Table 5.1: Wind availability and demand across the three periods.

We assume that the operating costs of gas are high while investment costs for both gas and wind onshore are relatively low, making it more cost-effective to invest in two units of wind onshore rather than operating one unit of gas. In an optimal full-model solution, the best strategy would be:

- Invest in one unit of gas, since period 1 has no wind and demand must still be met.
- Invest in two units of wind onshore, which can satisfy demand in periods 2 and 3.

Suppose now that we choose two representative periods from the three options. A convex hull is formed by the extreme periods, meaning the representatives would be period 1 and period 3. Since period 3 requires only one unit of wind onshore to meet demand, the model would install just one unit. However, in the full model, this would lead to the use of gas in period 2, even though, in the optimal solution, this demand would have been met by one additional wind onshore unit. Thus, using a convex hull results in higher costs compared to the optimal solution of the full model.

Alternatively, k-medoids finds two clusters and chooses the periods best representing that cluster as cluster centers. Since periods 2 and 3 are closer to each other than periods 1 and 2, it will see period 1 as one cluster and the other two as one cluster. Suppose it then selects periods 1 and 2 as cluster centers. Since period 2 requires 0.6 wind availability, the model would invest in two units of wind onshore and one unit of gas, matching the optimal solution of the full model.

However, k-medoids can also fail. If period 2 had a lower availability (e.g., 0.4), the k-medoids method might instead select periods 2 and 3 as cluster centers. This would lead to an incorrect assumption in the reduced model that gas is unnecessary. As a result, the model would invest only in wind onshore, but this would be infeasible in period 1, where no wind is available.

This simple example highlights the importance of investigating the effect of the different methods on both the feasibility and optimality of the produced solution. To examine the potential (dis)advantages of the methods, we compare them using artificially created case studies that exhibit specific characteristics. We then evaluate the clustering methods introduced in Section 3.3: k-means, k-medoids, greedy convex hull and greedy bounded conical hull on those case studies. In Section 5.1, the setup and motivation behind the case studies is explained in more detail. The metrics on which we evaluate are further explained in Section 5.2. The results will be presented in Section 5.3. The goal of this chapter is not to identify the best method, but to build insight. Since we experiment on simplified, artificial case studies, the results may not fully reflect realistic scenarios. Still, understanding how and why certain methods behave the way they do will help us reason through more realistic case studies later on.

5.1. Case studies

We created four case studies, each with its own characteristics, to evaluate feasibility and optimality under all four methods. Since we use a greedy method to construct the hulls, we are not certain that a valid hull will always be found. Therefore, we start with a case study where the convex hull is known beforehand. This way, the hull methods, which try to find the periods forming the corner points of the hull, are both expected to succeed and should be able to generate a feasible solution with a small number of representative periods. In the following case studies, we gradually introduce more variation into the data points, making it more difficult for the approximate methods to find a hull. This allows us to observe when k-means or k-medoids might provide a better approximation of the optimal solution of the full model. Across all cases, whether a hull is found or not, we explore the trade-off between feasibility and optimality. Below, we describe and motivate each of the case studies in more detail, while Figure 5.1 gives a visual overview.

For all four case studies, the same three days from a provided dataset were selected as initial periods. These days were chosen to ensure they are sufficiently distinct in terms of their demand and availability of generation technologies profiles. They consist of 20 locations and 24 time steps per period. Since we cluster on availability for three sorts of generation technologies and the demand values, this results in a 1920-dimensional space in which we find the representative periods. Based on these three initial periods, we created different case studies that include the three initial days along with 90 additional days, which are distributed in some way between or around the initial three periods. To solely focus on the research question of this chapter, we assume no inter-period constraints and no scenarios. For the scenarios, they can be omitted without loss of information, since we are not considering inter-period constraints. Collecting the created data into a single scenario or distributing it across multiple equally likely scenarios would thus not affect the feasibility, as long as we assume that the cross-scenario method would be used for applying the representative periods.

While each period consists of 1920 dimensions, Figure 5.1 shows only two of those dimensions to give an indication of how the created periods are distributed relative to the initial three periods. The initial samples in the graphs are larger in size and shown in a different color, while the generated points are visualized with small dots. In two of the case studies, we also show two values from a different location to better visualize the correlation between them. The following descriptions explain the case studies one by one:

Convex The periods for this case study are visualized in the upper left part of Figure 5.1. We want to start with a case study for which we know that a representative set forming a convex or bounded conical hull exists with only three representatives. To do so, we generate 90 artificial points, with all new points being convex combinations of the original three. We ensure that the generated points are not too close to the boundary. This way, it should be easier for the greedy hull methods to correctly identify the hull. We achieve this by using the softmax operation to generate convex weights from three initial values randomly drawn between -1 and 1. Figure 5.1 clearly shows that the data points are concentrated around the center rather than the boundaries of the hull.

Centered cluster The periods for this case study are visualized in the upper right part of Figure 5.1. To add more variation within the data points, we again assume that the original three points are outliers, but the other points are no longer necessarily convex combinations of them. Each period is created by adding random noise to the average values of the three initial periods. The direction and size of the noise vary per location, but remain constant across

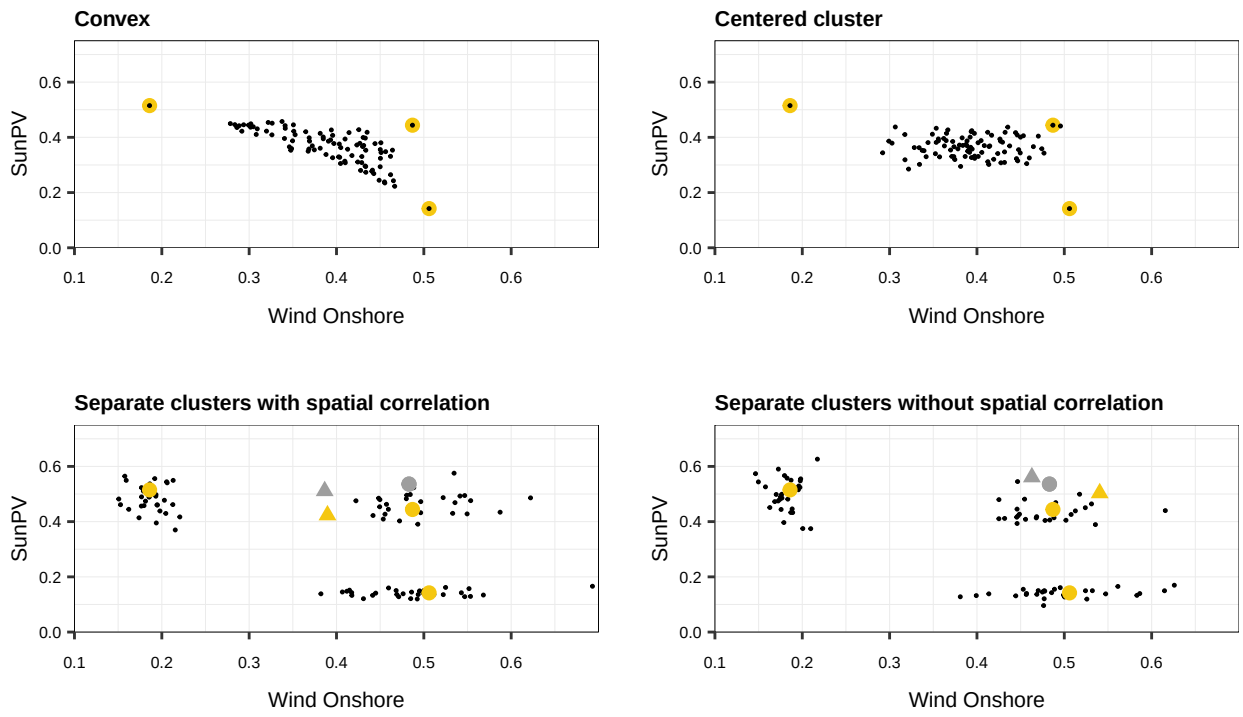

Figure 5.1: Distribution of Wind Onshore and SolarPV availability values in the created case studies, at the 12th hour of each period. The large points represent the initial periods, from which the other periods were derived. Each case study consists of 93 different periods. All black and yellow points correspond to values in the Netherlands. The triangles in the plots at the bottom resemble one specific period, period 44. The grey points correspond to one of the initial points and period 44, but for Germany instead of the Netherlands.

all time steps at that location. While the three initial periods can still be seen as outliers, they may not form a perfect hull for the other periods. This can also be observed in the figure, where one of the new periods at the top right lies outside the convex hull formed by the yellow dots.

Separate clusters with spatial correlation In the previous two case studies, the majority of the periods were centered between the initial three periods, making the initial samples act like outliers. In this and the next case study, we explore the behavior of the methods when that is not necessarily the case. This should make it harder for the greedy hull methods to find a hull using a small number of representative periods. In this case study, shown in the bottom left of Figure 5.1, we consider three clusters, each centered around one of the initial periods. Each period within a cluster is generated by adding random noise to the values of the cluster center. The size and direction of the noise are consistent across locations and time steps within each period, but vary between availability of different types of generation technologies or demand. In Figure 5.1, we also show values at a different location for two of these periods. This highlights the spatial correlation: for each created period, the shift occurs in a similar direction across all locations. We chose this setup assuming that it may make it easier to identify a hull in the high-dimensional space with the greedy algorithm. However, a larger number of representative periods will likely be needed to construct a hull compared to the previous cases.

Separate clusters without spatial correlation The periods for the final case study are shown in the bottom right of Figure 5.1. This case study follows a similar structure to the previous one, with three clusters around the original periods. However, we now add different random noise at different locations. This is visualized by the triangles, which represent the values of the same period at another location. It can be seen that these are shifted from the initial value at that location in a different direction. In general, this may be a more realistic setting, as more sun at one location does not necessarily imply more sun elsewhere. However, it likely increases the difficulty of identifying a hull using the greedy methods, since there is less correlation between periods.

There are, of course, many other possibilities for designing artificial case studies. The goal here is not to provide a comprehensive overview, but rather to investigate the behavior of individual methods in a controlled setting, and to gain more intuition and insight into how they perform. In particular, we are interested in identifying the conditions under which the greedy hull methods succeed in finding a valid hull and generating good solutions, as well as the situations in which they fail. This way, we can better understand the limitations of these approaches and identify where improvements might be needed.

5.2. Evaluation method

In this section, we define how we evaluate the effect of the different methods on both optimality and feasibility. First, we introduce two metrics that will be used for this purpose. After that, we describe the experimental setup. Finally, as we need to distinguish between in-sample and out-of-sample evaluation, we explain this distinction at the end of this section.

5.2.1. Metrics

We will define two metrics, since we want to see the effect on optimality as well as the effect on feasibility for all methods. As seen in the introduction of the chapter, there are cases in which the hull points might provide investment decisions that result in a feasible solution, but might not be the most optimal choice in terms of costs, whereas other methods might be able to identify those representatives important for optimality earlier on. By looking at both optimality and feasibility individually, we can identify when this behavior occurs. Moreover, we might be able to see at which point adding more representatives is not necessary in terms of feasibility, but can help in finding a more optimal set of investment decisions.

First, consider the evaluation of optimality. For a given combination of distance metric, clustering method, and number of representative days, we first solve the reduced problem. The resulting investment decisions are then fixed and used as input parameters in a second run of the original problem, where only the second-stage variables remain as decision variables. This provides the optimal total cost when investment decisions are obtained from the reduced problem. Finally, this cost is compared to the optimal cost obtained by solving the full problem without any reduction. The percentage difference represents the relative regret when using representative periods rather than solving the full model directly. This value is always non-negative, as costs can never be lower than the optimal value.

For feasibility, we want to measure the occurrence of loss of load. Since a penalty is applied for loss of load, any infeasibility will already be reflected in a higher relative regret. However, we want to explicitly identify when a higher relative regret is due to an increased total loss of load. Therefore, we count the number of time steps where loss of load occurs. This count is then compared to the number of time steps with loss of load in the optimal solution of the original problem. The difference between these counts indicates additional time steps with loss of load, introduced when using representative periods. This difference can be negative, as the reduced problem might select investment decisions that result in no loss of load, while the full problem might allow some loss of load if it leads to a lower overall objective value.

5.2.2. Experimental setup

We specifically want to find out the effect of using the three clustering methods for the selection of representative periods. For a meaningful comparison of the metrics, all other parameters have to be kept constant. However, we also want to analyze the effect of different distance metrics and how the results change when increasing the number of representative periods. To account for this, we repeat each experiment for different parameter settings. Specifically, we vary:

- Number of representative periods, ranging from 3 to 41, in steps of 2.
- Distance metric, where we compare squared euclidean distance and cosine distance.
- Clustering method, where we compare k-means, k-medoids, the greedy bounded conical hull and the greedy convex hull approach.

The distance metrics were chosen as squared euclidean distance is mostly used, while the cosine distance showed some initial good results when used in combination with the greedy convex hull method [23]. The cosine distance can, however, not be used in combination with the greedy bounded conical hull method. This method requires a distance metric for which the distance to the zero vector is defined. For the greedy bounded conical hull method, we will thus only show results in combination with the squared euclidean distance.

Since k-means and k-medoids involve a random initialization step, the selected representative periods might differ across runs. To account for this variability, we generate ten different random seeds for each number of representative periods and repeat all experiments accordingly. In the results, we present the median values along with the 25% and 75% quantiles to illustrate the variability introduced by randomness.

5.2.3. Out-of-sample evaluation

The metrics defined in Subsection 5.2.1 provide insight into the in-sample performance of the methods. However, in practice, we want solutions that are robust to variations in the data rather than optimal for a single specific realization. If the selection of representative periods does not generalize well, the method may overfit to a specific realization, leading to investment decisions that might not be the most cost-effective approach for different realizations.

To assess robustness against variations in the samples, we include an out-of-sample evaluation. For this evaluation, we create test sets with a similar underlying distribution to the artificially generated case studies. The investment decisions obtained from the reduced problem are then used to solve the dispatch variables of the test set to optimality. The resulting objective function and number of time steps with loss of load are compared to the corresponding values obtained when solving the full problem for the test set, resulting in the same two metrics as for the in-sample evaluation.

5.3. Results

In this section, we present the results from the four case studies with different distributions of periods described in Section 5.1 and evaluate them using the relative regret and increase in time steps with loss of load as metrics. The greedy convex hull, greedy bounded conical hull, k-means and k-medoids method are compared, as outlined previously in Section 5.2. In particular, we aim to assess the impact of each method on both feasibility and optimality, and to identify where the approximate hull methods begin to reach their limitations.

There are two variants per experiment, as both the cosine distance and squared euclidean distance are used in finding representative periods. Each case study will be examined individually to highlight the relevant conclusions. Overall, we observe a strong correlation between relative regret and the additional time steps with loss of load, particularly due to the penalization of loss of load in the objective function. For this reason, we chose not to present the latter graphically, as the trends would be similar to those observed by looking at the relative regret. However, it is useful to highlight the cases where no additional loss of load occurs in comparison to the optimal solution. To do so, we include additional data in tables, with a complete overview in Appendix B.

Convex For this case study, we expect both greedy hull methods to easily identify the hull, since the three original periods were designed to form one. This allows us to clearly observe the effect on feasibility compared to the other methods.

In Figure 5.2, we can see the relative regret for the in-sample and out-of-sample experiments and the two different distance metrics. All four graphs show a very similar pattern. In all cases, the greedy convex hull method has best performance from the beginning on. It obtains a relative regret close to 0 with a much lower number of representative periods used than k-means and k-medoids: with five representative periods the greedy convex hull method in combination with the cosine distance leads to a relative regret under 1%. For the greedy bounded conical hull method, we observe a similar outcome when using the squared euclidean distance. For k-means and k-medoids, the median relative regret drops below 1% only when using 15 and 11 representative periods, respectively. This indicates that a larger number of representative periods is required to achieve comparable performance in terms of costs. For these two methods, an even larger number of representative periods is required to achieve consistent results. The 75th percentile curves for both methods still exhibit spikes beyond 21 representative periods, indicating that for at least 2 out of 10 seeds the relative regret exceeds 25%.

As expected in this case study, the hull methods need only a small number of representatives to come close to an optimal solution. We also analyze the infeasibility in more detail. For this, we examine Table 5.2. As the distance metric seems to have only a small influence in Figure 5.2, we chose to only show the values corresponding to the squared euclidean distance. Also, we show only 3 up to 11 representative periods, since we want to focus on the behavior when using a low number of representatives. The full table with all values can be found in Appendix B.

The results in Table 5.2 confirm that a hull was indeed found with three representative periods: there are no time steps with loss of load in either the in-sample or out-of-sample evaluation. Since the optimal solution of the full model had seven time steps with loss of load, this even results in a negative value in the table. The absence of infeasibility does

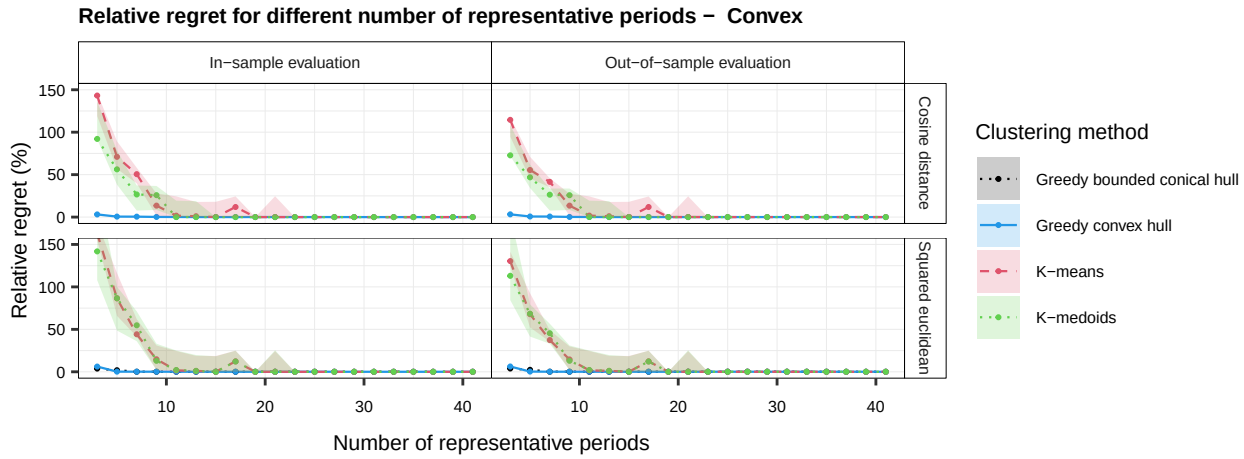

Figure 5.2: Relative regret (%) per number of representative periods in the in-sample and out-of-sample experiments on the convex case study, for both the cosine distance and squared euclidean distance. The different methods for selecting the representative periods are in different colors. The area for k-means and k-medoids represents the 25% and 75% quantiles.

not of course not imply that the most optimal investment decisions were made. This becomes clear when increasing the number of representative periods for the greedy convex hull method, from three to five, which results in more time steps with loss of load but also leads to a lower relative regret. This indicates that the first three representatives were essential for feasibility, while the additional two improved optimality. The table also shows that for k-means and k-medoids, even with 11 representative periods, there are still some time steps with additional loss of load compared to the optimal solutions. Moreover, these are median values and do not capture the full variation in the results, as shown by the quantiles in Figure 5.2.

In general, we can conclude that for this case study, where both a convex and bounded conical hull are easily found, the greedy hull methods show important advantages over the other two methods. They achieve zero infeasibility and a low relative regret with fewer representative periods and are less affected by randomness than k-means and k-medoids. This randomness caused sudden spikes in relative regret for certain seeds, even when a larger number of representative periods was used. In this case study, we also observed that the results were similar across both distance metrics and in both the in-sample and out-of-sample evaluations.

Centered cluster In the case study with one centered cluster of periods with the three initial periods as outliers, the three initial periods are no longer forming a hull. We would thus expect that both hull methods might need more representative periods to form a valid hull and to achieve no infeasibility.

We can see from Figure 5.3 that both greedy hull methods again outperform the other two methods, in terms of relative regret. The greedy convex hull and greedy bounded conical hull methods already approach a solution with nearly 0% relative regret with 5 representative periods in each of the graphs, while a similar relative regret is reached only when using at least 13 representative periods in the other two methods. It can also be seen that there is again fluctuation in the case of k-means and k-medoids, and that there are quite large differences in seeds. For example, with 11 representative periods, the 25% quantile shows a relative regret close to 0% for k-medoids with cosine distance, while the 75% quantile is still above 150%. In terms of distance metrics, the behavior in the results is quite similar between the two, not indicating that one outperforms the other.

To get a more detailed overview of infeasibility, we can look at Table 5.3. As before, only a limited number of representative periods is shown, and only for the squared euclidean distance. We observe that with just three representative periods, the greedy bounded conical hull method already results in no additional loss of load in the original model. For the greedy convex hull method, this is achieved with five representative periods. Also, when using one of the other methods, the number of time steps with additional loss of load is higher than in the previous case study. This suggests that the differences between the hull methods and the clustering algorithms are even more visible in this case. We also see that there seems to be more infeasibility in the out-of-sample evaluation for these two, while this is not the

Method	Representatives	In-sample		Out-of-sample	
		Relative regret (%)	LoL	Relative regret (%)	LoL
greedy bounded conical hull	3	3.8	-7	3.9	-7
	5	1.8	0	2.3	0
	7	0.1	0	0.1	0
	9	0.1	0	0.1	0
	11	0.1	0	0.1	0
greedy convex hull	3	6.2	-7	6.2	-7
	5	0.2	0	0.3	0
	7	0.1	0	0.1	0
	9	0.1	0	0.1	0
	11	0.1	0	0.1	0
***k*-Means** (Median)	3	163.1	686	130.4	531
	5	86.6	357	67.9	268
	7	44.2	190	37.4	125
	9	14.7	56	14.5	49
	11	1.7	10	1.7	10
***k*-Medoids** (Median)	3	141.8	580	113.1	438
	5	86.6	361	68.4	287
	7	54.7	213	45.4	165
	9	12.9	47	12.9	49
	11	2.1	10	2.2	10

Table 5.2: Relative regret (%) and increase in time steps with loss of load (LoL) for the in-sample and out-of-sample experiments on the "Convex" case study. Multiple combinations of the number of representative periods and methods for finding the representative periods are shown. Only the results for the squared euclidean distance are in the table.

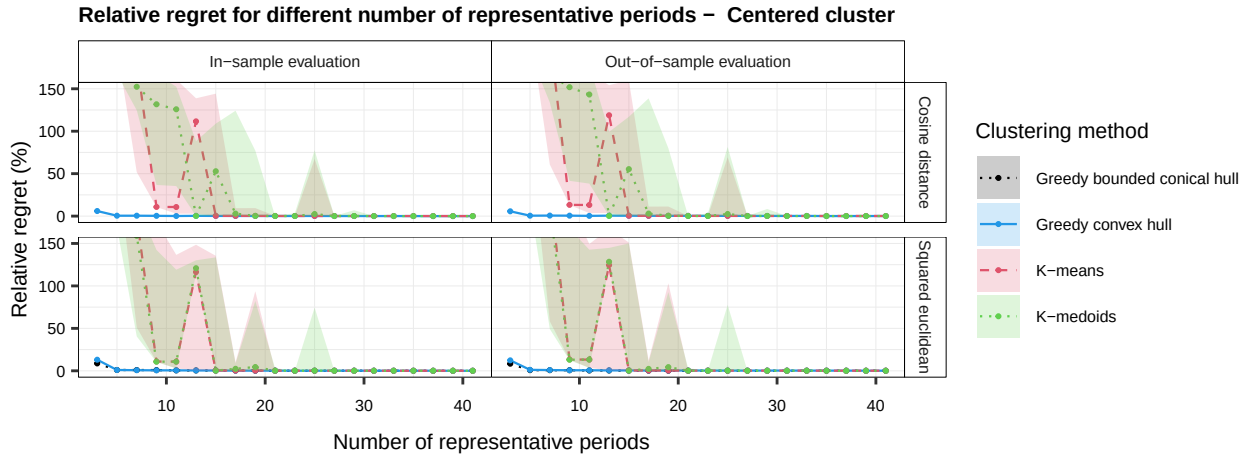

Figure 5.3: Relative regret (%) per number of representative periods in the in-sample and out-of-sample experiments on the "Centered cluster" case study, for both the cosine distance and squared euclidean distance. The different methods for selecting the representative periods are in different colors. The area for *k*-means and *k*-medoids represents the 25% and 75% quantiles.

case for the hull methods. The results for *k*-means and *k*-medoids are too similar to draw a meaningful distinction between them.

We can again conclude that, for this case study, the hull methods show that only a small number of representatives is needed to form a hull and thus ensure no additional loss of load. The instability observed for *k*-means and *k*-medoids at lower numbers of representative periods was even more pronounced than in the previous case study. Finally, no substantial differences were observed between in-sample and out-of-sample evaluations or the distance metrics.

Separate clusters with spatial correlation In this case study, we expect that the greedy hull methods will require more representative periods to achieve a low relative regret, since there is more variability. We are particularly interested in how quickly a hull is found.

Method	Representatives	In-sample Relative regret (%)	In-sample LoL	Out-of-sample Relative regret (%)	Out-of-sample LoL
greedy bounded conical hull	3	8.7	-7	8.3	-7
	5	1.0	0	0.8	0
	7	0.9	0	0.7	0
	9	0.8	0	0.7	0
	11	0.5	0	0.4	0
greedy convex hull	3	13.1	11	12.3	9
	5	1.0	0	1.1	0
	7	0.7	0	0.9	0
	9	0.6	0	0.6	0
	11	0.5	0	0.6	0
***k*-Means** (Median)	3	301.1	1608	329.7	1911
	5	280.0	1283	309.5	1628
	7	180.8	682	212.6	803
	9	10.9	82	13.4	111
	11	10.8	81	13.2	110
***k*-Medoids** (Median)	3	202.2	888	233.3	1051
	5	196.4	859	215.8	998
	7	159.0	565	192.4	808
	9	10.8	80	13.1	109
	11	10.8	82	13.2	111

Table 5.3: Relative regret (%) and increase in time steps with loss of load (LoL) for the in-sample and out-of-sample experiments on the "Centered cluster" case study. Multiple combinations of the number of representative periods and methods for finding the representative periods are shown. Only the results for the squared euclidean distance are in the table.

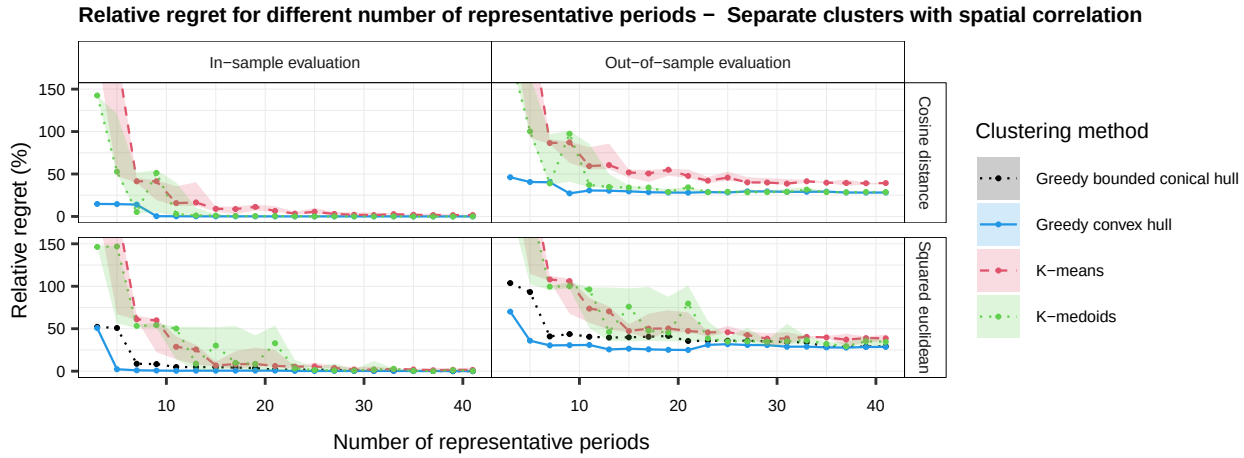

Figure 5.4: Relative regret (%) per number of representative periods in the in-sample and out-of-sample experiments on the "Separate clusters with spatial correlation" case study, for both the cosine distance and squared euclidean distance. The different methods for selecting the representative periods are in different colors. The area for *k*-means and *k*-medoids represents the 25% and 75% quantiles.

The relative regret graphs are shown in Figure 5.4. For the squared euclidean distance, the greedy convex hull method already achieves a near-zero relative regret, which is better than expected. In general, both hull-based methods show lower relative regret in most cases compared to the clustering algorithms. They continue to show fluctuations depending on the seed, even when more representative periods are used. One example is the combination of 21 representative periods, the squared euclidean distance, and the *k*-medoids method. Here the median value is above 40%. However, the fluctuations are less extreme than in the previous case study.

The most interesting difference in Figure 5.4 is that the out-of-sample evaluation shows that for all methods the chosen investment decisions do not generalize as well as before. By design, there was a lot of variation around the extreme values for this case study. It could thus be that for out-of sample case, this resulted in some extreme periods which were not covered at all by the investment decisions that were optimal for the first sample, the sample for which the

representative periods were found.

Method	Representatives	In-sample		Out-of-sample	
		Relative regret (%)	LoL	Relative regret (%)	LoL
greedy bounded conical hull	3	52.2	480	103.7	618
	5	50.8	444	93.2	568
	7	8.7	107	40.9	226
	9	8.4	117	43.7	242
	11	4.8	48	40.5	161
greedy convex hull	3	50.8	301	70.1	436
	5	2.3	-4	35.9	123
	7	1.0	-9	30.4	109
	9	0.7	-10	30.6	109
	11	0.4	-9	30.8	123
***k*-Means**	3	194.0	1358	238.2	1365
(Median)	5	190.9	1269	234.4	1270
	7	61.0	492	108.0	604
	9	59.9	470	106.1	573
	11	28.7	254	73.6	367
***k*-Medoids**	3	146.3	1025	192.7	1086
(Median)	5	146.9	975	192.8	1074
	7	53.3	432	99.5	544
	9	54.1	430	99.9	548
	11	50.2	403	96.3	516

Table 5.4: Relative regret (%) and increase in time steps with loss of load (LoL) for the in-sample and out-of-sample experiments on the "Separate clusters with spatial correlation" case study. Multiple combinations of the number of representative periods and methods for finding the representative periods are shown. Only the results for the squared euclidean distance are in the table.

To further investigate this behavior, we turn to Table 5.4. The increase in time steps with loss of load between the in-sample and out-of-sample evaluation is visible for all methods. Notably, the greedy convex hull method is able to achieve zero infeasibility for the in-sample periods, while the greedy bounded conical hull method is not. This highlights that, depending on the structure of the data, one method may successfully identify a feasible hull while the other does not, even when both are theoretically possible. As in previous case studies, both k-means and k-medoids result in a significantly higher number of time steps with additional loss of load compared to the hull-based methods.

In this case study, the greedy convex hull method again results in the best performance, and there is a clear distinction between the hull methods and the clustering algorithms. The greedy bounded conical hull was not able to find a hull with fewer than 11 representative periods, which highlights that this case study indeed contains more variation, especially around the extremes, than before. This is also seen in the out-of-sample results, which have higher relative regret than before for each of the methods. In terms of distance metrics, patterns were similar.

Separate clusters without spatial correlation Without spatial correlation, obtaining a hull can become more difficult for the greedy methods, as an increase in availability of a generation technology at one location does not imply a similar increase at others. As a result, we expect both hull methods to require more representative periods to achieve zero infeasibility.

The effect is clearly visible in the graphs in Figure 5.5. A clear distinction between the hull methods and the clustering algorithms cannot be made, and all methods perform worse compared to the previous case studies. Notably, both k-means and k-medoids require a very large number of representative periods, 29 for k-medoids and 37 for k-means (in the case of cosine distance), to reduce the relative regret below 10%. This is significantly higher than in previous case studies, indicating that when spatial correlation is removed, more representative periods are required across all methods to capture enough information. This highlights that the number of representative periods needed to achieve a given level of relative regret is highly dependent on the structure of the dataset.

For the first time, we also observe a noticeable difference between the cosine distance and the squared euclidean distance, particularly in combination with the greedy convex hull method. With the cosine distance, this method achieves the lowest regret across most numbers of representative periods, whereas with the squared euclidean distance, its regret remains significantly higher than that of the other methods. This suggests that when a valid hull is not guaranteed, the performance of the greedy convex hull method can vary considerably depending on the choice of distance metric and the specific representative periods selected.

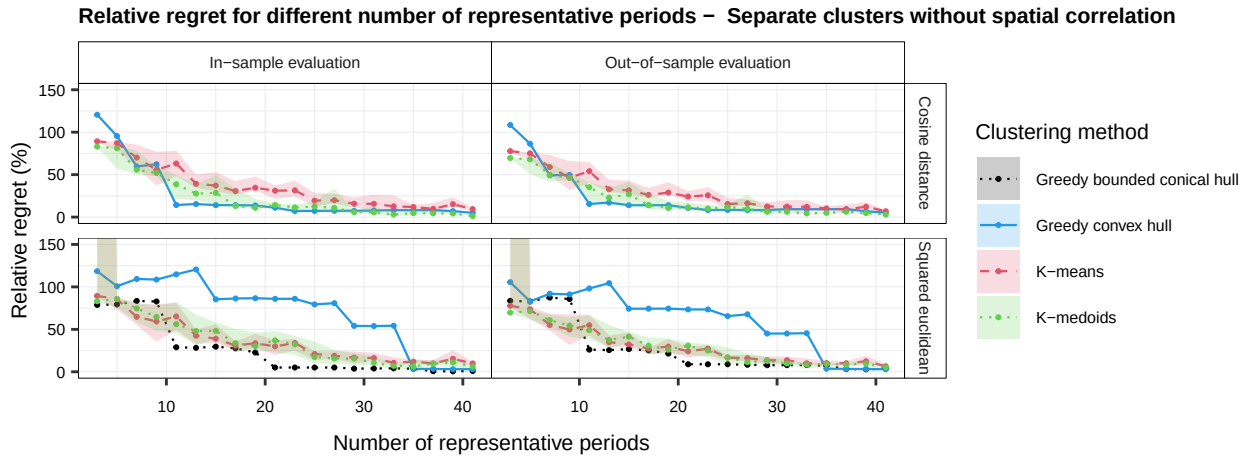

Figure 5.5: Relative regret (%) per number of representative periods in the in-sample and out-of-sample experiments on the "Separate clusters without spatial correlation" case study, for both the cosine distance and squared euclidean distance. The different methods for selecting the representative periods are in different colors. The area for k-means and k-medoids represents the 25% and 75% quantiles.

Method	**Representatives**	**In-sample**		**Out-of-sample**	
		Relative regret (%)	**LoL**	**Relative regret (%)**	**LoL**
greedy bounded conical hull	3	78.8	431	83.6	497
	5	79.4	394	82.8	416
	7	83.4	453	87.4	512
	9	82.8	412	85.7	435
	11	28.9	254	26.1	264
greedy convex hull	3	118.5	710	105.6	694
	5	100.8	611	82.9	564
	7	109.3	592	91.9	519
	9	108.6	567	91.2	532
	11	114.8	566	98.2	544
k-**Means** (Median)	3	89.4	748	77.8	734
	5	85.6	690	73.7	635
	7	64.8	596	55.2	542
	9	59.4	518	49.8	478
	11	65.2	567	54.8	532
k-**Medoids** (Median)	3	83.1	694	69.7	655
	5	85.9	693	71.5	606
	7	74.6	623	60.9	587
	9	64.6	543	54.4	534
	11	56.1	483	49.0	465

Table 5.5: Relative regret (%) and increase in time steps with loss of load (LoL) for the in-sample and out-of-sample experiments on the "Separate clusters without spatial correlation" case study. Multiple combinations of the number of representative periods and methods for finding the representative periods are shown. Only the results for the squared euclidean distance are in the table.

To assess whether the high relative regret is indeed associated with additional loss of load, and thus supports the hypothesis that a hull is not found, we examine the detailed results for up to 11 representative periods in Table 5.5. We observe that the elevated relative regret, particularly for the greedy convex hull method, corresponds to a substantial increase in time steps with additional loss of load compared to the previous case studies. A value of zero or below is not reached for any method–distance combination within this range of number of representative periods, as shown in the complete overview in Table B.7 and B.8. This suggests that a valid hull is not formed when using a small number of representative periods. This may be due to the approximate algorithm failing to identify the correct periods, or because no convex or bounded conical hull exists with that number of representative periods. In either case, it is worth investigating how the method can be improved, either by selecting more relevant periods or by exploring alternative ways to improve feasibility. This is particularly relevant in light of the previous case studies, where the hull methods

clearly demonstrated their advantages when a valid hull could be found, resulting in both zero infeasibility and low relative regret.

5.4. Conclusion

The experiments in this chapter compared greedy hull methods and traditional clustering approaches for selecting representative periods, focusing on feasibility and optimality across case studies with distinct characteristics. Results show that the greedy convex hull and greedy bounded conical hull methods are highly effective when a hull can be formed. In these cases, feasibility was ensured, and relative regret dropped below 1% with just a few representatives. Adding more periods further improved the regret once feasibility was achieved. In contrast, k-means and k-medoids reached similarly low regret only with more representatives and still failed to guarantee feasibility. This supports the idea that hull methods are well-suited for finding representatives essential to feasibility. This also suggests that once a hull is identified, other methods could be used to add further points, an idea explored in the next chapter. A key shortcoming of the clustering methods was their variability due to randomness. In some cases, increasing the number of representatives led to a sharp rise in median regret. The spread between the 25% and 75% quantiles also remained visible even with more representatives.

The final case study, with less spatial correlation, was challenging for all methods. None achieved low regret with a limited number of representatives. Since the hull methods suddenly were not able to ensure feasibility, we suspect that either no convex or bounded conical hull exists for that number of representatives in that dataset, or the approximation algorithm failed to find it. Possible improvements are discussed in the next chapter. Since trends in in-sample evaluation mirrored those in out-of-sample evaluation, we will focus on in-sample results in the next chapter. Out-of-sample evaluation returns in the later European case study.

6

Enhancing the representative set with artificial periods for feasibility

In Chapter 4, we proved that when the set of representative periods formed a convex or bounded conical hull containing all other periods, a solution with no additional infeasibility could be constructed from the optimal solution of the reduced problem. To test this in practice using a greedy implementation for finding the hull, we analyzed the infeasibility and relative regret on four different case studies in Chapter 5. In the first three cases, the hull was identified with a small number of representatives, leading to a feasible solution for the full problem with lower relative regret compared to alternative methods. However, in the final case study, both hull methods were unable to construct a hull for the given number of representatives. As a result, the relative regret was comparable to or worse than k-means and k-medoids, and feasibility was no longer ensured. The key difference in this last case was that there was less correlation between periods.

Our hypothesis is that in high-dimensional spaces, distributions with less correlation between dimensions and more distance between the vectors, make it more difficult to construct a convex or bounded conical hull. In this context, "more difficult" means that the greedy hull method may fail, while a higher-runtime algorithm could still succeed, or that no convex or bounded conical hull consisting solely of corner points from the representative set exists for the given number of representative periods. The latter can even occur in lower dimensions: a clear example is a circle, where a convex hull composed only of points from the circle would require all points on that circle (See Figure 6.1).

To address this, we propose modifying our approach by introducing representative periods that are not part of the original set of periods. The idea of “artificial” representatives is not new: by finding cluster centroids, k-means does essentially the same. The values of the artificial representatives that we will use are chosen to be in the extremes, to form a bounding polytope that contains the hull. This can lead to a set that is larger than the actual hull in terms of volume, but requires fewer representatives. In this way, we are able to ensure feasibility with fewer representatives. This can, however, come at the cost of optimality, as the newly added extreme representatives may lead to unnecessary investments. To explore this, we first illustrate the approach with a two-dimensional example in Section 6.1. To further refine the method, we also investigate the effect of adding these points earlier in the process, allowing them to be taken into account when finding the rest of the alternatives.

To further optimize this process, we can make a distinction between representative periods added for feasibility in the early selection process (e.g., representatives that are on the extreme side) and representative periods added for optimality (e.g., cluster centers added after the extremes). This idea is similar to previous studies where worst-case samples were added in later stages, sometimes even after optimizing, to reduce infeasibility in k-means and k-medoids clustering [18], but now the extreme points are created before clustering instead of being taken from the data after clustering. We expect that the influence on optimality depends largely on the weights assigned to these artificial representatives compared to the other representatives. To test this, we also apply the blended weights of Subsection 6.1.3. In Section 6.2, we extend the methods identified in two-dimensions to higher dimensions and apply them to the case study from the previous chapter.

6.1. Finding a bounding polytope for the hull in a two-dimensional space

While the high-dimensional space in which we are looking for representatives makes it harder to obtain a hull, problems can occur in lower-dimensional examples as well. To understand why this happens and what possible directions for improvement are, we start this chapter by looking at a two-dimensional example. We first present the example in Subsection 6.1.1. In this example, the data is distributed in such a way that every period belongs to the convex hull, making it impossible to approximate the convex hull well with just a few data points. We thus propose a method of finding a bounding polytope for the convex hull with artificial representatives in Subsection 6.1.2. This approach does not extend to higher dimensions, so we try to identify the relevant points in the 2-dimensional example for generation expansion planning specifically, to see if we can gather insights that are useful for higher dimensions. This last part is presented in Section 6.1.4 and is the foundation for the methods proposed in Section 6.2.

6.1.1. Two-dimensional example

In the two-dimensional case, there is one value for demand and one value for availability per period. We ignore inter-period constraints. The goal is to test on an example for which finding a convex or bounded conical hull is not possible with a limited number of representative periods. For this example, we created 100 synthetic periods evenly distributed along a perfect ellipse in two-dimensional space. This is visualized in Figure 6.1, where demand has been scaled by dividing all values by the maximum demand to ensure it is on a scale similar to the availability. This was also done when finding the representatives, just as in the previous experiment. Since this example is for visualization purposes only, we do not use any additional normalization here. Figure 6.1 shows the representatives selected by the four previously tested methods when using three representatives, combined with the squared euclidean distance.

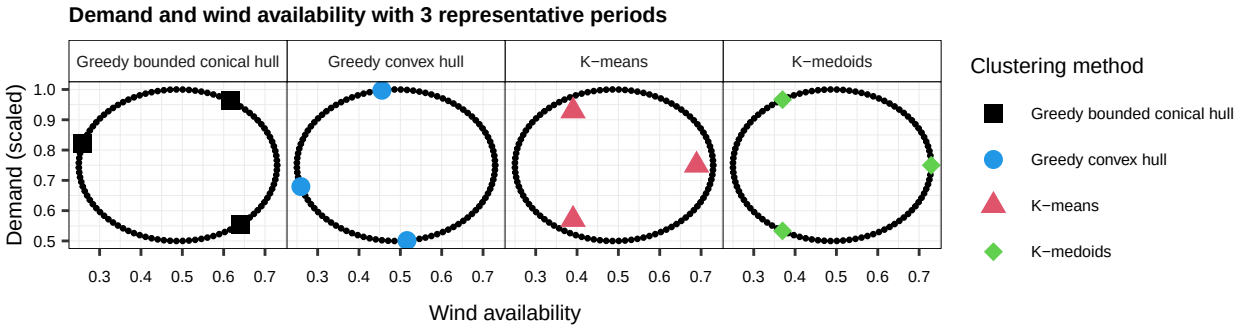

Figure 6.1: The full set of 100 periods (in black) on which the four representative period methods are applied for finding three representative periods. The larger markers indicate these representative periods selected by each method in combination with the squared euclidean distance metric.

In Figure 6.1, we see that in every case, all the samples fall outside the convex hull formed by the representatives, except for the representatives themselves. This is always the case for an ellipse: no matter which points are selected as representatives, the convex hull formed by them will always exclude the rest of the samples on the border. This means that feasibility cannot be guaranteed using just the selected representatives, at least not when those representatives are constrained to be original data points. We can conclude that the representatives found by the greedy convex hull method will not form a hull containing all periods, as will none of the other methods, as they also do not select representatives outside the ellipse. This is reflected in the results when we try to optimize the original periods under the investment decisions from the reduced model. We used a similar experimental setup as in the previous chapter, now applied to the two-dimensional case.

Figure 6.2 shows the relative regret for these experiments. Unlike earlier case studies, the hull methods do not start with low regret, as they are unable to construct a convex hull with few representative periods. In fact, they show more variability in regret, including cases where regret increases with more representatives, a pattern previously observed only for the clustering algorithms. The choice of distance metric also has a notable impact. The greedy convex hull method using cosine distance still performs best and is the first to reach a relative regret below 5%, while the same metric leads to higher regret for the other two methods compared to squared euclidean distance. However, with squared euclidean distance, the greedy convex hull method requires more than 85 representative periods to achieve a relative regret under 1%. The greedy bounded conical hull method shows similar behavior, but shows a sudden

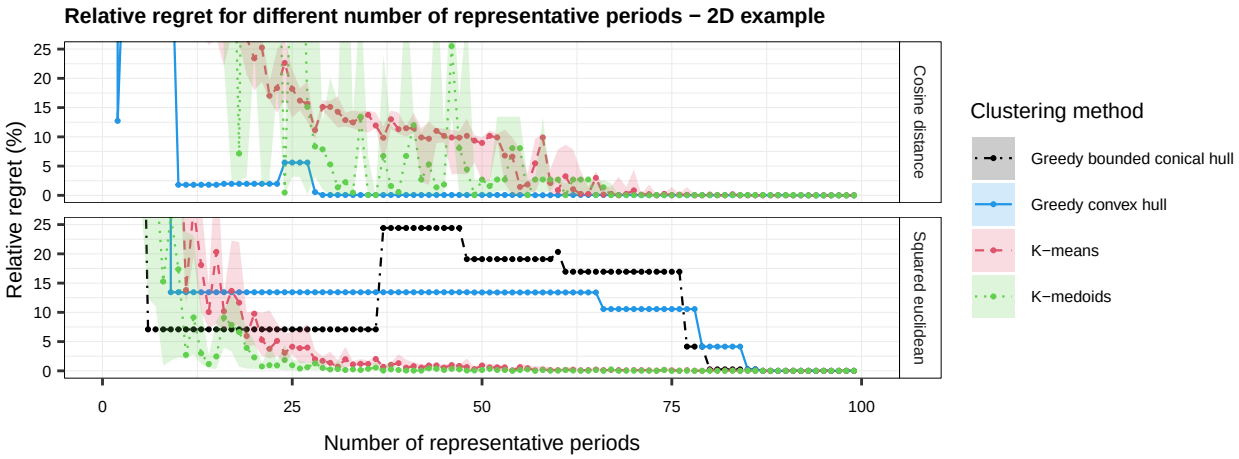

Figure 6.2: Relative regret (%) per number of representative periods for the 2-dimensional experiment, for both the cosine distance and squared euclidean distance. The different methods for selecting the representative periods are in different colors. The area for k-means and k-medoids represents the 25% and 75% quantiles.

increase in regret at 37 representatives. Its results are only shown for squared euclidean distance, since the method requires a metric where the distance to the origin is defined.

A look at the infeasibility of the solution provides us with even more interesting insights. We know beforehand that feasibility is not ensured by the hull methods, since an actual hull cannot be formed. Figure 6.3 shows the results on infeasibility for each number of representative periods, measured by the increase in the number of time steps with loss of load compared to the optimal solution. We see that for every combination of distance metric and representative period method, there are still time steps with loss of load up to at least 29 representative periods. At that number, the greedy convex hull method combined with cosine distance succeeds in achieving zero additional infeasibility compared to the optimal solution. What this experiment shows is that both hull methods, when a hull is not actually found, do not ensure feasibility at all. The feasibility and optimality highly depend on the actual representative periods chosen and their position within the dataset, where sudden spikes can be caused by selecting the wrong representative, as seen for the greedy bounded conical hull method with 37 representative periods.

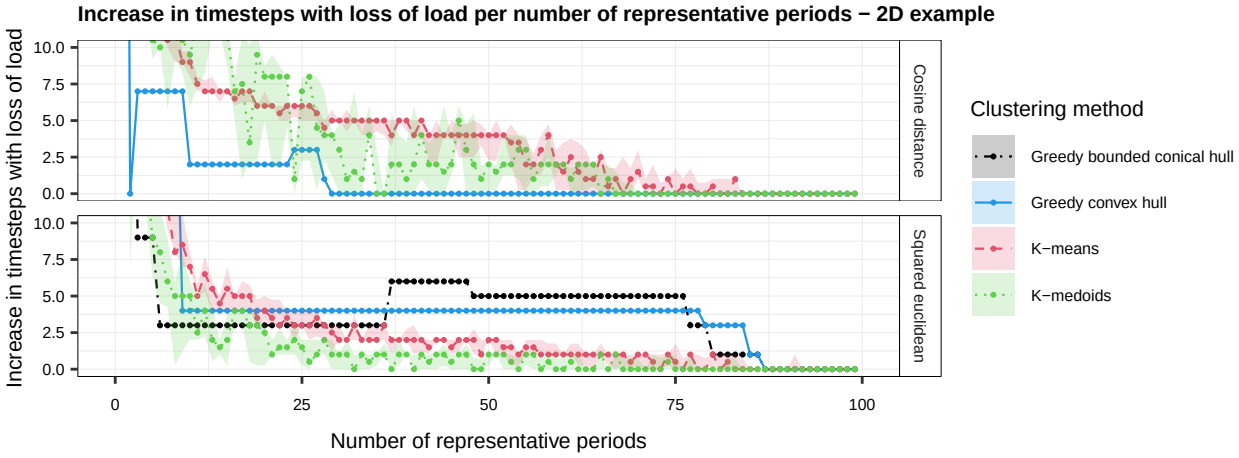

Figure 6.3: Increase in time steps containing loss of load per number of representative periods for the 2-dimensional experiment, for both the cosine distance and squared euclidean distance. The different methods for selecting the representative periods are in different colors. The area for k-means and k-medoids represents the 25% and 75% quantiles.

The only exception to all of this is that with just two representative periods, the greedy convex hull method in combination with the cosine distance metric has selected a combination of periods that leads to zero infeasibility elsewhere. Adding more representative periods leads to an increase in both infeasibility and costs. We hypothesize that one of the first two representatives has a very positive influence on the feasibility, and adding more representatives decreases this influence. We will investigate why this happens in Subsection 6.1.4, where we formalize why some representatives on the hull are more influential for feasibility than others in the context of the generation expansion model. But even with this exception, none of the methods can approach a stable low relative regret with a small number of representative periods, suggesting it is worth exploring whether we can already make some improvements in the two-dimensional space before moving to higher dimensions. This information can then be used in higher dimensions.

6.1.2. Bounding polytope for the convex hull

In two dimensions, it is easy to visualize a bounding polytope around the dataset using just three representatives, by imagining a large triangle around the samples. The tightest way to do this for our example is when the edges of the triangle each touch exactly one point on the ellipse. The initial ellipse is inscribed within the triangle. To find all representatives for which this is true, we consider an ellipse where each individual point has twice the distance to the center compared to the original ellipse. If we pick three equally distant points on this larger ellipse, they form a bounding polytope containing the hull of the samples. In theory, infinitely many combinations are possible, but we conducted experiments on 12 representative configurations, evenly spaced across the range. This ensures that each set differs enough to make a meaningful comparison in performance, while still providing a good overall coverage of the possibilities. The configurations are shown in Figure 6.4.

The bounding polytope corner points can be seen as artificially created periods, which we add to the set of representatives during clustering. Depending on the chosen configuration, the outcome can be very different. For example, triangle 3 optimizes for a point with high demand combined with low availability, most likely leading to high investments in the non-renewable generation technology. Thus, we can already imagine which configurations might lead to more conservative solutions than others.

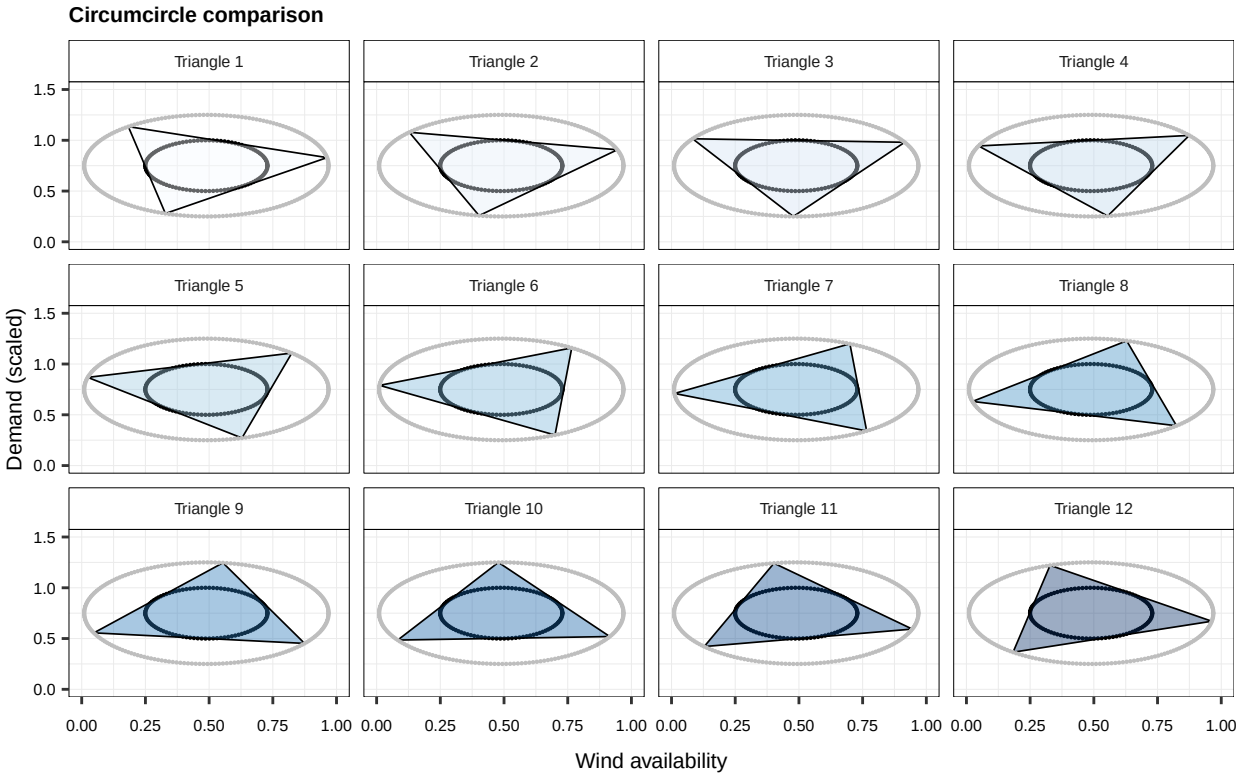

Figure 6.4: Options for three representatives tightly forming a bounding polytope for the original ellipse.

We first test the problem using only these three artificial representatives, with their weight in the reduced problem

equally distributed. This corresponds to the leftmost point in Figure 6.5 for each of the blue lines. As a reference, we also added the results for the original clustering methods without the bounding polytope representatives to the graph, so the k-means and k-medoids lines are equal to the ones previously shown in 6.2. Below we will look at the continuation of the line, but for now we first focus on the leftmost point. For three representatives, we already see that the relative regret is much lower than when using just k-medoids or k-means.

We also see that the relative regret differs largely per configuration. For triangle 10 we have a relative regret of 4%, while for the lighter lines the relative regret is higher than 25%. We can thus deduce that some of the artificial representatives result in more conservative investment decisions associated with higher costs, while others are more closely related to the investment decisions optimal for the original problem.

To further improve the results, we add more representative periods to the bounding polytope representatives, focusing on adding periods that improve optimality rather than feasibility. We do this because infeasibility in the original problem is already zero when the bounding polytope representatives are included in the solution, giving us the luxury to focus solely on optimality from this point onwards. To combine the triangle samples with extra representatives, we first find representatives with k-means and k-medoids, and then add the artificial triangle samples afterwards, assigning them a weight equal to the appropriate fraction of the total weights, depending on the number of representative periods used. The weights are then rescaled to ensure the same total weight in the objective as before. We do not combine the bounding polytope with the greedy convex hull method or bounded conical hull method, since we are sure that the representatives already contain the hull. Therefore, it would not make sense to look for additional hull points.

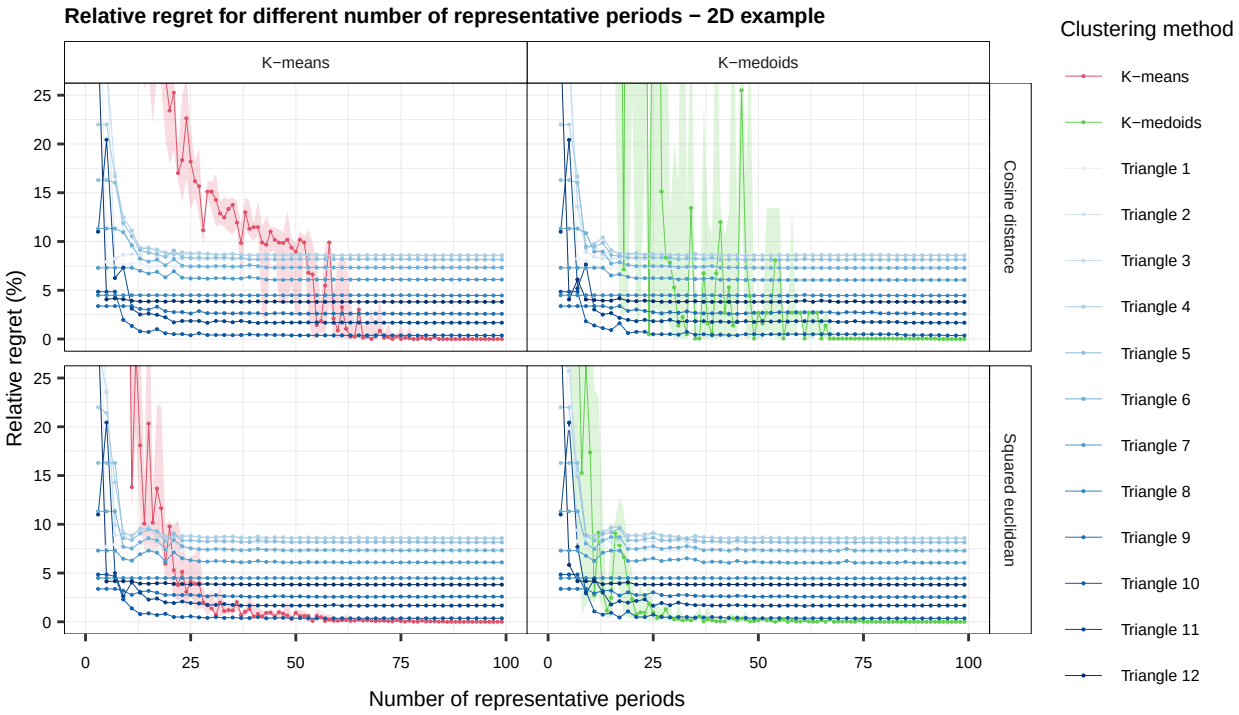

Figure 6.5: Relative regret (%) per number of representative periods for the 2-dimensional experiment, for both the cosine distance and squared euclidean distance. Different methods for selecting the representative periods are shown in different colors. The shaded areas represent the 25% and 75% quantiles across different random initializations for k-means and k-medoids.

As mentioned above, all number of representative periods lead to zero additional loss of load in the original problem, a great achievement. Adding the bounding polytope points thus eliminated all infeasibility, as expected based on the earlier proofs for the hull in Chapter 4. The expected downside is that the relative regret might not be able to decrease to 0%, even when using a large number of representative periods. The artificially added periods that ensure feasibility also cause the decision variables to take on more extreme values than would be necessary for the full problem. This

results in solutions that might be too conservative.

This is indeed reflected in Figure 6.5. For all triangle configurations, even with additional representative periods, a lower bound on relative regret is reached. When using a larger number of representative periods, the original k-means and k-medoids methods have better performance in terms of relative regret. For a low number of representative periods, however, there are clear advantages to using the bounding polytope. The relative regret is mostly below the original methods when fewer than 15 representative periods are used.

As expected, we observe significant differences between the configurations. The best configuration seems to be triangle 10, which consistently shows the lowest relative regret of all configurations. It is followed by triangles 11 and 9. Especially for triangle 10, the relative regret is very low from the beginning onward. When using a low number of representative periods, below 25 for example, this configuration would give the best results over all proposed methods. This is very important, as in practice we would like to find the method that is best with a low number of representative periods, as then the best speedup can be achieved.

In practice, we would not be able to inspect all configurations before selecting the best. Therefore, we want to understand why some perform better than others. Looking at the best performing configurations in Figure 6.4, they appear to correspond to an extreme point at the top of the ellipse. We will explore a potential reason on why these configurations perform best in Subsection 6.1.4. Second, we want to deal with the conservativeness of this approach. Adjusting the weights in the objective function might overcome these conservative solutions. This way, the extreme points can gradually lose weight when better representatives for the actual periods are found. This will be investigated in the next subsection.

6.1.3. Incorporating blended weights

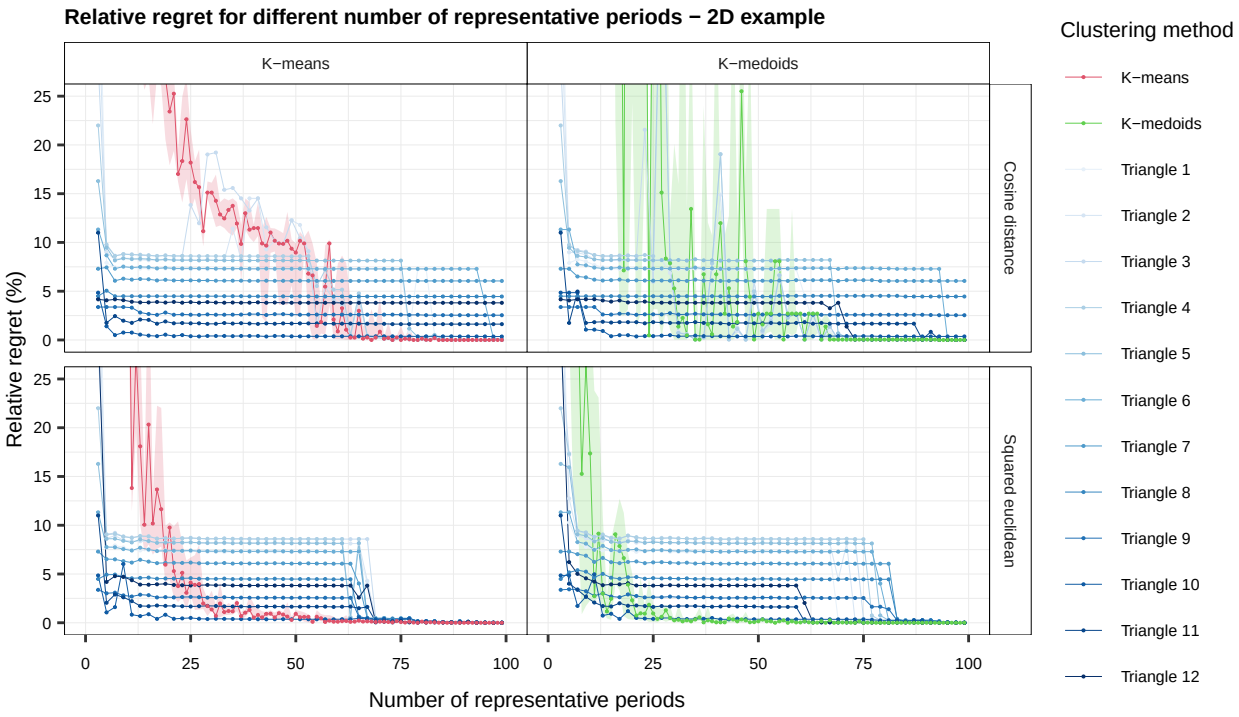

Figure 6.6: Relative regret (%) per number of representative periods for the 2-dimensional experiment, for both the cosine distance and squared euclidean distance. The different methods for selecting the representative periods are in different colors. The areas represent the 25% and 75% quantiles based on different random initializations for k-means and k-medoids.

The zero infeasibility achieved by using the bounding polytope in the previous subsection is very valuable. However, when more representative periods are used, the relative regret becomes higher than when the bounding polytope

is not used as method. This can be seen in Figure 6.5, for example in the combination of k-medoids and squared euclidean distance after 25 representative periods.

The bounding polytope representatives were added after selecting the other periods, and their weights were scaled to have the average weight of one representative. We expect that if these weights could decrease more when there are enough other representatives that are closer to the original periods, the influence of the artificial points would slowly fade. This could lead to a lower relative regret when more representative periods are used. The blended weights approach, explained earlier in Subsection 3.3.4, allows for this kind of dynamic behavior. To make this work, the artificial representatives need to be added before selecting the other representatives instead of after. This way, we can build the correct initial weight matrices and enable blended weights.

We run similar experiments for the blended weights approach as before, combining the different bounding polytope configurations with both k-means and k-medoids. Figure 6.6 shows the relative regret from those experiments, where each blue line represents a different configuration. With blended weights, the bounding polytope approach succeeds to achieve a relative regret close to 0\$. However, for a small number of representative periods, the results are similar to what we saw in Figure 6.5, with no real improvement in relative regret. The effect of the blended weights is only visible once we use more than 60 representative periods, which makes it less useful in this particular example. At that point, both k-means and k-medoids without artificial representatives already have a lower regret than the bounding polytope plus blended weights combinations. So, blending weights does decrease the relative regret, but can be improved to affect lower number of representatives as well.

6.1.4. Worst-case artificial representatives relevant for generation expansion planning

In Figure 6.3, we observe that there was a sharp improvement when using two representative periods for the greedy convex hull method with the cosine distance metric, even resulting in zero infeasibility. Surprisingly, adding more periods can increase both infeasibility and cost, as also seen in Figure 6.2. Such a sharp improvement appears again when using 10 and 29 representative periods for this particular method combination. To better visualize which points contribute to this positive effect, we highlight them in Figure 6.7. All the 'important' representative periods are located in the upper-left quadrant of the ellipse. As we explain below, this is not a coincidence.

In the generation expansion planning (GEP) model described in Section 3.1, we assume that curtailment of renewable energy is always possible. This means that even if more energy is available than needed, overproduction can be avoided. From this, we can reason that any period with similar demand but higher availability than a chosen representative period will not contain infeasibility. This suggests that, rather than constructing a convex or bounded conical hull over all points, we might only need to consider periods with lower availability. However, as shown in Figure 6.7, the " important" representatives are not just on the left side of the circle with a low availability, but specifically in the upper-left quadrant. This is due to the higher demand associated with those points. To have a robust investment decision the representative set should include one representative with a relative high demand for the system, especially if that demand is associated with low availability in renewable energy sources. Taking this together, for feasibility in the original problem we would like to find a representative in the reduced problem that has a low availability-to-demand ratio, combined with a high demand.

To explain why these features are relevant, we first prove, in two dimensions, which points are guaranteed to have no infeasibility when a single representative is selected. We will later generalize this result to higher-dimensional data. To start, we present the formal definition of the simplified two-dimensional model used before. Since only one location is considered, we remove the location index and constraints related to energy flow. Each period includes only one hour, so the index set $\mathcal{I}$ contains combinations of scenarios and periods, but not hours. We also assume that only one generation technology, denoted g_1, has uncertain availability. For all $g \in \mathcal{G} \setminus g_1$ and all $i \in \mathcal{I}$, we set $A_{g,i} = 1$. This leads to the following original model:

$$\min \quad c^{\text{inv}} + AF \cdot c^{\text{op}} \tag{6.1a}$$

$$\text{s.t.} \quad c^{\text{inv}} = \sum_{g \in \mathcal{G}} I_g \cdot U_g \cdot i_g \tag{6.1b}$$

$$c^{\text{op}} = \sum_{i \in \mathcal{I}} \left(\sum_{g) \in \mathcal{G}} V_g \cdot prod_{g,i} + V^{\text{loss}} \cdot l_i \right) \tag{6.1c}$$

$$\sum_{g \in \mathcal{G}} prod_{g,i} + l_i = D_i \qquad \forall i \in \mathcal{I} \tag{6.1d}$$

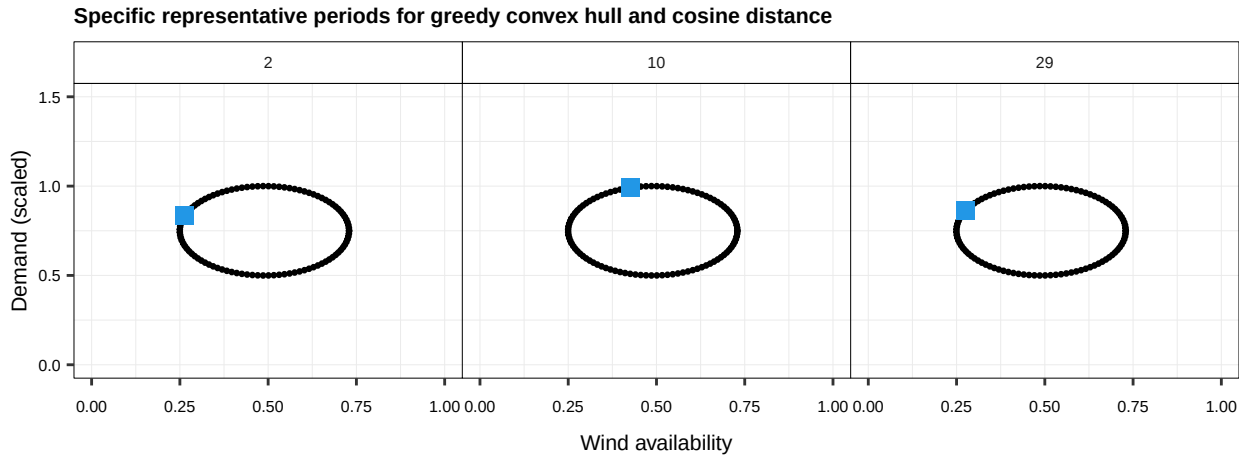

Figure 6.7: The three specific representative periods that lead to large improvements in optimality and feasibility, when using the greedy convex hull method in combination with the cosine distance metric on the 2-dimensional example.

$$prod_{g,i} \leq A_{g,i} \cdot i_g \cdot U_g \qquad \forall g \in \mathcal{G}, i \in \mathcal{I} \tag{6.1e}$$

The reduced model becomes:

$$\min \quad c^{\text{inv}} + AF \cdot c^{\text{op}} \tag{6.2a}$$

$$\text{s.t.} \quad c^{\text{inv}} = \sum_{g \in \mathcal{G}} I_g \cdot U_g \cdot i_g \tag{6.2b}$$

$$c^{\text{op}} = \sum_{r \in \mathcal{R}} W_r \left(\sum_{g \in \mathcal{G}} V_g \cdot prod_{g,r} + V^{\text{loss}} \cdot l_r \right) \tag{6.2c}$$

$$\sum_{g \in \mathcal{G}} prod_{g,r} + l_r = D_r \qquad \forall r \in \mathcal{R} \tag{6.2d}$$

$$prod_{g,r} \leq A_{g,r} \cdot i_g \cdot U_g \qquad \forall g \in \mathcal{G}, r \in \mathcal{R} \tag{6.2e}$$

Assume we have a representative r in the reduced model that does not lead to any loss of load. We will examine which periods i this representative alone can ensure feasibility for, when it is added to the full set of representatives $\mathcal{R}$. By analyzing this, we gain insight into why certain representatives contribute more to overall feasibility, making them more valuable to include in the model.

Just as in the normalization step, we assume that if a location has a demand at some point in time higher than zero, it will have a demand higher than 0 all times. If there are locations that only produce energy and do not have a demand, they should be merged with another node to be taken into account.

Theorem 6.1. *Consider an instance of the two-dimensional GEP model as presented in* (6.1)*. We use a representative set $\mathcal{R}$ for the original period set $\mathcal{I}$. Solve the reduced model with this representative set to optimality, for which the model in* (6.2) *is used. Denote the optimal values of the decision variables as $i_g^\star$, $l_r^\star$ and $prod_{g,r}^\star$.*

Let $r \in \mathcal{R}$. Assume that $l_r^\star = 0$. Then for all $i \in \mathcal{I}$ for which (6.3) *and* (6.4) *hold, we can ensure that there exist a solution to the original model with the optimal investment decisions $i_g^\star$ of the reduced model, which ensures that $l_i = 0$.*

$$\frac{A_{g_1,i}}{D_i} \geq \frac{A_{g_1,r}}{D_r}. \tag{6.3}$$

$$D_i \leq D_r. \tag{6.4}$$

The decision variables that will satisfy this for i are defined as:

$$prod_{g,i} = \frac{D_i}{D_r} \cdot prod^{\star}_{g,r} \qquad \forall g \in \mathcal{G}. \tag{6.5}$$

Proof. We show that when a representative is included in the reduced model, all original periods that satisfy (6.3) and (6.4) indeed satisfy the constraints in the original model with zero loss of load with the proposed solution.

As stated in the theorem, $l^{\star}_r = 0$. We can thus remove the loss of load variable in the demand constraint met by the reduced model, like in (6.6). Moreover, we split the availability constraints of the reduced model into (6.7) and (6.8) to highlight the difference between the uncertain generator g_1 and the other generation technologies:

$$\sum_{g \in \mathcal{G}} prod^{\star}_{g,r} = D_r, \tag{6.6}$$

$$prod^{\star}_{g,r} \leq i^{\star}_g \cdot U_g \qquad \forall g \in \mathcal{G} \backslash \{g_1\}, \tag{6.7}$$

$$prod^{\star}_{g_1,r} \leq A_{g_1,r} \cdot i^{\star}_{g_1} \cdot U_{g_1}. \tag{6.8}$$

Now we show that when using the optimal investment decision variables i_g from the reduced model, the production decision variables defined by (6.5) will indeed provide a feasible solution with no loss of load.

First we show that the demand constraint still holds for all i, with no infeasibility occurring:

$$\begin{aligned} \sum_{g \in \mathcal{G}} prod_{g,i} &= \sum_{g \in \mathcal{G}} \frac{D_i}{D_r} prod^{\star}_{g,r} \\ &= \frac{D_i}{D_r} \sum_{g \in \mathcal{G}} prod^{\star}_{g,r} \\ &= \frac{D_i}{D_r} D_r \\ &= D_i. \end{aligned} \tag{6.9}$$

This is fully based on the how we defined the production values to be in (6.5) and the demand constraint in the reduced model given by (6.6).

Now, we show that the availability constraint is satisfied under this solution. We do this first for uncertain generation technology g_1. The following holds:

$$\begin{aligned} prod_{g_1,i} &= \frac{D_i}{D_r} prod^{\star}_{g_1,r} \\ &\leq \frac{D_i}{D_r} \cdot A_{g_1,r} \cdot i^{\star}_{g_1} \cdot U_{g_1} \\ &= \frac{A_{g_1,r}}{D_r} \cdot D_i \cdot i^{\star}_{g_1} \cdot U_{g_1} \\ &\leq \frac{A_{g_1,i}}{D_i} \cdot D_i \cdot i^{\star}_{g_1} \cdot U_{g_1} \\ &= A_{g_1,i} \cdot i^{\star}_{g_1} \cdot U_{g_1}. \end{aligned} \tag{6.10}$$

This shows that the availability constraint is satisfied for the uncertain generation technology g_1. We used the condition (6.3) from the theorem. Now we show the same for the certain generation technologies. Let $g \in \mathcal{G} \backslash \{g_1\}$, then:

$$\begin{aligned} prod_{g,i} &= \frac{D_i}{D_r} prod^{\star}_{g,r} \\ &\leq \frac{D_i}{D_r} \cdot i^{\star}_g \cdot U_g \\ &\leq i^{\star}_g \cdot U_g. \end{aligned} \tag{6.11}$$

For the last step, we use that the demand value of the representative was constrained to be higher than the demand value of the periods it could represent, as mentioned in (6.4).

We can conclude that the constraints hold for i in the original model, concluding the proof. □

The theorem can be interpreted in two ways. First, it explains why the representative periods in the upper-left quadrant of the ellipse lead to the sudden drops in costs and infeasibility: they combine a low availability-to-demand ratio with high demand, which, based on the proof, means coverage of a larger set of original periods. Second, it also shows that choosing one representative period can ensure feasibility for all periods in $\mathcal{I}$ in the original model, namely the (artificial) representative period that satisfies:

$$D_r = \max_{i \in \mathcal{I}} D_i \tag{6.12}$$

$$\frac{A_{g_1,r}}{D_r} = \min_{i \in \mathcal{I}} \frac{A_{g_1,i}}{D_i} \tag{6.13}$$

Including only this point in the reduced model would guarantee feasibility. Therefore, in two-dimensional examples, this representative could replace the bounding polytope or a convex or bounded conical hull. Other representative periods could then be found to improve optimality.

This is a very important implication, and it can be extended to higher dimensions and multiple scenarios. Only when we add ramping constraints or transmission lines, we can not apply the results. Nevertheless, we can easily extend it to cases with more than one uncertain generation technology and locations, as long as we do not allow any transmission between locations. We do that below.

Corollary 6.2. *Consider an instance of the GEP model as presented in* (6.1)*, but with multiple locations $n \in \mathcal{N}$, time steps within periods $h \in \mathcal{H}$ and multiple uncertain generation technologies. We define the total set of uncertain generation technologies to be $\mathcal{G}^U \subset \mathcal{G}$ and the combination of uncertain generation technologies and locations to be $\mathcal{NG}^U \subset \mathcal{N} \times \mathcal{G}^U$. We use a representative set $\mathcal{R}$ for the original period set $\mathcal{J}$. Solve the reduced model with this representative set to optimality, for which the model in* (6.2) *is used, again with multiple locations and time steps within periods. Denote the optimal values of the decision variables as $i^\star_{n,g}$, $l^\star_{n,(r,h)}$ and $prod^\star_{n,g,(r,h)}$.*

Let $r \in \mathcal{R}$. Assume that $l^\star_{n,(r,h)} = 0$ for all $n \in \mathcal{N}, h \in \mathcal{H}$. Then for all $j = (s,p) \in \mathcal{J}$ for which (6.14) *and* (6.15) *hold, we can ensure that there exist a solution to the original model with the optimal investment decisions $i^\star_{n,g}$ of the reduced model, which ensures that $l_{n,(j,h)} = 0$, for all $n \in \mathcal{N}$ and $h \in \mathcal{H}$.*

$$\frac{A_{n,g,(j,h)}}{D_{n,(j,h)}} \geq \frac{A_{n,g,(r,h)}}{D_{n,(r,h)}} \qquad \forall h \in \mathcal{H}, (n,g) \in \mathcal{NG}^U \tag{6.14}$$

$$D_{n,(j,h)} \leq D_{n,(r,h)} \qquad \forall n \in \mathcal{N}, h \in \mathcal{H} \tag{6.15}$$

The decision variables that will satisfy this for j are defined as:

$$prod_{n,g,(j,h)} = \frac{D_{n,(j,h)}}{D_{n,(r,h)}} \cdot prod^\star_{n,g,(r,h)} \qquad \forall h \in \mathcal{H}, (n,g) \in \mathcal{NG} \tag{6.16}$$

Proof sketch. The proof follows the exact reasoning of the proof for Theorem 6.1, while using $(n,g) \in \mathcal{NG}^U$ where g_1 was used. Also, every step is the case for all $h \in \mathcal{H}$ and $n \in \mathcal{N}$. □

Given these theorems, we can better explain the results that we observed previously in this chapter. We first discuss why the second representative period when using the greedy convex hull method and cosine distance on the two-dimensional case, already led to zero infeasibility. In Figure 6.7 we can see that it looks like it is the period which satisfied the minimum ratio condition in (6.13). However, the demand in that representative is not equal to the maximum demand. This is because, in the solution of that reduced model, investment was made solely in the uncertain generation technology. Since the constraint on having the maximum demand was not necessary for uncertain technologies, the condition in (6.12) was not required.

We also would like to understand why some bounding polytope configurations performed better than others. However, the previous theorem only explains feasibility and says nothing about optimality. Since all triangle configurations satisfied feasibility, the insights from the theorem do not help. However, we do see that the specific bounding

polytope whose points had no availability-to-demand ratio lower than necessary, triangle 11, was among the best-performing configurations. Its regret was second lowest, with triangle 10 being the configuration with the lowest relative regret (See Figure 6.4 for the configurations). So while the theorem gives a nice direction for points that we should consider as (artificial) representatives, it is not a panacea: choosing a representative period combination in which some representatives have a lower than necessary ratio might still result in a solution with lower relative regret.

To conclude, selecting a worst-case point for feasibility and complementing it with other points for optimality is an effective and theoretically promising approach. However, the theoretical guarantees on feasibility presented in this chapter do not trivially extend to a case including transmission and ramping. Nevertheless, we want to see whether the improvements observed in two dimensions are still reproducible in higher dimensions.

6.2. Identifying worst-case artificial representatives in a higher-dimensional space

In two dimensions, we see that constructing a bounded conical or convex hull is not always possible with a limited number of representative periods, such as in the case of the ellipse. Moreover, it could be that the greedy implementation does not necessarily select the correct hull points due to its approximate nature. This leads to two hypotheses for why the greedy convex hull and greedy bounded conical hull methods performed less effectively in the final case study presented in Chapter 5.

In two dimensions, adding artificial extremes to construct a bounding polytope for the hull as the first set of representatives showed potential: infeasibility was immediately eliminated, and depending on the configuration, the results when including the artificial representatives were better than just using k-means and k-medoids. Since forming a bounding polytope, as done in the previous section, would be intractable in high dimensions, with points needed growing exponentially, we focus on incorporating what we learned while adjusting for high dimensions.

We propose creating one worst-case artificial representative to try to ensure feasibility with. Although Theorem 6.1 did not trivially extends to the high-dimensional case with ramping and transmission lines, we believe that it can still be interesting to test its effect. To this end, we propose adding one worst-case representative period that covers all minimum availability-to-demand ratios and maximum demand. We first find the maximum demand for each time step and each location, and then, for each uncertain generation technology, choose availability such that the availability-to-demand ratio is equal to the minimum for this generation technology. We do this for each time step and location individually, thus creating essentially a worst-case period.

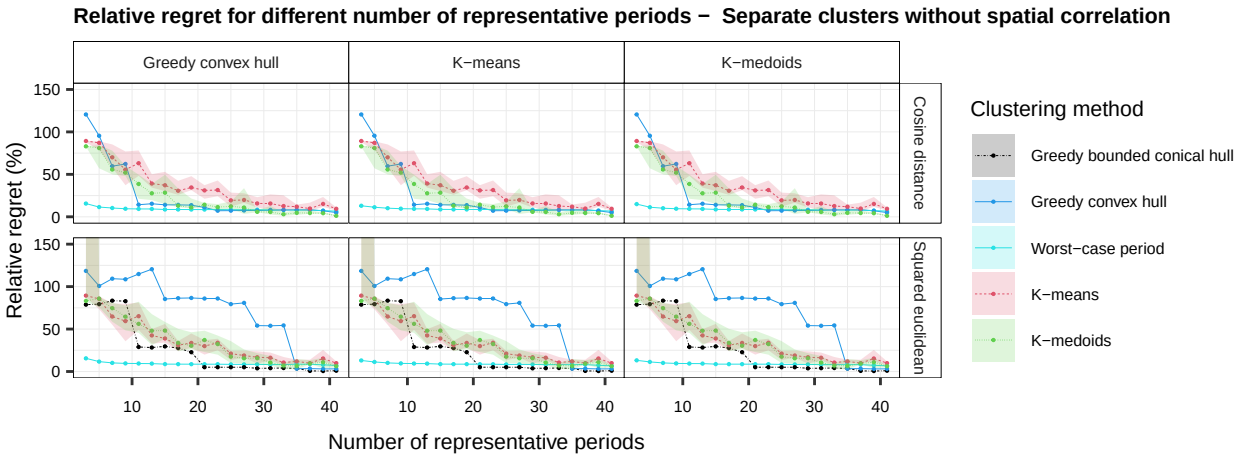

Figure 6.8: Relative regret (%) per number of representative periods for the “Separate clusters without spatial correlation” case study, for both the cosine distance and squared euclidean distance. The different methods for selecting the representative periods are in different colors. The “Worst-case period” method is combined with either greedy convex hull, k-means or k-medoids. The area for k-means and k-medoids represents the 25% and 75% quantiles

We test the proposed method on the previously discussed case study: separate clusters without spatial correlation. For this case study, we were not able to achieve a successful result for each one of the methods, where “successful” is defined as achieving a low relative regret when using a low number of representative periods. We combine the

artificial worst-case representative with k-means and k-medoids, just like was done for the bounding polytope in the previous section. However, we also combine the artificial worst-case representative with the convex hull method. Earlier, we did not opt for this as the bounding polotype was guaranteed to already contain the hull. Now, a hull is not necessarily contained by the artificial representative, and feasibility is also not necessarily ensured. It therefore makes sense to continue looking for a convex hull.

Figure 6.8 shows the relative regret when using this artificial worst-case period. In the graphs, we also show the lines representing the results of the experiments in the previous chapter, in which the original four methods were tested on these experiments. The "Worst-case period" line shows the combination of this artificial representative period with each of the three other clustering methods mentioned. In general, we see that adding this one worst-case period has a very positive influence. We obtain a relative regret below 10% with only five representative periods. It is interesting to see that the differences between combining this artificial representative with the greedy convex hull method, k-means, or k-medoids are minimal. This indicates that the results are largely influenced by the artificial representative.

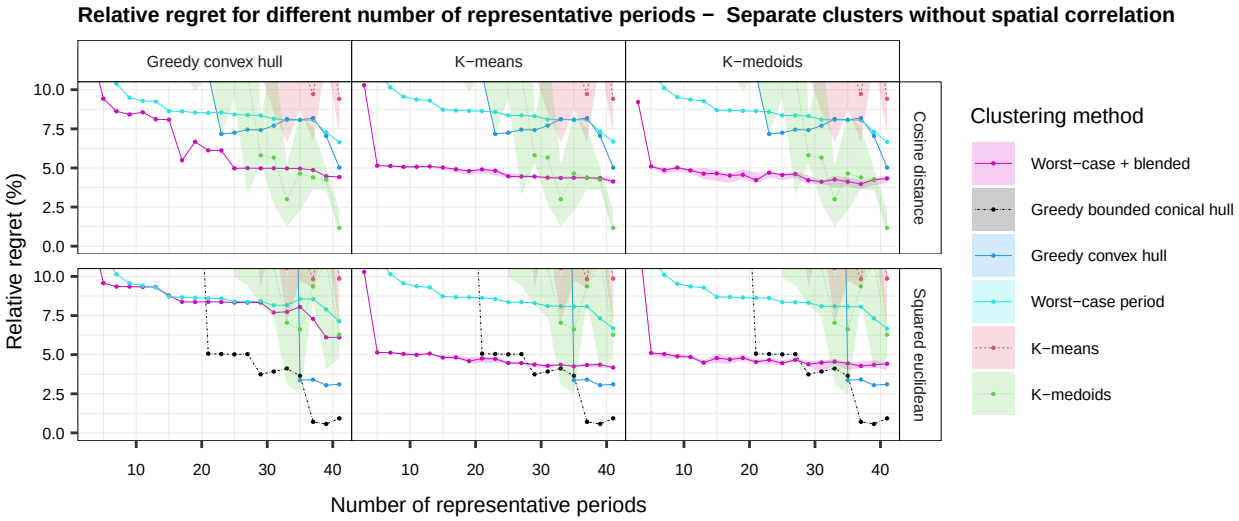

Figure 6.9: Relative regret (%) per number of representative periods for the "Separate clusters without spatial correlation" case study, for both the cosine distance and squared euclidean distance. The different methods for selecting the representative periods are in different colors. The "Worst-case period" method is combined with either greedy convex hull, k-means or k-medoids. The area for k-means and k-medoids represents the 25% and 75% quantiles.

Just as with the bounding polytope before, at some point, the solution with the artificial worst-case period cannot achieve a lower regret, while the other methods will be able to do so when a high number of representative periods are used. We see this for the greedy bounded conical hull, which achieves lower regret when more than 21 representative periods are used. We once more try the blended weights mentioned before to slowly decrease the influence of the worst-case period. The results of this can be seen in Figure 6.9. The graph is zoomed in to highlight differences between the methods.

We see a clear advantage when using blended weights in combination with k-means or k-medoids clustering. This is interesting, as the difference was only minor when weights were not blended. It could be that, since k-means and k-medoids both focus on finding centers, the periods that get more weight when using the blended weights are more closely located to a majority of the periods. Therefore, they are better for optimality. For the greedy convex hull method, although less weight is assigned to the one worst-case artificial representative when using the blended approach, the weight is still assigned to representatives that are more on the extreme side of the period set. We can conclude that for a small number of representative periods, using blended weights can significantly reduce relative regret. However, even when many representative periods are available, the solutions that include the worst-case period remain too conservative, resulting in a higher relative regret compared to, for example, the greedy bounded conical hull method.

In addition to optimality, we also consider the increase in time steps with loss of load in Figure 6.10. Here, we observe

a clear difference between the methods that include the worst-case period and the original methods. When the worst-case representative is added, there is no additional loss of load anywhere, while for the greedy bounded conical hull, despite having the lowest regret when using more than 30 representative periods, the number of time steps with loss of load remains above zero throughout. So when evaluating performance, it is important to consider that avoiding any loss of load is also a critical aspect of system reliability.

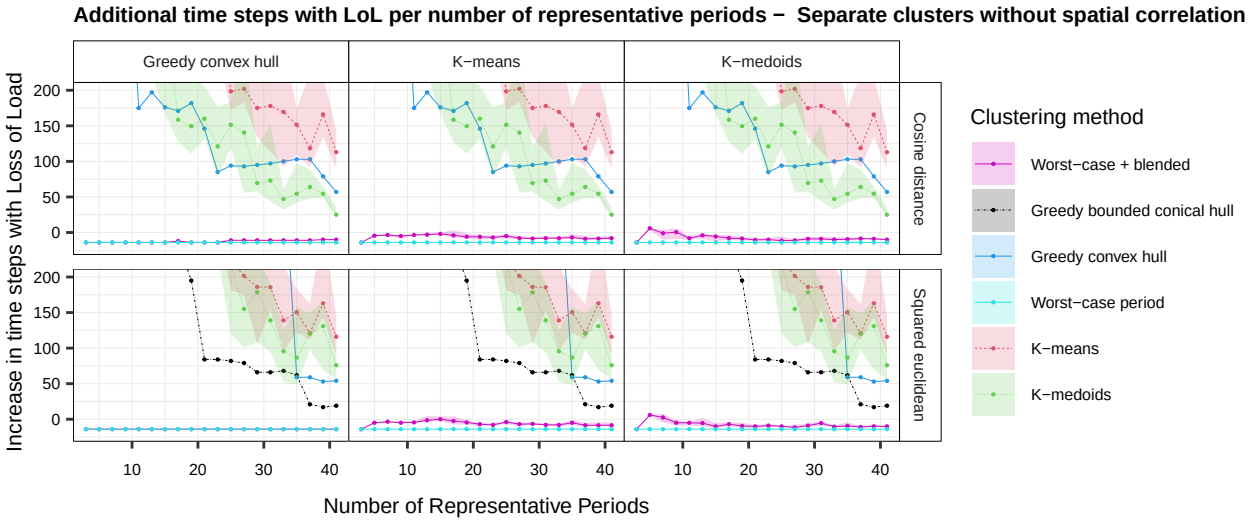

Figure 6.10: Increase in time steps containing loss of load per number of representative periods for the "Separate clusters without spatial correlation" case study, for both the cosine distance and squared euclidean distance. The different methods for selecting the representative periods are in different colors. The "Worst-case period" method is combined with either greedy convex hull, k-means or k-medoids. The area for k-means and k-medoids represents the 25% and 75% quantiles.

6.3. Conclusion

This chapter demonstrated that even in simple two-dimensional settings, constructing a convex or bounded conical hull from original data points can be infeasible with a limited number of representative periods. To address this, we introduced artificial representatives that form a bounding polytope enclosing the convex hull of the data. This guarantees feasibility with only a limited number of representatives and, in those cases, also achieves a lower relative regret than alternative methods. While effective in two dimensions, the approach scales poorly to higher dimensions due to the exponential increase in required points and computational complexity.

In the context of Generation Expansion Planning (GEP), we can reason more structurally about which periods are critical. In two dimensions, we showed that a single worst-case representative, with maximum demand and a minimum availability-to-demand ratio, can ensure feasibility for all original periods under simplified assumptions. Extending this to higher dimensions, we proposed generating a worst-case artificial representative period that captures these extreme conditions. While this no longer guarantees feasibility due to added constraints like ramping and transmission, results on the case study from the previous chapter show that this worst-case period reduces infeasibility and regret. Blended weights help reduce conservativeness as more representatives are added. The experiments also support a hybrid approach: using an artificial worst-case period to ensure feasibility, then complementing it with additional periods aimed at improving optimality. This, in combination with the blended weights approach, proved promising.

In conclusion, artificial representatives are a valuable tool for improving feasibility when few representative periods can be used. For practical applications, we recommend combining artificial worst-case representatives with blended weights and supplementing them with periods selected for optimality. In the next chapter, we will apply this method to a larger, more realistic case study.

7

European case study

Up to this point, we have explored both the theoretical results of the two hull methods and their performance in small-scale case studies. We found that when a hull is identified, these methods guarantee feasibility and often lead to a very positive effect on relative regret compared to alternative approaches. Although identifying such a hull can be challenging in data with high variability, this challenge was successfully addressed by introducing artificial representatives with extreme, worst-case values, ensuring feasibility. This approach performed well in the final case study of the previous chapter. In this chapter, we extend that analysis to a case study with the size and complexity of real-world applications. The European case study we use features a high share of renewables and reflects the variability common in energy systems. It is adapted from a deterministic model by Gao et al. [11], with additional variability introduced through scenarios based on climate and demand data from the 2022 TYNDP Scenarios [7]. This setup enables us to evaluate the feasibility and optimality of all previously proposed methods, as well as their impact on processing time. This is particularly relevant, as the full model cannot be solved on a standard laptop within reasonable time and memory constraints. In Section 7.1, we first explain how the case study is adapted and which assumptions are made. Section 7.2 then outlines the experimental setup in more detail. The experiments include the clustering methods, hull methods, the addition of worst-case representatives, and the use of blended weights. The results are presented and discussed in Section 7.3.

7.1. Case study

We define several criteria for selecting a suitable case study to test the proposed methods on. First, the case study should feature a high share of renewables in the generation mix, as such systems are most sensitive to temporal detail. Second, since scenario development is a complex field on its own, we aim to build on existing scenario data rather than constructing entirely new ones. Third, we select a case that does not rely on storage options, as this thesis does not focus on inter-period constraints. We do motivate future work to incorporate this, as storage is very relevant in current energy systems. Fourth, the case study should be large enough to reflect real-world complexity and computational limitations, particularly for stochastic programming, where the full problem size becomes infeasible to solve on a standard laptop.

The data we used was made publicly available by Gao [10], and it was earlier obtained from the 2022 TYNDP Scenarios [7]. It is well-suited to our purpose, as it focuses entirely on renewable generation, making it highly sensitive to temporal resolution. Moreover, its size and complexity reflect real-world case studies. While the original implementation is deterministic and relies only on data from the 2008 climate year, it is based on a dataset that includes 35 years of availability and demand data, enabling the construction of stochastic scenarios. In this chapter, we make use of this broader dataset to generate multiple scenarios for the model, allowing us to evaluate the performance of the proposed methods under a stochastic programming setting. The full dataset is too large to solve directly with stochastic programming, which further motivates the use of temporal reduction techniques.

To meet the needs of this thesis, several adjustments to the original case study are necessary. We begin by describing the process of adding scenario data. The original case study includes hourly data per location for both electricity demand and the availability of renewable energy sources (onshore wind, offshore wind, and solar PV). This data is

based on the climate year 2008, with demand projections taken from ENTSO-E's 2030 national trends and climate data from the Pan-European Climate Database (PECD) [7]. Since both the national trends and the PECD also provide data for multiple climate years, we extend the case study to include all available years from 1982 to 2016. This results in 35 complete scenarios, each with a full year of hourly input data. A few caveats must be addressed when extending the dataset.

1. The original case study includes 30 European countries, which is fewer than those available in the PECD. We therefore restrict the scenario data to the countries present in the original dataset. Those are all the EU countries (27 in total) and the United Kingdom, Norway and Switzerland.
2. Some countries are split into multiple regions in the ENTSO-E demand scenario data. For France, Greece, and the United Kingdom, we use only the region marked "00", following the approach used in the original case study. For Denmark, Italy, Luxembourg, Norway, and Sweden, we aggregate demand across regions, again following the original setup.
3. Availability values per generation technology in the PECD scenarios are provided per region, with varying spatial resolution across countries. The zones used here are even more detailed than before. Except for the countries mentioned next, we use the region marked "00". For Italy, Norway, and Sweden, we average across multiple regions, consistent with the original case study processing.
4. The PECD includes more technology-location combinations than those permitted in the original case study. We filter the dataset accordingly and include only the combinations allowed in the original setup.
5. While the original case study included Malta, we exclude it from our dataset. Under the given assumptions and transmission capacities, Malta would experience loss of load in nearly every period across all scenarios. This is because in the case study it is only allowed to produce solar energy, which is unavailable at night, and imports from other countries are insufficient to meet demand. Including Malta would therefore distort the loss of load metric, which we use to evaluate the quality of our reductions. By excluding it, we hope to ensure that any observed loss of load reflects the performance of the reduction methods rather than structural infeasibilities in the model setup.

With these adjustments, we construct a consistent set of 35 scenarios with a full year of hourly data for demand and renewable availability across the original set of countries and technologies. This extended dataset forms the basis for evaluating the proposed reduction methods under realistic temporal and spatial variability.

In terms of technology options, the original case study allows new investments exclusively in renewable generation. Storage and hydropower are included in the original dataset, but we exclude them from our model. This decision reflects limitations discussed earlier: modeling storage properly requires handling intertemporal constraints across representative periods, such as the techniques proposed by Tejada-Arango et al. [35], which fall outside the scope of this thesis. Nuclear power is included in the original case at its existing capacity, without the option for expansion. Our original model did not account for fixed capacities or investment bounds, so we adapt it by setting the investment cost of nuclear to zero, ensuring full installation of existing capacity, and by limiting the investment variable $i_{n,g}$ at the current installed capacity to prevent additional investment. For other technology-location combinations, the original case study also imposes investment limits. However, after removing hydropower, these limits become too restrictive, often leading to infeasible or unrealistic system configurations. We therefore relax these bounds to maintain model feasibility while preserving the general structure of the original setup.

No ramping data is available in the original dataset. To introduce some realism without significantly increasing model complexity, we adopt ramping parameters from the PyPSA model [3], applying a ramping limit of 0.3 of the total capacity exclusively to nuclear power. Finally, other parameters, such as transmission capacity, variable costs, and investment costs, are taken directly from the original case study without modification. We also retain the original value of lost load, set at one thousand euros per megawatt. This is relatively low compared to estimates proposed by, for example, the Netherlands Authority for Consumers and Markets [22], which suggests a value more than 60 times higher.

7.2. Experimental setup

As mentioned earlier, the case study provides 35 full-year scenarios of hourly data. For our experiments, we randomly select ten scenarios for in-sample use, meaning they are used in the process of selecting representative periods. An additional nine scenarios are selected for out-of-sample evaluation, where the quality of the selected representatives is tested on unseen data. This split allows us to assess both how well the reduction performs during training and how

robust it is to new conditions. Using all remaining scenarios for out-of-sample evaluation was not feasible due to the time limits of this thesis. Below, we first describe the setup used to generate the different reduced models, followed by the setup used to evaluate their performance.

7.2.1. Reduced model

Even with representative periods, the size and complexity of the case study result in relatively long computation times. Moreover, multiple parameters influence the outcome, and some methods involve randomness, such as k-means and k-medoids, requiring repeated runs to obtain more reliable estimates. To balance comprehensiveness with computational feasibility, we fix several components and vary only the most relevant ones. Fixed are the normalization method and the length of one period as 24 hours, as described in Chapter 3. We also fix the distance metric. Previously, we experimented with both cosine distance and squared euclidean distance. However, the differences in performance were not consistent. Moreover, cosine distance offers less graphical interpretability and is not compatible with one of the methods (greedy bounded conical hull), leading us to focus on the more commonly used squared euclidean distance.

The goal of the experiments is two-fold: first, we want to understand the differences in how representative periods are handled, either per-scenario or cross-scenario. As described earlier in this thesis, the per-scenario approach selects representative periods within each scenario, while the cross-scenario approach treats the entire set of periods as one and performs the selection over that combined set. We aim to assess whether one of these approaches offers advantages, in terms of computational time and accuracy.

In addition to comparing these approaches, the main focus of the experiments is to evaluate the performance of different selection methods. These include the four techniques discussed earlier: k-means, k-medoids, greedy convex hull, and greedy bounded conical hull. In the previous chapters, we saw that in cases with high variability between periods, all methods can produce undesirable results. These can be improved by adding artificial representatives, particularly to enhance feasibility. Therefore, we test the original methods alongside enhanced versions where one artificial representative is added afterwards per scenario, following the same approach as in Section 6.2. We choose to add one worst-case period per scenario, rather than one for the entire dataset, for two reasons: first, we expect that this approach will be less conservative, which is important especially in large datasets; second, it allows us to use the same set of artificial representatives for both the per-scenario and the cross-scenario approaches, making it easier to compare their performance based solely on the representatives selected by the original methods. Since we expect this setup to be slightly conservative, we test it both in isolation and in combination with the blended weights instead of the Dirac weights, as described in Section 3.3.4 and previously applied in the experiments in Chapter 6.

Temporal reduction is a trade-off between reducing computational complexity and maintaining accuracy. We therefore test multiple values for the number of representatives, to see whether some methods perform best with fewer representatives while others are more optimal when more representatives are used. In earlier experiments, especially with the clustering methods, we saw that performance can depend on the number of representatives, and that increasing this number does not always lead to better accuracy. Including all possible values would be computationally infeasible, so we focus on six levels: 20, 50, 100, 200, 400, and 800 representative periods. For cross-scenario selection, this refers to the total number of representatives selected from the full set of 3650 periods (365 days across 10 scenarios). For per-scenario selection, each scenario contributes a proportional share, meaning 2, 5, 10, 20, 40, or 80 representatives per scenario from its 365 days.

7.2.2. Evaluation

To assess the quality of the selected representative periods, we use the same set of evaluation metrics that capture both optimality and computational performance as before. These metrics are applied consistently across all experiments to enable fair and meaningful comparisons.

To evaluate computational performance, we measure the total time required for both the selection of representative periods and the solution of the reduced model. Each experiment with the greedy hull methods is run multiple times to capture variability in runtime. For the clustering-based methods, which involve randomness, we run the experiments using 10 different seeds. We then calculate the speedup based on these run times, where the full stochastic problem is ran once. Average speedup is reported along with 95% confidence intervals, estimated using bootstrapping.

For cost and feasibility evaluation, we use the investment decisions from the reduced model to solve the full operational problem for each of the scenarios, both in-sample and out-of-sample. This allows us to compute the expected

system cost and the expected number of time steps with loss of load. For both in-sample and out-of-sample evaluations, we report the relative regret compared to the full stochastic solution based on these costs along with confidence intervals to show variation. Since k-means and k-medoids are run with ten different seeds, their confidence intervals may be narrower than those of the greedy hull methods, which are deterministic in the way they are implemented.

7.3. Results

We want to understand the effect of all proposed methods on optimality, feasibility and runtime. We will do so one by one, as different insight can be taken from each case. But first of all, we want to explain that we do not include k-means in any of the detailed results plots. The reason for this is that using the original methods without worst-case periods added led to consistently higher relative regret when applying k-means compared to the other methods, while similar speedup was observed. This effect is illustrated in Figure 7.1, which provides an overview of all original methods, without artificial representatives added. The values are averages over all in-sample scenarios and repeated runs. To improve clarity in the subsequent figures, we thus exclude k-means from the remaining original experiment plots.

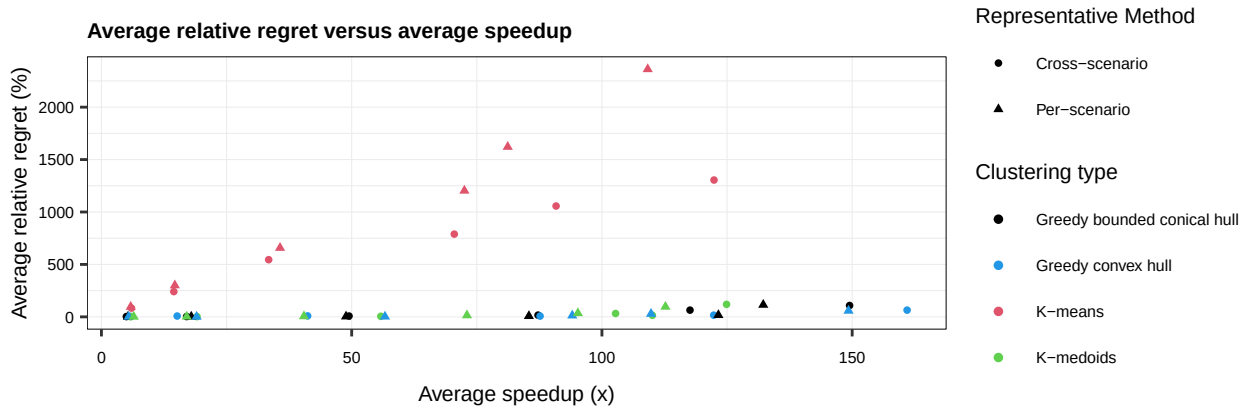

Figure 7.1: Total speedup and average relative regret (%) over all in-sample scenarios under all different original experiment settings, without artificial representatives. The combination of colour and shape indicates the method used. Run time includes both the selection of representative periods and the reduced model run.

Without k-means in the plot, we can better observe the differences in results. To start, we analyze the average speedup versus the average relative regret across all in-sample scenarios for the different combinations of selection methods and implementations: the original methods without artificial representatives, the original methods with worst-case periods added per scenario, and the original methods with worst-case periods added before using blended weights. This is illustrated in Figure 7.2. Afterwards, we delve deeper into the relative regret, loss, and speedup to gain clearer insights into the differences between the selection methods and between per- and cross-scenario evaluations. We also show the confidence intervals for the calculated averages.

When we look at the methods that achieve the highest speedup while also having better relative regret than alternative approaches, the methods that include worst-case periods stand out. The difference in relative regret between using worst-case periods with or without blended weights is very small in this case, as we also observed in other case studies with a limited number of representative periods. It does seem that with blended weights, the maximum speedup is lower than when not using blended weights. This makes sense, since blended weights introduce an additional processing step.

For a speedup below 100 times faster, the original methods, without enhancement through worst-case periods, yield the lowest relative regret. We saw this as well in previous case studies, where worst-case periods introduced a lower bound for relative regret, where adding more periods did not necessarily led to a lower regret. This effect seems even more pronounced in this case study. There are several possible explanations for this. First, the value of lost load was relatively low, especially compared to estimates such as those by Netherlands Authority for Consumers and Markets [22]. It was also ten times lower than what was used in previous case studies in this thesis. This could mean that, in this case study, allowing some loss of load is more cost-effective than making additional investments, which makes the more robust methods have a higher relative regret. Methods that incorporate worst-case periods tend to produce

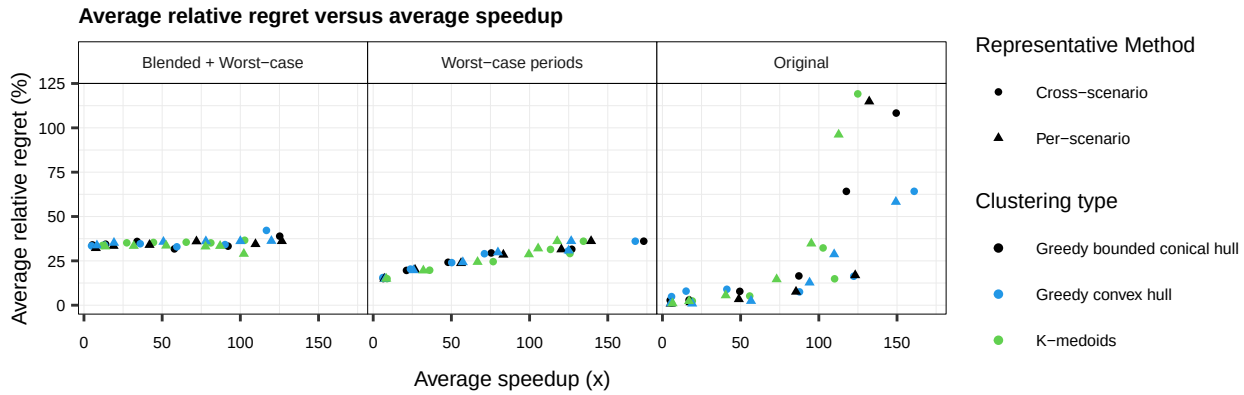

Figure 7.2: Average relative regret (%) versus average speedup (x) over all in-sample scenarios under different settings. The combination of colour and shape indicates the method used.

representatives where there is high risk for loss of load, pushing the system to invest more conservatively. This can result in suboptimal investment choices compared to just allowing limited loss of load, which is likely what happens for the original methods. Second, the case includes only renewable energy sources as investable technologies. In worst-case scenarios with low availability, the system is forced to invest even more in renewables to cover demand. If availability approaches zero, the required capacity becomes unrealistically high and results in high investment costs. In more realistic settings, other generation types with higher operational costs could be used to cover peaks or extreme shortages. This would reduce total investment and overall costs. Therefore, we expect different outcomes in case studies with a higher value of lost load and a more diverse set of investable technologies, which could be a future extension of this research.

We observe a difference in the trajectory of relative regret between using only worst-case periods and combining them with blended weights. With blended weights, relative regret remains stable across different numbers of representative periods, so similar outcomes are reached for both the highest and lowest speedups. For the method using worst-case periods without blended weights, we see some improvement in relative regret as speedup decreases. We believe this is caused by the weighting scheme, which automatically assigns lower weights to the worst-case periods as the total number of periods increases. This reduces their influence, making the method less conservative and improving optimality. With a different weighting method, this trend might not be visible.

In-sample cost evaluation

We continue with the optimality outcomes shown in Figure 7.3. This figure presents the average relative regret across all in-sample scenarios, using the fixed investments from the experimental setup. The results are shown separately for different numbers of representative periods and for the different methods (original, with worst-case representatives, and with worst-case and blended weights). The scale is consistent, allowing for fair comparison. To improve readability, results for per-scenario and cross-scenario evaluations are placed in separate columns within each graph.

In Figure 7.3 we see that the earlier observed change regarding whether adding artificial representatives is beneficial for the relative regret or not happens around 50 representative periods. With 50 representative periods, some combinations of the original methods have a lower relative regret compared to when worst-case periods are added. The best performing with this low number of representative periods is the greedy convex hull. When using more representative periods, there is no longer a clear distinction between methods, and no single method consistently outperforms the others. Their intervals often overlap, and the means are similar.

Differences between per-scenario and cross-scenario evaluations are also small and vary across selection methods. For the hull methods without artificial representatives, per-scenario results show slightly lower means, although the confidence intervals overlap, so no significant conclusion can be drawn. We expect the greedy hull algorithms to perform better on smaller, more coherent sets, as their approximations can be more accurate. In the per-scenario case, the sample size is smaller and the periods are likely more similar, which may support better approximations than in the cross-scenario case, where all periods are combined into a single set.

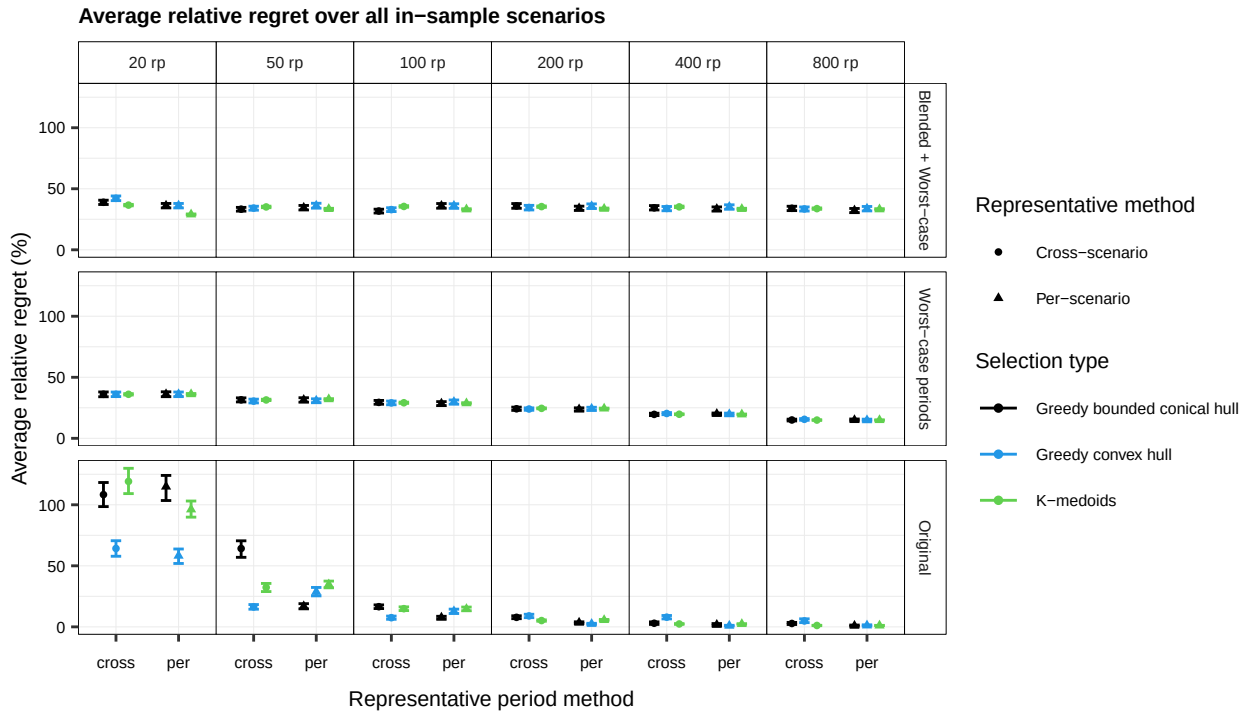

Figure 7.3: Average relative regret (%) over all in-sample scenarios under different settings. The combination of colour and shape indicates the method used. The error bars show the 95% confidence intervals for the mean, based on bootstrapping.

In-sample loss of load evaluation

Figure 7.4 shows the average number of additional time steps with loss of load compared to the stochastic solution, for different methods and variations. As expected we see here the advantages of adding the worst-case periods. The methods using these, with or without blended weights, result in significantly fewer hours with loss of load than the original methods. This supports the idea that the increased costs reflect a trade-off for improved robustness. Even with only 20 representative periods, there is no additional loss of load compared to the stochastic solution when using worst-case periods.

Another observation from the figure is that the lower relative regret for the greedy convex hull method without artificial representatives and 20 representative periods is related to less loss of load compared to k-medoids. However, the number of time steps with loss of load is still very high, over 1500 hours across the 10 scenarios. This suggests that a perfect hull was not necessarily found, which aligns with earlier indications in previous chapters.

For the original methods, the reduction in costs with more representative periods seen in Figure 7.3 coincides with a reduction in infeasibility. This suggests that later-added representatives help improve feasibility and, in turn, optimality. For the methods using worst-case periods, however, the level of infeasibility remains fairly stable as more representatives are added. This implies that most of the loss of load that could be avoided, or was optimal to avoid, was already addressed by the worst-case periods and the first few representatives. Combined with the earlier observation that cost reductions in these methods are likely due to the weighting scheme rather than additional representatives, this suggests that the optimal number of representatives for achieving feasibility might be reached relatively early in the process. Beyond that point, it might be more interesting to see the effect of different methods for calculating the weights rather than adding more and more representatives to include extra information.

Out-of-sample evaluation

For the out-of-sample evaluation, we tested the fixed investment decisions on 9 unseen scenarios to check for any major differences. Testing on more scenarios was not possible due to time considerations. Similar to what we observed in Chapter 5, the trends remain consistent, and no new conclusions emerge from the figures. We see that with

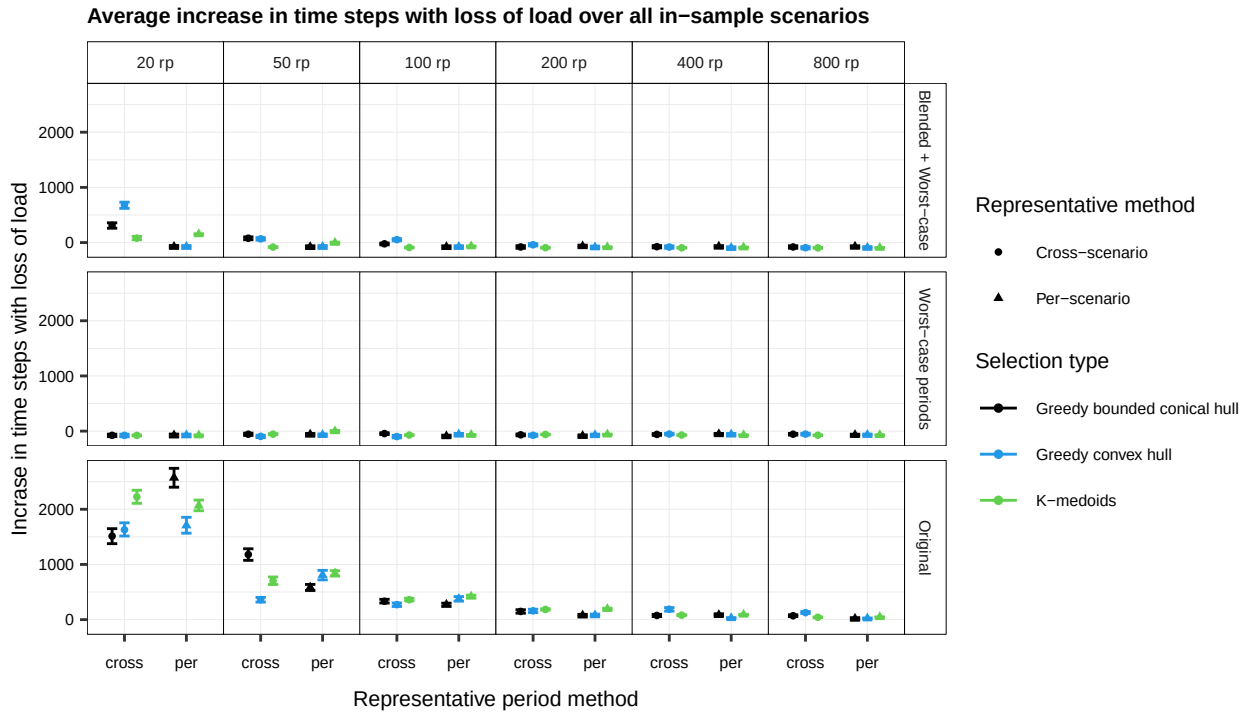

Figure 7.4: Average increase in time steps with loss of load over all in-sample scenarios under different settings. The combination of colour and shape indicates the method used. The error bars show the 95% confidence intervals for the mean, based on bootstrapping.

limited number of representatives, adding worst-case periods without blended weights again leads to no additional loss of load compared to those of the full stochastic model. The relative regret is in that case most optimal among alternatives. When increasing the number of representatives, costs are lower for the original methods, while loss of load is consistently more reduced when using the methods that include worst-case periods. Among the original methods, the greedy convex hull again shows best results. When using more representatives, the performance differences between k-medoids and the hull methods, show overlapping results, with no method clearly outperforming the others consistently.

Across all methods, improvements tend to plateau after around 200 representative periods. The only exception is the method using worst-case periods without blended weights, which continues to reduce costs. As noted earlier, this is most likely due to the decreasing weights of the extreme scenarios as more representatives are added, leading to less overly conservative investment decisions. For a full overview of these results, we refer to Appendix C, where the relevant plots are provided. We do not discuss them in further detail here.

Runtime evaluation

Lastly, we evaluate the speedup when using the reduced models, which helps in assessing the trade-off between accuracy and computational effort. Figure 7.5 shows the average total speedup for each method, including runtime for both the selection of representative periods and the optimization of the reduced model.

In Figure 7.5, we see that for a small number of representative periods, e.g. 20, the hull methods are faster than k-medoids. The confidence intervals for the means do slightly overlap and are larger for the hull methods, since they are based on fewer runs. For k-medoids, the average is calculated over 10 runs (as 10 different seeds were used), whereas for the hull methods, due to time limitations, only 5 runs were done. Still, both a lower relative regret and faster run time means that the greedy convex hull is very attractive with this number of representatives.

When using more than 100 representative periods, the hull methods have lower speedup than k-medoids. This becomes more pronounced when using worst-case periods. To investigate this, we separated the total runtime into processing time and optimization time; see Appendix C for the detailed plots. Notably, the decrease in speedup is

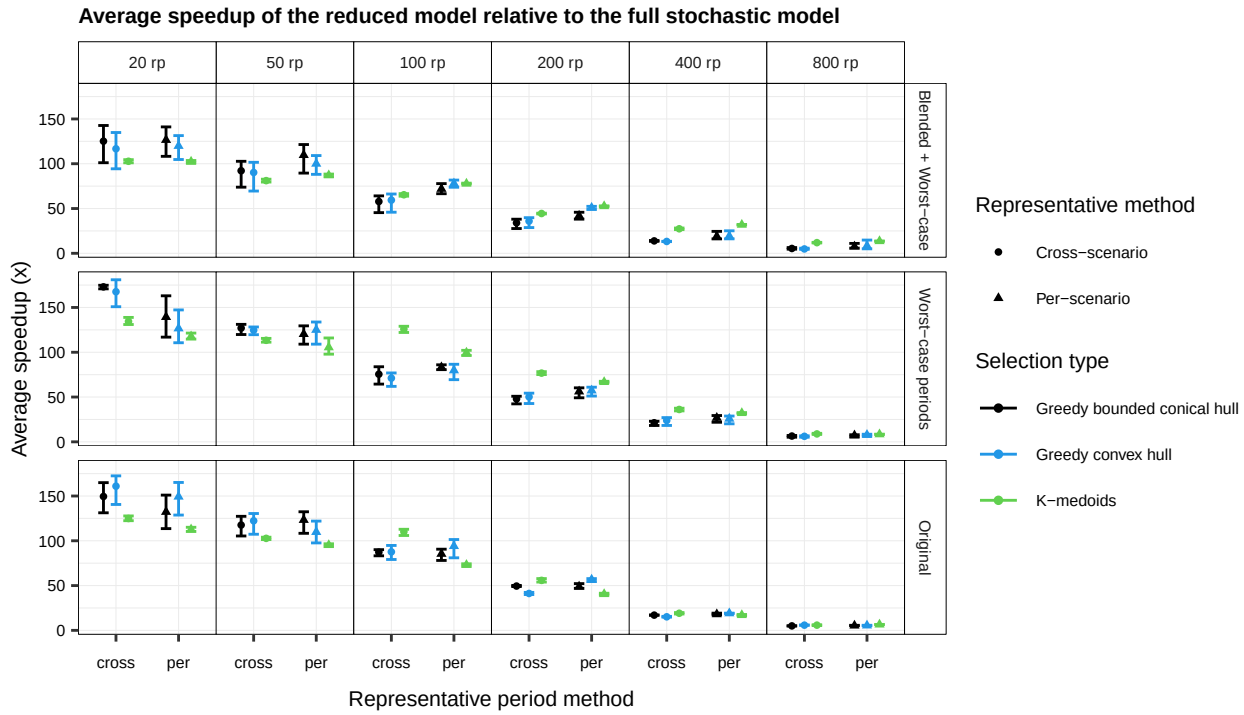

Figure 7.5: Average speedup (x) of the reduced model compared to the full stochastic model, calculated by dividing the total runtime of the full model by the total runtime of the reduced model. This includes the selection of representatives and creating and optimizing the reduced model. The combination of colour and shape indicates the method used. The error bars show the 95% confidence intervals for the mean, based on bootstrapping.

mostly due to higher processing time. Further inspection of the implementation revealed that, for the greedy hull method, the projected subgradient descent algorithm struggled to update these distances efficiently, spending more time on small incremental updates. This is an implementation detail that can likely be improved. This became a larger problem when multiple worst-case periods were added beforehand, as in the cross-scenario methods in the top row. This explains why the cross-scenario approach is, in that case, slower than the per-scenario approach.

7.4. Conclusion

Overall, we can conclude that for this particular case study and for both in-sample and out-of-sample evaluation, the newly investigated methods are an improvement over standard algorithms when a limited number of representatives is used. This is desirable, as often the computational complexity will only allow such a limited number. We summarize the findings once more below.

Although the original hull methods were not able to find a valid hull and did not result in additional loss of load, they did show a lower relative regret and less infeasibility than the alternative k-medoids when using 20 representatives. With more representatives, this distinction did not persist. This suggests that further work is needed to improve the implementation or investigate whether more exact methods could make hull-based approaches more effective. What we can conclude, however, is that both hull methods and k-medoids are preferable to k-means, which showed consistently higher costs across all tested values for the number of representative periods.

We did, however, observe that using additional worst-case periods immensely improves feasibility. This came at the cost of increased system costs in this setting. When computational limitations only allow a small number of representative periods to be used, in this case below 50 representatives for the total of 3650 days, the use of worst-case periods can ensure no additional loss of load and best relative regret among alternatives. However, for a higher number of representatives, the solution with worst-case periods results in higher relative regret than not using them, which does not improve at a high rate, and original methods might be more suitable. However, loss of load is consistently higher when using the original methods, so for robustness, the methods using worst-case periods should be used.

We believe further cost improvements could be achieved by adapting the weighting scheme when using worst-case periods. In particular, we observed that with Dirac weights, costs decreased as the number of representatives increased, likely because the weight assigned to worst-case periods decreased, while the level of loss of load remained stable. Although we initially hypothesized similar effects for blended weights, we saw the opposite. It is unclear whether this is due to the concept of blended weights itself, or whether the implementation using projected subgradient descent is simply not suited to large-scale cases where samples differ widely. Further research could explore alternative weighting approaches.

In terms of selection methods, differences between per-scenario and cross-scenario approaches were relatively small for the number of representatives tested. In terms of speed-up, the hull methods are faster than k-medoids for a low number of representative periods. They get slower than the alternative when more representatives need to be found. This effect was especially visible when using worst-case periods. It is likely linked to the implementation of the subgradient descent, which fails to calculate distances to the hull correctly. When using worst-case periods, this resulted in more distance calculations and therefore, an increase in runtime.

While these findings offer valuable insights, we also note that some of the case-specific choices may limit generalization. We could not extend the case study further due to the time limit of this thesis, but we do have advice on future work, especially since we believe that the proposed methods might be even more beneficial in those cases. For instance, cross-scenario selection may be more effective with fewer representatives or in systems with stronger seasonal patterns, such as those with seasonal storage. Likewise, the cost difference introduced by using worst-case periods might be smaller in cases with a higher, and more realistic, value of lost load, or with a more diverse set of investable technologies beyond just renewables. We therefore recommend extending the current case study in future work to include a broader set of generation technologies, a higher value of lost load, and additional complexity such as storage. This would help clarify the effects of the proposed methods under more realistic system conditions.

8

Conclusion

In this thesis, we explored how representative periods can be used in stochastic programming formulations of energy models, with a strong emphasis on ensuring feasibility before addressing optimality. We showed that by constructing either a convex or a bounded conical hull over the original set of periods, feasibility for the full problem can be guaranteed. This is a significant theoretical result, and when the greedy implementation method succeeds, it offers strong advantages. However, identifying a suitable convex or bounded conical hull is not always straightforward, and in some cases, it may not be possible at all. To address this, we proposed enhancements that focus on selecting extreme, worst-case periods that are particularly relevant for feasibility. These can be combined with clustering approaches that emphasize cluster centers, enabling a balance between feasibility and optimality. Blended weights further help to reduce over-conservativeness. We applied the method to a European case study to demonstrate its potential, especially when we can only use a limited number of representatives. Further work could improve the current approximation strategy for finding a hull and focus on further managing the conservativeness introduced by the selected worst-case periods.

In this final chapter, we first provide a detailed summary of the main findings and revisit the research questions in Section 8.1. Afterwards, we reflect on the limitations of our approach and highlight promising directions for future research 8.2.

8.1. Summary of findings

In energy models, the demand for greater technical, spatial, and temporal detail has increased due to the growing share of renewable energy sources. This has led to research focused on reducing model size while retaining essential characteristics. At the same time, uncertainty in both renewable energy availability and demand levels has made models based on average predictions more costly in practice than those that explicitly incorporate uncertainty. While various methods exist for modeling uncertainty, such as stochastic programming and robust optimization, these approaches further increase computational complexity, which is already a challenge in deterministic settings for large-scale problems.

This thesis addressed that challenge by exploring strategies to reduce complexity in uncertain energy models. Specifically, we focused on temporal reduction in stochastic programming, as this can complement other established techniques such as scenario reduction and decomposition strategies. Among existing temporal reduction methods, representative periods are the most widely used. Multiple approaches exist for using them, but no clear best method has emerged, and earlier work has emphasized the importance of capturing worst-case periods to ensure feasibility. Therefore, we further investigated existing methods and the recently introduced hull-based approach, which focuses more explicitly on extreme periods, and applied it specifically to stochastic programming. While our research is largely based on generation expansion planning, the underlying methods and theoretical contributions have broader applicability across energy system models, although further research may be needed for those extensions. Below, we will walk through each of our key findings.

First, we demonstrated how representative periods can be used effectively in stochastic programming. While earlier work showed that representative periods were typically selected per scenario, we showed that they can also be shared

across multiple scenarios, making them scenario-independent. This avoids redundancy caused by selecting similar periods separately for each scenario. However, because the selection of representative periods relies on approximate algorithms, these may perform better on smaller sets (per-scenario) than on larger, combined sets. In the final case study, we did not observe a significant difference in average performance, both in runtime and cost, between the per-scenario and cross-scenario approaches. Moreover, all theoretical results apply equally to both approaches, making the choice between them less critical from a feasibility standpoint. Still, we believe that with improved approximation algorithms, as discussed later, the benefits of the cross-scenario method may become more apparent.

We investigated and compared the two most common clustering approaches, k-means and k-medoids, to the greedy convex hull and greedy bounded conical hull methods. In previous research, it became evident that among the clustering algorithms k-means and k-medoids, performance is case-dependent, and neither provides guarantees regarding feasibility. Our results support this observation and show that hull methods, based on finding either a convex or a bounded conical hull as currently implemented in TulipaClustering.jl [23], can provide such feasibility guarantees. We investigated this further and proved that in a stochastic programming formulation for generation expansion planning, when the representatives indeed form one of the two hulls, a feasible solution can be constructed from the optimal solution of the reduced model. Moreover, if the reduced model shows no loss of load, there will also be no loss of load in the original model under this solution. When using blended weights, also implemented in TulipaClustering.jl, which applies the convex or conical weights in the objective, we also proved that the solution provided by the reduced model gives an upper bound on the system costs. This makes the use of these hull methods very interesting. The results are based on the assumption that either the convex or bounded conical hull was indeed found, which is not always the case in practice. We also ignored inter-period constraints, such as ramping or storage, which connect periods chronologically and influence operational decisions. This is a common assumption. However, we still proposed ways of handling these constraints. While this can ensure feasibility in the reduced model, as in the case without inter-period constraints, it may limit the practical benefits in terms of computational efficiency or optimality.

To test the effect of the theoretical results, we compared a greedy implementation for finding the convex or bounded conical hull with standard clustering algorithms, using specifically designed case studies. These case studies were constructed with different levels of correlation and random noise, to explore where limitations in one of the methods might occur. In the first case study, the samples were designed so that a convex hull could be formed with just three representatives, which were easily found by the greedy algorithm. We saw that the theoretical results indeed applied: feasibility was ensured by both the greedy convex hull and greedy bounded conical hull method, and thanks to this, a close-to-optimal solution was found using fewer representative periods than required by the standard clustering algorithms. As we increased the variety in two additional case studies, the hull methods still succeeded in finding a hull and ensuring feasibility. They required slightly more representatives than in the first case, but still fewer than the clustering methods. Only in the final case study was a hull not found, and all methods, both the hull-based and the clustering algorithms, struggled to achieve the same low relative regret as before. This motivated us to investigate how to enhance the methods further to ensure feasibility in more difficult cases.

Across all designed case studies, we also observed that while feasibility and optimality are related, a solution with no loss of load does not necessarily mean the most optimal solution is found. Adding more representative periods after identifying a hull can help improve optimality. For k-means and k-medoids, we also saw that a low relative regret was not always associated with zero additional loss of load. In addition to this, the results of these two methods showed considerable variation due to their random initialization. Using 10 different seeds, we observed large interquantile intervals, with sharp increases and decreases in relative regret occurring from only small changes in the number of representative periods.

To improve performance of the greedy hull methods in cases where all methods struggled to achieve low relative regret, we examined a two-dimensional example that shared this difficulty. In two dimensions, we were able to construct a bounding polytope around the convex hull of the samples, which led to artificially created representatives that ensured feasibility. This bounding polytope can then be combined with for example k-means or k-medoids to find representatives important for optimality rather than feasibility. With these artificial representatives, we were able to achieve lower relative regret and zero infeasibility, with only a limited number of representatives. When increasing the number of representatives, they resulted in a more conservative investment solution than alternatives, leading to higher costs. Using blended weights, we could scale down the influence of the worst-case representative as more representatives were added, reducing costs again.

From analyzing the two-dimensional case study behaviour, we were able to identify a specific worst-case representative that ensured feasibility in two dimensions, one with the highest demand and lowest availability-to-demand ratio, which we proved. While not directly extendable to higher dimensions, this served as a useful heuristic for improving

the earlier case study in which all methods struggled. By combining this single worst-case representative with one of the original methods for selecting the remaining representatives, we saw a large improvement. Blending weights again had a positive effect on costs, better balancing the influence of the extreme point.

Overall, when a convex or bounded conical hull was found, the results were very positive, with both approaches performing similarly. However, in cases where a hull was not found, additional extreme, worst-case representatives may be needed to maintain feasibility. To test the methods on a more realistic case, we applied them to a European case study with high renewable penetration and multiple scenarios. The representative periods were created from 10 scenarios, and decisions were evaluated on 9 additional scenarios.

The results from the earlier case studies extend to the large European case study. While the original greedy hull methods again failed to ensure feasibility, likely because they did not successfully identify a valid hull, adding worst-case periods significantly reduces infeasibility. Even when using only 20 representatives, the additional number of time steps with loss of load compared to the full stochastic solution was below 0. This comes at a higher relative regret, but when using a limited number of representatives, this regret is still lower than not using artificial representatives.

When comparing the greedy convex hull methods with the clustering methods, we saw that for a low number of representative periods, the greedy convex hull method performed best in terms of relative regret, among the original methods. As more periods were added, the differences between the hull methods and the k-medoids method became smaller, with overlapping confidence intervals. In terms of speed, the hull methods were faster with fewer representatives, but slowed down as more representatives were included. Across the tested number of representative periods, no clear performance difference emerged between per-scenario and cross-scenario selection. Lastly, k-means consistently performed worse in both cost and feasibility.

Differences between different methods and between per- or cross-scenario approaches were very small when using worst-case periods. Differences between blended and non-blended weights also appeared small for a lower number of representatives, where using non-blended weights was slightly better in terms of loss of load and relative regret. Even with a higher number of representatives, blended weights did not yield improvements for the high relative regret. However, we observed that when using Dirac weights, reducing the weight of worst-case representatives through the natural effect of using more representatives lowered costs while maintaining stable levels of loss of load. This shows that the weighting strategy is a promising area for future research, as it can largely influence the costs when the most important representatives are already identified.

8.2. Discussion and future work

We showed how representative periods can be used in stochastic programming and, more generally, that when feasibility is a key concern, alternative selection methods beyond standard clustering approaches can be more effective. In particular, methods based on constructing a convex or bounded conical hull provide strong theoretical guarantees and can significantly reduce the number of representative periods needed. However, these methods still require improvement before they can be reliably used as a default option for temporal reduction. Below, we give suggestions for future work and discuss how the proposed methods could be integrated into other types of models, along with the further research needed to support this. We also highlight assumptions made throughout this thesis that should be investigated further.

First, we believe that the construction of a convex or bounded conical hull can be improved to perform even better on large-scale studies. At the moment, a greedy implementation is used, which works well in smaller examples. However, in larger settings such as the final case study, the projected gradient descent used to calculate distance to the hull and the corresponding blended weights showed limitations. The current approach may fail to compute accurate weights depending on parameters like the learning rate, number of iterations, or tolerance. While tuning these can help, it may be worthwhile to explore alternative approximation algorithms that are more stable or better suited for high-dimensional cases, without adding significant runtime. Improving this would have a strong positive impact on the practical usability of the hull-based methods. Also, from the large European case study we noted that distance calculations in the process of finding the hull often led to small differences and were sometimes incorrect, which could be due to the large sample size or large differences between samples. As this is part of the hull implementation, it could be interesting to see whether improvement here would even speed up the process more.

A second direction for future work is improving how worst-case representatives are identified and used to ensure feasibility. This approach, which can act as a fallback when finding a full hull is not possible, already showed its potential. In two dimensions, we could guarantee feasibility based on a well-defined worst-case representative, but the proof is

not yet extended to higher dimensional models with additional constraints such as transmission, ramping and storage. Defining artificial worst-case representatives differently or adapting the selection method may lead to stronger guarantees and better applicability to more complex models. Since the worst-case representative already performed well in practice, strengthening this method could offer significant benefits. Also, we saw from the European case study that the method for calculating the weights had no influence on loss of load, while having a significant influence on the costs. Future research in this field, with better refinement and testing of different weighting methods, could lead to much better results in terms of costs and finding the balance between conservativeness and feasibility. Moreover, currently the worst-case representatives for the European case study were identified per-scenario, why one over all scenarios might be more optimal and less redundant.

These two directions both aim to improve feasibility when using representative periods. Currently, representative periods are not selected explicitly based on feasibility, and no clear metric exists for evaluating this across different selection methods. We therefore recommend that robustness and feasibility receive more attention when applying temporal reduction techniques in energy models. Standard clustering algorithms like k-means and k-medoids are often used due to their simplicity, but their dependence on random initialization leads to inconsistent results. In our experiments, relative regret sometimes spiked from small changes in the number of representatives or due to random seeds. Also, their inability to select points on the boundary lead consistently to more infeasibility. While they may perform well in some cases, they do not offer the theoretical guarantees that methods like the hull-based approaches can provide. Moreover, k-means in particular was not able to generalize well to the large case study and should only be carefully used in energy model applications.

Another important area for future research is incorporating inter-period constraints. While our theoretical results rely on ignoring these constraints, they are essential in many real-world models, especially when storage dynamics are involved. We discussed possible ways to include ramping constraints, but acknowledge that most are either impractical or overly conservative. Further research should investigate how these constraints can be integrated with representative periods, especially in cross-scenario settings. Extending our methods to other types of models may require subtle changes to preserve feasibility guarantees or to achieve similar practical performance. However, this is not a limitation specific to our approach, any method using representative periods must eventually address this issue. We especially advise this to repeat cross-scenario and per-scenario experiments when seasonality due to storage is added, as differences might be more distinctive in that case.

Throughout this thesis, we highlighted the trade-off between feasibility and optimality. Hull-based and worst-case-focused methods are strong in ensuring feasibility, but may miss out on more central periods that are important for cost-effectiveness. We believe future work should explore how to balance both the number of extreme representatives and the number of total representatives necessary for improvements.

Another factor that deserves more attention is the choice of distance metric, normalization, and weighting strategy when selecting representative periods. These are often set upfront without much analysis, but they can affect performance. In our work, we used both cosine and squared euclidean distances, and while they performed similarly in most cases, there were examples where one clearly outperformed the other. We also relied on a single normalization approach and tested two weighting strategies, but many other options exist. A systematic review of these choices could reveal which combinations work best under different modeling assumptions.

Lastly, for practical use in real-world studies, additional work is needed to refine how these methods are applied. Temporal reduction techniques should not be considered in isolation but used in balance with other methods such as scenario reduction, Benders decomposition, or spatial aggregation. Whether to prioritize spatial, temporal, or technical detail will depend heavily on the case and the model. A careful analysis should precede any decision. Moreover, combining reduction techniques could allow higher resolution in other model dimensions, for instance, keeping spatial detail while simplifying the temporal side, and we encourage future research to explore how these combinations affect performance and solution quality.

A

Feasibility in the hull for general stochastic programming models

We extend Theorem 4.1 for Generation Expansion Planning to more arbitrary formulations of stochastic programming problems. To do so, we consider a formulation of the following form:

$$\min \quad f(\vec{x}) + \sum_{j=(s,p)\in\mathcal{J}} \pi_s \cdot g(\vec{y}_j) \tag{A.1a}$$

$$\text{s.t.} \quad A\vec{y}_j = \vec{d}_j \qquad \forall j \in \mathcal{J} \tag{A.1b}$$

$$B\vec{y}_j + C_j\vec{x} \leq \vec{v}_j \quad \forall j \in \mathcal{J} \tag{A.1c}$$

$$\vec{y}_j \in \mathbb{R}^m \qquad \forall j \in \mathcal{J} \tag{A.1d}$$

$$\vec{x} \in \mathbb{R}^n_{\geq 0} \tag{A.1e}$$

Here the vector $\vec{x}$ represents the first stage decision variables which are made before the exact scenario is known. The second stage variables are represented by the vector $\vec{y}_j$ and we optimize their expected value. We then consider two types of constraints: an equality constraint (A.1b) and inequality (A.1c) constraint. The equality constraint in the formulation is just a matrix representation of the balance constraint which only consist of second stage decision variables. The right hand-side is indexed by j, to allow different values per period. The inequality constraint covers all other types of intra-period constraints. In this general formulation of Theorem 4.1, we do not allow inter-period constraints.

For finding a representative set $\mathcal{R}$ for $\mathcal{J}$, we assume that the representative periods are constructed based on $\vec{d}_j$ and the uncertain components of $\vec{v}_j$ and C_j. Here, the uncertain parts refer to the period-dependent components, which we denote with the suffix u. When these are combined with the constant, period-independent parts (denoted with the suffix c), we obtain the original vectors and matrices:

$$\vec{v}_j = \vec{v}^c + \vec{v}^u_j \quad \forall j \in J, \tag{A.2}$$

$$C_j = C^c + C^u_j \quad \forall j \in J. \tag{A.3}$$

Define the reduced model based on representative periods $\mathcal{R}$ as follows:

$$\min \quad f(\vec{x}) + \sum_{r\in\mathcal{R}} W_r \cdot g(\vec{y}_r) \tag{A.4a}$$

$$\text{s.t.} \quad A\vec{y}_r = \vec{d}_r \qquad \forall r \in \mathcal{R} \tag{A.4b}$$

$$B\vec{y}_r + C_r\vec{x} \leq \vec{v}_r \quad \forall r \in \mathcal{R} \tag{A.4c}$$

Without imposing additional restrictions on $\vec{v}^c$ and C^c, the previous theorems extend to this general formulation only if the weights satisfy Condition 3. In contrast, when a conical hull is obtained instead of a convex hull (as in Condition

2), further assumptions are required. We elaborate on this in the subsequent corollary.

Theorem A.1. *Consider the stochastic programming formulation in* (A.1) *and find a representative set $\mathcal{R}$ perfectly representing $\mathcal{J}$, where the weights satisfy Condition 1 and 3. Let $(\vec{x}^\star, \vec{y}_r^\star)$ denote the optimal solution to the reduced model in* (A.4).

For the original problem, we construct a solution by setting $\vec{x} = \vec{x}^\star$ and defining the second-stage variables as follows:

$$\vec{y}_j = \sum_{r \in \mathcal{R}} \lambda_{r,j} \cdot \vec{y}_r^\star, \quad \forall j \in \mathcal{J} \tag{A.5}$$

This solution will be a feasible solution to the original problem.

Proof. To show that the proposed solution is a feasible solution in the model formulated by (A.1), we have to verify the equality constraints and then the inequality constraints. But first, we show that the based on Condition 3 and the way that we clustered the weights are convex, we can deduce the following equalities:

$$\vec{d}_j = \sum_{r \in \mathcal{R}} \lambda_{r,j} \vec{d}_r \tag{A.6}$$

$$\vec{v}_j = \sum_{r \in \mathcal{R}} \lambda_{r,j} \vec{v}_r \tag{A.7}$$

$$C_j = \sum_{r \in \mathcal{R}} \lambda_{r,j} C_r \tag{A.8}$$

This will be used in the remainder of the proof. We start with the equality constraints given by (A.1b). Let $j \in \mathcal{J}$. We rewrite the left-hand side of the equality constraints:

$$\begin{aligned} A\vec{y}_j &= A \sum_{r \in \mathcal{R}} \lambda_{r,j} \cdot \vec{y}_r^\star \\ &= \sum_{r \in \mathcal{R}} \lambda_{r,j} \cdot A\vec{y}_r^\star \\ &= \sum_{r \in \mathcal{R}} \lambda_{r,j} \cdot \vec{d}_r \\ &= \vec{d}_j \end{aligned} \tag{A.9}$$

The last line is equal to the right-hand side. This proofs that all the equality constraints are still satisfied. For the inequality constraints given by (A.1c), we perform similar steps:

$$\begin{aligned} B\vec{y}_j + C_j\vec{x} &= B \sum_{r \in \mathcal{R}} \left(\lambda_{r,j} \cdot \vec{y}_r^\star\right) + C_j\vec{x} \\ &= \sum_{r \in \mathcal{R}} \left(\lambda_{r,j} \cdot B\vec{y}_r^\star\right) + C_j\vec{x} \\ &\leq \sum_{r \in \mathcal{R}} \lambda_{r,j} \left(\vec{v}_r - C_r\vec{x}\right) + C_j\vec{x} \\ &= \vec{v}_j - \sum_{r \in \mathcal{R}} \left(\lambda_{r,j} C_r\vec{x}\right) + C_j\vec{x} \\ &= \vec{v}_j \end{aligned} \tag{A.10}$$

We conclude that the inequality constraints are also satisfied by the provided solution. Since both types of constraints hold, the solution is feasible. □

It is possible to extend this theorem to representative periods that form a bounded conical hull, so when the weights satisfy Condition 2 rather than Condition 3. For the uncertain parts, we still ensure that the following holds:

$$\vec{v}_j^u = \sum_{r \in \mathcal{R}} \lambda_{r,j} \vec{v}_r^u \tag{A.11}$$

$$C_j^u = \sum_{r \in \mathcal{R}} \lambda_{r,j} C_r^u \tag{A.12}$$

This remains unchanged from the previous formulation. However, for the certain parts, we also assume that the following holds:

$$\vec{v}^c \geq \sum_{r\in\mathcal{R}} \lambda_{r,j}\vec{v}^c \tag{A.13}$$

$$C^c\vec{x} \leq \sum_{r\in\mathcal{R}} \lambda_{r,j}C^c\vec{x} \tag{A.14}$$

Comparison is done element-wise. Since both the weights and the first-stage variables are non-negative, it follows that $\vec{v}^c$ must consist of non-negative entries and that C^c must be a matrix with non-positive entries. We present the implication as a corollary:

Corollary A.2. *Consider the stochastic programming formulation in* (A.1) *and find a representative set $\mathcal{R}$ perfectly representing $\mathcal{J}$, where the weights satisfy Condition 1 and 2. Additionally, assume that the entries of $\vec{v}^c$ are non-negative and that the entries of C^c are non-positive. Let $(\vec{x}^\star, \vec{y}_r^\star)$ denote the optimal solution to the reduced model in* (A.4).

For the original problem, we construct a solution by setting $\vec{x} = \vec{x}^\star$ and defining the second-stage variables as follows:

$$\vec{y}_j = \sum_{r\in\mathcal{R}} \lambda_{r,j}\cdot\vec{y}_r^\star \qquad\qquad \forall j \in \mathcal{J} \tag{A.15}$$

This solution will be a feasible solution to the original problem.

Proof. We only need to show that the inequality constraint still holds under this assumption, the proof for the equality constraint remains unchanged from Theorem A.1.

First, note that we can combine the assumptions in (A.13) and (A.14) with Condition 2 to obtain the following:

$$\begin{aligned}
\vec{v}_j &= \vec{v}^c + \vec{v}_j^u && \text{(A.16)}\\
&= \vec{v}^c + \sum_{r\in\mathcal{R}} \lambda_{r,j}\vec{v}_r^u && \text{(A.17)}\\
&\geq \sum_{r\in\mathcal{R}} \lambda_{r,j}\left(\vec{v}^c + \vec{v}_r^u\right) && \text{(A.18)}\\
&= \sum_{r\in\mathcal{R}} \lambda_{r,j}\vec{v}_r && \text{(A.19)}
\end{aligned}$$

A similar argument leads to the following inequality for C_j:

$$\begin{aligned}
C_j &= C^c + C_j^u && \text{(A.20)}\\
&= C^c + \sum_{r\in\mathcal{R}} \lambda_{r,j}C_r^u && \text{(A.21)}\\
&\leq \sum_{r\in\mathcal{R}} \lambda_{r,j}\left(C^c + C_r^u\right) && \text{(A.22)}\\
&= \sum_{r\in\mathcal{R}} \lambda_{r,j}C_r && \text{(A.23)}
\end{aligned}$$

With these two inequalities, we can prove that the inequality constraints in the original model still hold. Let $j \in \mathcal{J}$. We start by rewriting the left-hand side of the inequality constraint (A.1c):

$$\begin{aligned}
B\vec{y}_j + C_j\vec{x} &= B\sum_{r\in\mathcal{R}} \lambda_{r,j}\vec{y}_r^\star + C_j\vec{x}\\
&= \sum_{r\in\mathcal{R}} \lambda_{r,j}B\vec{y}_r^\star + C_j\vec{x}\\
&\leq \sum_{r\in\mathcal{R}} \lambda_{r,j}\left(\vec{v}_r - C_r\vec{x}\right) + C_j\vec{x}\\
&\leq \vec{v}_j - \sum_{r\in\mathcal{R}} \lambda_{r,j}C_r\vec{x} + C_j\vec{x}\\
&\leq \vec{v}_j - C_j\vec{x} + C_j\vec{x}\\
&= \vec{v}_j
\end{aligned} \tag{A.24}$$

Since the final line equals the right-hand side of the inequality constraint, the constraint is satisfied. We conclude that the theorem holds. □

We also extend this result by showing that, when using blended weights as in (4.27), the objective value of the constructed solution is equal to that of the reduced model.

Theorem A.3. *Consider the stochastic programming formulation in* (A.1) *with the additional assumption that the cost functions are linear. Find a representative set $\mathcal{R}$ perfectly representing $\mathcal{J}$, where the weights satisfy Condition 1 and 3. Let $(\vec{x}^\star, \vec{y}_r^\star)$ denote the optimal solution to the reduced model in* (A.4) *when the weights are defined as in* (4.27)*. Let $z^\star$ denote the corresponding objective value.*

For the original problem, we define a solution by setting $\vec{x} = \vec{x}^\star$ and defining the second-stage variables as follows:

$$\vec{y}_j = \sum_{r \in \mathcal{R}} \lambda_{r,j} \cdot \vec{y}_r^\star, \quad \forall j \in \mathcal{J} \tag{A.25}$$

This solution is feasible for the original problem. Moreover, its objective value in the original problem is equal to $z^\star$.

Proof. In Theorem A.1, we already showed that the proposed decision variables form a feasible solution to the original model. It remains to show that, using the defined weights, the cost of this solution equals the optimal objective value of the reduced model. The objective function of the original model is given in (A.1). Substituting the proposed solution into this expression gives:

$$f(\vec{x}^\star) + \sum_{j=(s,p) \in \mathcal{J}} \pi_s \cdot g\left(\sum_{r \in \mathcal{R}} \lambda_{r,j} \cdot \vec{y}_r^\star \right) \tag{A.26}$$

By linearity of the function g, this is equal to:

$$f(\vec{x}^\star) + \sum_{j=(s,p) \in \mathcal{J}} \pi_s \cdot \sum_{r \in \mathcal{R}} \lambda_{r,j} \cdot g(\vec{y}_r^\star) \tag{A.27}$$

Reordering the sums and using (4.27) gives:

$$f(\vec{x}^\star) + \sum_{r \in \mathcal{R}} W_r \cdot g(\vec{y}_r^\star) \tag{A.28}$$

This is exactly the objective function of the reduced model with blended weights. Since the decision variables are optimal for that model, the resulting cost equals $z^\star$. This completes the proof. □

B

Feasibility and optimality results

Method	Representatives	In-sample		Out-of-sample	
		Relative regret (%)	LoL	Relative regret (%)	LoL
greedy conical hull	3	3.8	-7	3.9	-7
	5	1.8	0	2.3	0
	7	0.1	0	0.1	0
	9	0.1	0	0.1	0
	11	0.1	0	0.1	0
	13	0.1	0	0.1	0
	15	0.1	0	0.1	0
	17	0.0	0	0.1	0
	19	0.0	0	0.1	0
	21	0.0	0	0.1	0
	23	0.0	0	0.1	0
	25	0.0	0	0.1	0
	27	0.0	0	0.1	0
	29	0.0	0	0.1	0
	31	0.0	0	0.1	0
	33	0.0	0	0.1	0
	35	0.0	0	0.1	0
	37	0.0	0	0.0	0
	39	0.0	0	0.0	0
	41	0.0	0	0.0	0
greedy convex hull	3	6.2	-7	6.2	-7
	5	0.2	0	0.3	0
	7	0.1	0	0.1	0
	9	0.1	0	0.1	0
	11	0.1	0	0.1	0
	13	0.1	0	0.1	0
	15	0.0	0	0.1	0
	17	0.0	0	0.0	0
	19	0.0	0	0.1	0
	21	0.0	0	0.1	0
	23	0.0	0	0.1	0
	25	0.0	0	0.1	0
	27	0.0	0	0.1	0
	29	0.0	0	0.1	0
	31	0.0	0	0.1	0

	33	0.0	0	0.1	0
	35	0.0	0	0.1	0
	37	0.0	0	0.1	0
	39	0.0	0	0.1	0
	41	0.0	0	0.0	0
***k*-Means**	3	163.1	686	130.4	531
	5	86.6	357	67.9	268
	7	44.2	190	37.4	125
	9	14.7	56	14.5	49
	11	1.7	10	1.7	10
	13	1.0	5	1.1	5
	15	0.0	0	0.1	0
	17	12.3	42	12.3	42
	19	0.0	0	0.0	0
	21	0.0	0	0.1	0
	23	0.0	0	0.0	0
	25	0.0	0	0.0	0
	27	0.0	0	0.0	0
	29	0.0	0	0.0	0
	31	0.0	0	0.0	0
	33	0.0	0	0.0	0
	35	0.0	0	0.0	0
	37	0.0	0	0.0	0
	39	0.0	0	0.0	0
	41	0.0	0	0.0	0
***k*-Medoids**	3	141.8	580	113.1	438
	5	86.6	361	68.4	287
	7	54.7	213	45.4	165
	9	12.9	47	12.9	49
	11	2.1	10	2.2	10
	13	1.0	5	1.1	5
	15	0.0	0	0.1	0
	17	12.1	42	12.1	40
	19	0.0	0	0.0	0
	21	0.0	0	0.1	0
	23	0.0	0	0.0	0
	25	0.0	0	0.0	0
	27	0.0	0	0.0	0
	29	0.0	0	0.0	0
	31	0.0	0	0.0	0
	33	0.0	0	0.0	0
	35	0.0	0	0.0	0
	37	0.0	0	0.0	0
	39	0.0	0	0.0	0
	41	0.0	0	0.0	0

Table B.1: Relative regret (%) and increase in time steps with loss of load (LoL) for the in-sample and out-of-sample experiments on the "Centered cluster" case study. Multiple combinations of the number of representative periods and methods for finding the representative periods are shown. Only the results for the squared euclidean distance are in the table.

Method	**Representatives**	**In-sample**		**Out-of-sample**	
		Relative regret (%)	**LoL**	**Relative regret (%)**	**LoL**
greedy convex hull	3	3.1	-7	3.2	-7
	5	0.6	0	0.7	0

	7	0.5	0	0.6	0
	9	0.2	0	0.2	0
	11	0.1	0	0.1	0
	13	0.1	0	0.0	0
	15	0.0	0	0.0	0
	17	0.0	0	0.0	0
	19	0.1	0	0.0	0
	21	0.1	0	0.0	0
	23	0.0	0	0.0	0
	25	0.0	0	0.0	0
	27	0.0	0	0.0	0
	29	0.0	0	0.0	0
	31	0.0	0	0.0	0
	33	0.0	0	0.0	0
	35	0.0	0	0.0	0
	37	0.0	0	0.0	0
	39	0.0	0	0.0	0
	41	0.0	0	0.0	0
***k*-Means**	3	143.1	622	114.5	480
	5	71.0	286	55.5	223
	7	50.6	192	41.5	136
	9	13.5	52	13.5	49
	11	1.7	10	1.7	10
	13	1.0	5	1.0	5
	15	0.0	0	0.1	0
	17	11.9	41	11.9	39
	19	0.0	0	0.0	0
	21	0.0	0	0.1	0
	23	0.0	0	0.0	0
	25	0.0	0	0.0	0
	27	0.0	0	0.0	0
	29	0.0	0	0.0	0
	31	0.0	0	0.0	0
	33	0.0	0	0.0	0
	35	0.0	0	0.0	0
	37	0.0	0	0.0	0
	39	0.0	0	0.0	0
	41	0.0	0	0.0	0
***k*-Medoids**	3	92.1	424	72.7	295
	5	56.3	235	46.8	179
	7	26.7	101	26.4	96
	9	25.8	91	25.8	84
	11	0.0	0	0.1	0
	13	0.0	0	0.1	0
	15	0.0	0	0.1	0
	17	0.0	0	0.1	0
	19	0.0	0	0.1	0
	21	0.0	0	0.0	0
	23	0.0	0	0.0	0
	25	0.0	0	0.0	0
	27	0.0	0	0.0	0
	29	0.0	0	0.0	0
	31	0.0	0	0.0	0
	33	0.0	0	0.0	0
	35	0.0	0	0.0	0
	37	0.0	0	0.0	0

	39	0.0	0	0.0	0
	41	0.0	0	0.0	0

Table B.2: Relative regret (%) and increase in time steps with loss of load (LoL) for the in-sample and out-of-sample experiments on the "Convex" case study. Multiple combinations of the number of representative periods and methods for finding the representative periods are shown. Only the results for the cosine distance are in the table.

Method	**Representatives**	**In-sample**		**Out-of-sample**	
		Relative regret (%)	**LoL**	**Relative regret (%)**	**LoL**
greedy conical hull	3	8.7	-7	8.3	-7
	5	1.0	0	0.8	0
	7	0.9	0	0.7	0
	9	0.8	0	0.7	0
	11	0.5	0	0.4	0
	13	0.4	0	0.3	0
	15	0.3	0	0.2	0
	17	0.3	0	0.2	0
	19	0.1	0	0.1	0
	21	0.1	0	0.1	0
	23	0.1	0	0.1	0
	25	0.1	0	0.1	0
	27	0.1	0	0.1	0
	29	0.1	0	0.1	0
	31	0.1	0	0.0	0
	33	0.0	0	0.0	0
	35	0.0	0	0.0	0
	37	0.0	0	0.1	0
	39	0.0	0	0.1	0
	41	0.0	0	0.1	0
greedy convex hull	3	13.1	11	12.3	9
	5	1.0	0	1.1	0
	7	0.7	0	0.9	0
	9	0.6	0	0.6	0
	11	0.5	0	0.6	0
	13	0.4	0	0.5	0
	15	0.4	0	0.4	0
	17	0.2	0	0.3	0
	19	0.2	0	0.4	0
	21	0.2	0	0.4	0
	23	0.1	0	0.3	0
	25	0.1	0	0.3	0
	27	0.1	0	0.3	0
	29	0.1	0	0.3	0
	31	0.1	0	0.3	0
	33	0.1	0	0.2	0
	35	0.1	0	0.2	0
	37	0.1	0	0.2	0
	39	0.0	0	0.1	0
	41	0.0	0	0.1	0
***k*-Means**	3	301.1	1608	329.7	1911
	5	280.0	1283	309.5	1628
	7	180.8	682	212.6	803
	9	10.9	82	13.4	111
	11	10.8	81	13.2	110
	13	116.4	343	124.5	382

	15	0.3	0	0.4	0
	17	0.3	0	0.4	0
	19	0.2	0	0.3	0
	21	0.2	0	0.2	0
	23	0.2	0	0.2	0
	25	0.1	0	0.2	0
	27	0.1	0	0.2	0
	29	0.1	0	0.2	0
	31	0.1	0	0.2	0
	33	0.1	0	0.2	0
	35	0.1	0	0.2	0
	37	0.1	0	0.1	0
	39	0.1	0	0.1	0
	41	0.1	0	0.1	0
***k*-Medoids**	3	202.2	888	233.3	1051
	5	196.4	859	215.8	998
	7	159.0	565	192.4	808
	9	10.8	80	13.1	109
	11	10.8	82	13.2	111
	13	120.9	328	128.5	385
	15	0.2	0	0.2	0
	17	2.4	15	2.3	20
	19	4.5	31	4.4	40
	21	0.1	0	0.1	0
	23	0.1	0	0.1	0
	25	0.1	0	0.1	0
	27	0.1	0	0.1	0
	29	0.1	0	0.1	0
	31	0.0	0	0.1	0
	33	0.1	0	0.1	0
	35	0.0	0	0.1	0
	37	0.1	0	0.1	0
	39	0.0	0	0.1	0
	41	0.0	0	0.1	0

Table B.3: Relative regret (%) and increase in time steps with loss of load (LoL) for the in-sample and out-of-sample experiments on the "Centered cluster" case study. Multiple combinations of the number of representative periods and methods for finding the representative periods are shown. Only the results for the squared euclidean distance are in the table.

Method	**Representatives**	**In-sample**		**Out-of-sample**	
		Relative regret (%)	**LoL**	**Relative regret (%)**	**LoL**
greedy convex hull	3	6.0	-7	5.7	-7
	5	0.5	0	0.5	0
	7	0.5	0	0.6	0
	9	0.3	0	0.5	0
	11	0.1	0	0.3	0
	13	0.3	3	0.5	3
	15	0.2	0	0.5	0
	17	0.2	0	0.3	0
	19	0.2	0	0.4	0
	21	0.1	0	0.3	0
	23	0.1	0	0.3	0
	25	0.1	0	0.3	0
	27	0.1	0	0.2	0

	29	0.1	0	0.2	0
	31	0.1	0	0.2	0
	33	0.1	0	0.2	0
	35	0.0	0	0.1	0
	37	0.0	0	0.1	0
	39	0.0	0	0.1	0
	41	0.0	0	0.1	0
***k*-Means**	3	309.4	1624	341.5	1985
	5	246.6	1179	283.2	1422
	7	184.3	733	212.6	877
	9	10.8	81	13.2	110
	11	10.7	80	13.1	109
	13	111.6	291	118.8	341
	15	0.3	0	0.4	0
	17	0.3	0	0.4	0
	19	0.3	0	0.3	0
	21	0.2	0	0.3	0
	23	0.2	0	0.2	0
	25	0.2	0	0.2	0
	27	0.1	0	0.2	0
	29	0.1	0	0.2	0
	31	0.1	0	0.2	0
	33	0.1	0	0.2	0
	35	0.1	0	0.2	0
	37	0.1	0	0.2	0
	39	0.1	0	0.1	0
	41	0.1	0	0.1	0
***k*-Medoids**	3	211.6	973	244.5	1122
	5	186.7	752	204.2	903
	7	152.4	583	166.5	654
	9	131.7	441	151.8	548
	11	125.8	374	143.2	474
	13	0.2	0	0.2	0
	15	52.9	159	55.3	181
	17	2.9	23	3.0	25
	19	0.3	0	0.4	0
	21	0.1	0	0.1	0
	23	0.1	0	0.1	0
	25	2.5	17	2.5	23
	27	0.1	0	0.1	0
	29	0.1	0	0.1	0
	31	0.0	0	0.1	0
	33	0.0	0	0.1	0
	35	0.0	0	0.1	0
	37	0.0	0	0.1	0
	39	0.0	0	0.1	0
	41	0.0	0	0.1	0

Table B.4: Relative regret (%) and increase in time steps with loss of load (LoL) for the in-sample and out-of-sample experiments on the "Centered cluster" case study. Multiple combinations of the number of representative periods and methods for finding the representative periods are shown. Only the results for the cosine distance are in the table.

Method	**Representatives**	**In-sample**		**Out-of-sample**	
		Relative regret (%)	**LoL**	**Relative regret (%)**	**LoL**

greedy conical hull	3	52.2	480	103.7	618
	5	50.8	444	93.2	568
	7	8.7	107	40.9	226
	9	8.4	117	43.7	242
	11	4.8	48	40.5	161
	13	4.8	30	39.6	151
	15	4.7	39	39.9	152
	17	4.2	36	40.5	164
	19	3.6	34	41.4	159
	21	1.7	11	35.6	141
	23	1.9	9	35.9	141
	25	1.7	19	35.8	155
	27	1.7	16	35.8	149
	29	0.5	-2	35.3	137
	31	0.5	20	34.2	168
	33	0.5	19	34.1	171
	35	0.2	13	28.6	149
	37	0.2	13	28.7	149
	39	0.0	3	29.0	132
	41	0.0	3	29.3	132
greedy convex hull	3	50.8	301	70.1	436
	5	2.3	-4	35.9	123
	7	1.0	-9	30.4	109
	9	0.7	-10	30.6	109
	11	0.4	-9	30.8	123
	13	0.6	-9	25.6	110
	15	0.6	-7	26.5	129
	17	0.6	-8	25.7	114
	19	0.6	-7	25.1	108
	21	0.6	-8	24.9	119
	23	0.3	-2	31.0	137
	25	0.2	-2	31.8	136
	27	0.2	-2	30.8	138
	29	0.2	-2	30.7	123
	31	0.2	-2	28.8	121
	33	0.2	-2	28.9	127
	35	0.1	0	28.0	129
	37	0.1	0	27.7	132
	39	0.1	-1	28.4	130
	41	0.1	-1	28.6	133
***k*-Means**	3	194.0	1358	238.2	1365
	5	190.9	1269	234.4	1270
	7	61.0	492	108.0	604
	9	59.9	470	106.1	573
	11	28.7	254	73.6	367
	13	25.5	227	70.4	357
	15	6.7	98	47.3	240
	17	8.3	106	50.1	237
	19	8.4	102	50.3	244
	21	6.2	86	47.3	222
	23	5.3	74	45.8	211
	25	5.5	78	45.8	211
	27	3.7	62	42.6	199
	29	2.0	40	38.3	169
	31	1.8	36	38.5	172

	33	2.5	52	40.3	180
	35	2.0	37	39.7	179
	37	1.0	24	37.1	167
	39	1.6	33	38.9	174
	41	1.4	29	38.7	173
***k*-Medoids**	3	146.3	1025	192.7	1086
	5	146.9	975	192.8	1074
	7	53.3	432	99.5	544
	9	54.1	430	99.9	548
	11	50.2	403	96.3	516
	13	8.9	116	46.2	242
	15	30.2	260	75.9	384
	17	9.8	102	47.4	231
	19	8.6	106	45.5	237
	21	32.9	265	79.6	406
	23	3.2	58	38.6	194
	25	1.3	35	34.9	171
	27	0.9	30	35.2	176
	29	0.8	26	34.9	166
	31	1.7	37	35.2	172
	33	2.7	48	36.4	180
	35	0.4	14	32.2	157
	37	0.1	7	29.3	144
	39	1.1	24	35.1	160
	41	0.4	13	34.5	156

Table B.5: Relative regret (%) and increase in time steps with loss of load (LoL) for the in-sample and out-of-sample experiments on the "Separate cluster with spatial correlation" case study. Multiple combinations of the number of representative periods and methods for finding the representative periods are shown. Only the results for the squared euclidean distance are in the table.

Method	**Representatives**	**In-sample**		**Out-of-sample**	
		Relative regret (%)	**LoL**	**Relative regret (%)**	**LoL**
greedy convex hull	3	14.7	164	46.2	308
	5	14.6	162	40.6	290
	7	14.0	159	40.3	286
	9	0.3	-4	27.2	125
	11	0.2	-4	30.6	128
	13	0.2	-2	30.3	142
	15	0.1	0	29.6	141
	17	0.1	2	28.6	129
	19	0.1	1	28.2	133
	21	0.1	0	28.0	144
	23	0.1	0	28.7	147
	25	0.1	-2	28.3	127
	27	0.1	-2	29.5	136
	29	0.1	-1	29.5	140
	31	0.1	-1	29.0	141
	33	0.1	-1	29.1	133
	35	0.1	-1	29.2	140
	37	0.1	-1	28.2	127
	39	0.1	-1	28.1	124
	41	0.1	-1	28.2	127
***k*-Means**	3	194.0	1358	238.2	1365
	5	190.8	1287	234.3	1292

	7	41.3	348	86.5	467
	9	41.5	335	87.3	461
	11	15.7	173	59.3	308
	13	16.5	186	60.5	317
	15	9.0	118	51.8	258
	17	8.7	103	50.6	248
	19	11.2	129	55.0	262
	21	6.5	96	47.7	239
	23	3.4	62	42.1	197
	25	5.5	76	45.8	209
	27	3.0	53	40.1	183
	29	2.0	39	40.0	185
	31	1.6	34	38.6	177
	33	2.7	51	41.4	193
	35	2.0	38	39.8	181
	37	1.7	35	39.4	179
	39	1.6	35	39.2	179
	41	1.7	31	39.3	182
***k*-Medoids**	3	142.5	910	189.6	1026
	5	52.7	453	100.2	563
	7	5.2	98	38.9	239
	9	51.3	403	97.4	523
	11	2.8	68	37.1	210
	13	1.1	34	34.7	180
	15	1.0	33	34.1	166
	17	0.6	19	34.1	160
	19	0.4	16	28.9	148
	21	0.5	14	34.3	160
	23	0.3	10	28.6	142
	25	0.1	4	29.1	137
	27	0.2	8	28.4	151
	29	0.1	3	28.7	137
	31	0.1	4	28.9	139
	33	0.2	7	31.6	139
	35	0.1	4	28.6	137
	37	0.1	3	28.5	135
	39	0.0	3	28.9	138
	41	0.0	1	28.6	133

Table B.6: Relative regret (%) and increase in time steps with loss of load (LoL) for the in-sample and out-of-sample experiments on the "Separate clusters with spatial correlation" case study. Multiple combinations of the number of representative periods and methods for finding the representative periods are shown. Only the results for the cosine distance are in the table.

Method	**Representatives**	**In-sample**		**Out-of-sample**	
		Relative regret (%)	**LoL**	**Relative regret (%)**	**LoL**
greedy conical hull	3	78.8	431	83.6	497
	5	79.4	394	82.8	416
	7	83.4	453	87.4	512
	9	82.8	412	85.7	435
	11	28.9	254	26.1	264
	13	28.3	256	25.5	260
	15	29.6	261	26.8	272
	17	27.6	247	25.2	251
	19	22.8	195	21.4	201

	21	5.1	84	8.9	99
	23	5.0	84	8.9	99
	25	5.0	82	8.9	98
	27	5.0	79	8.3	88
	29	3.7	66	7.8	80
	31	3.9	66	7.8	83
	33	4.1	68	7.9	86
	35	3.7	62	7.1	81
	37	0.7	21	3.2	46
	39	0.6	17	2.9	40
	41	0.9	19	3.9	43
greedy convex hull	3	118.5	710	105.6	694
	5	100.8	611	82.9	564
	7	109.3	592	91.9	519
	9	108.6	567	91.2	532
	11	114.8	566	98.2	544
	13	120.5	607	104.4	578
	15	85.4	436	74.3	466
	17	86.4	413	74.3	434
	19	86.6	441	74.4	468
	21	85.9	456	73.4	449
	23	86.0	454	73.4	463
	25	79.4	428	65.6	423
	27	80.8	391	67.7	403
	29	54.1	335	45.0	365
	31	53.9	361	45.1	363
	33	54.3	364	45.4	354
	35	3.4	59	3.6	78
	37	3.4	59	3.4	59
	39	3.0	53	3.2	70
	41	3.1	54	3.0	51
***k*-Means**	3	89.4	748	77.8	734
	5	85.6	690	73.7	635
	7	64.8	596	55.2	542
	9	59.4	518	49.8	478
	11	65.2	567	54.8	532
	13	42.5	426	34.9	372
	15	38.9	365	32.2	343
	17	30.9	320	27.5	303
	19	33.7	358	29.6	328
	21	29.9	316	24.2	324
	23	33.9	324	27.2	316
	25	21.0	224	16.7	220
	27	18.9	202	16.1	193
	29	16.8	186	13.6	181
	31	16.2	186	13.7	196
	33	10.5	139	9.9	145
	35	11.9	151	10.3	143
	37	9.8	121	10.0	133
	39	15.3	163	12.3	186
	41	9.9	116	6.5	109
***k*-Medoids**	3	83.1	694	69.7	655
	5	85.9	693	71.5	606
	7	74.6	623	60.9	587
	9	64.6	543	54.4	534
	11	56.1	483	49.0	465

	13	48.0	461	37.6	374
	15	48.2	477	41.3	462
	17	33.7	320	30.9	347
	19	30.3	343	26.1	341
	21	37.1	378	30.9	359
	23	32.3	344	25.0	311
	25	17.5	217	16.9	248
	27	15.6	155	11.5	164
	29	15.6	179	12.8	185
	31	10.6	139	9.8	144
	33	7.0	96	8.5	108
	35	6.6	87	9.3	110
	37	9.4	121	8.4	128
	39	10.9	131	9.6	133
	41	6.3	76	5.3	83

Table B.7: Relative regret (%) and increase in time steps with loss of load (LoL) for the in-sample and out-of-sample experiments on the "Separate cluster without spatial correlation" case study. Multiple combinations of the number of representative periods and methods for finding the representative periods are shown. Only the results for the squared euclidean distance are in the table.

Method	**Representatives**	**In-sample**		**Out-of-sample**	
		Relative regret (%)	**LoL**	**Relative regret (%)**	**LoL**
greedy convex hull	3	120.5	815	108.5	768
	5	95.4	694	86.4	658
	7	59.6	459	49.2	448
	9	62.2	507	49.7	476
	11	14.2	175	15.5	164
	13	15.4	197	17.1	225
	15	14.1	176	14.0	176
	17	14.1	171	14.0	181
	19	13.7	182	14.1	176
	21	11.0	146	10.7	158
	23	7.2	85	8.3	94
	25	7.2	94	8.5	93
	27	7.5	93	8.3	103
	29	7.4	95	8.3	105
	31	7.7	97	9.3	123
	33	8.1	100	9.3	127
	35	8.1	103	9.3	123
	37	8.2	103	9.4	125
	39	7.0	79	6.9	106
	41	5.0	57	5.3	83
***k*-Means**	3	89.3	747	77.7	733
	5	87.1	714	75.1	654
	7	70.1	656	58.9	605
	9	55.6	532	46.4	498
	11	63.1	563	54.1	540
	13	39.2	405	32.7	374
	15	37.2	365	31.5	367
	17	30.7	318	26.1	293
	19	34.6	372	28.8	341
	21	31.1	321	24.2	307
	23	31.5	312	25.7	293
	25	19.4	199	15.4	189

	27	19.8	202	16.3	198
	29	15.8	175	12.5	163
	31	15.6	178	12.4	187
	33	12.7	170	12.1	170
	35	11.9	152	10.3	164
	37	9.7	119	9.1	145
	39	15.2	166	12.2	179
	41	9.4	113	6.9	111
***k*-Medoids**	3	83.0	685	69.6	655
	5	81.0	653	67.9	575
	7	55.7	532	49.8	528
	9	52.0	499	46.1	501
	11	38.7	379	35.2	391
	13	27.7	262	22.9	262
	15	28.4	320	26.8	319
	17	13.1	159	14.2	159
	19	10.9	150	10.7	161
	21	14.3	160	11.6	163
	23	11.5	121	10.2	141
	25	12.3	152	11.7	161
	27	10.9	141	9.8	151
	29	5.8	70	6.2	81
	31	5.7	73	6.2	85
	33	3.0	47	4.3	62
	35	4.6	55	5.0	72
	37	4.4	64	6.3	86
	39	4.3	55	4.8	62
	41	1.2	25	2.9	37

Table B.8: Relative regret (%) and increase in time steps with loss of load (LoL) for the in-sample and out-of-sample experiments on the "Separate clusters without spatial correlation" case study. Multiple combinations of the number of representative periods and methods for finding the representative periods are shown. Only the results for the cosine distance are in the table.

C

European out-of-sample evaluation results

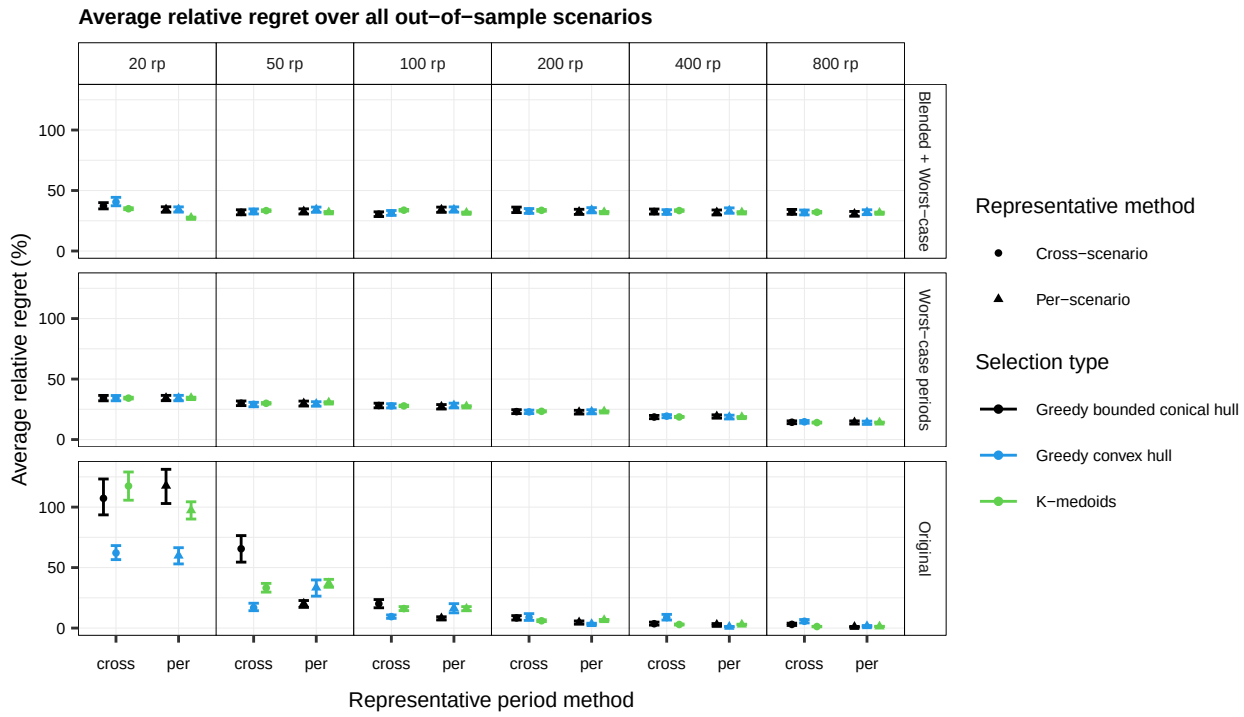

Figure C.1: Average relative regret (%) over all out-of-sample scenarios under different settings. The combination of colour and shape indicates the method used. The error bars show the 95% confidence intervals for the mean, based on bootstrapping.

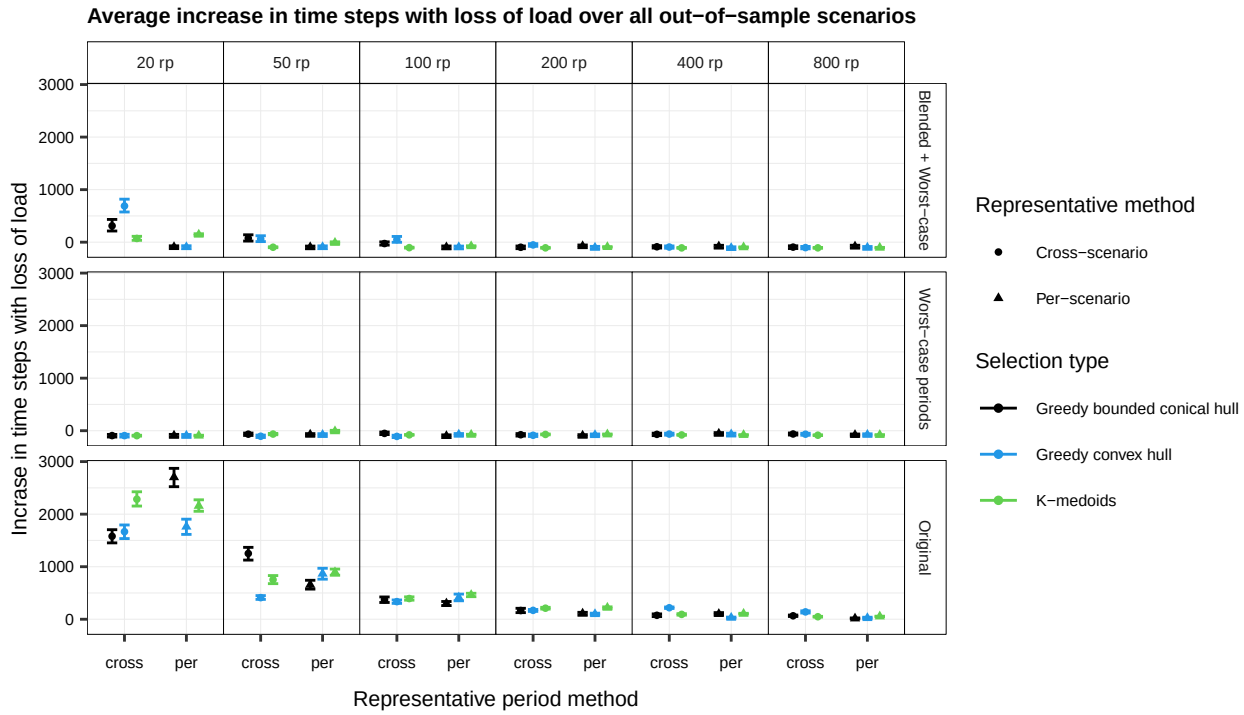

Figure C.2: Average increase in time steps with loss of load over all out-of-sample scenarios under different settings. The combination of colour and shape indicates the method used. The error bars show the 95% confidence intervals for the mean, based on bootstrapping.

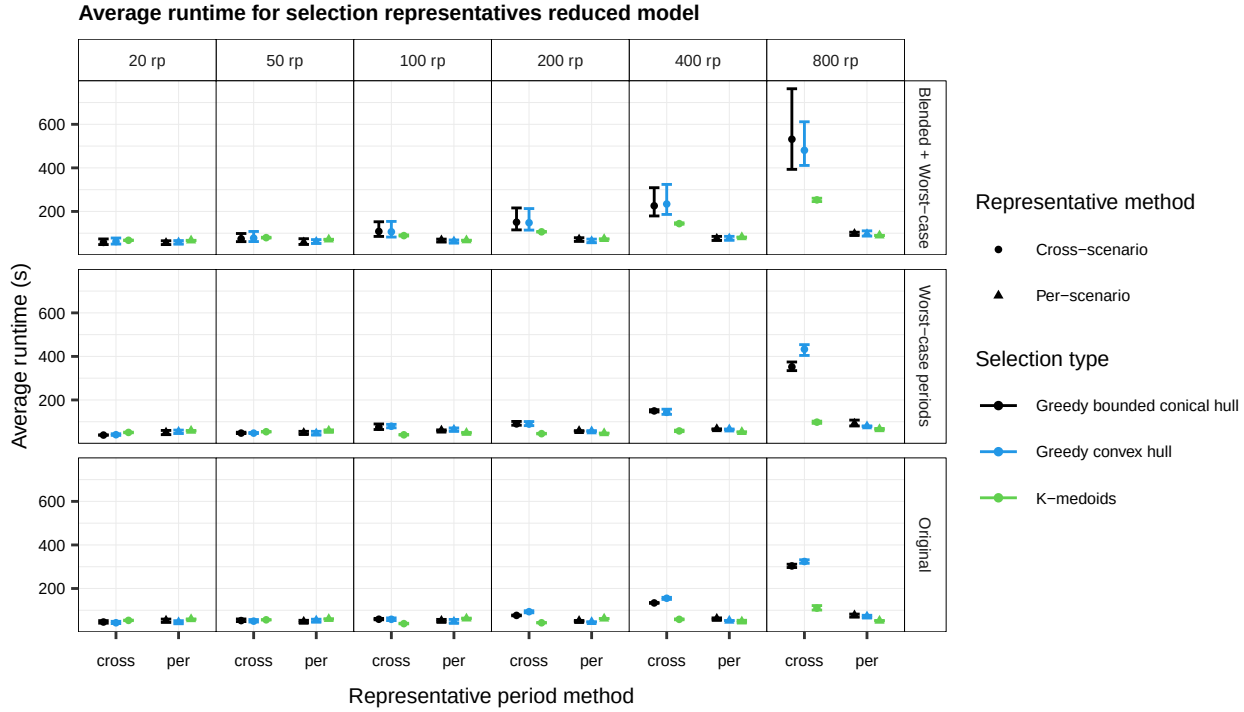

Figure C.3: Average runtime spent finding representatives for the reduced model. The combination of colour and shape indicates the method used. The error bars show the 95% confidence intervals for the mean, based on bootstrapping.

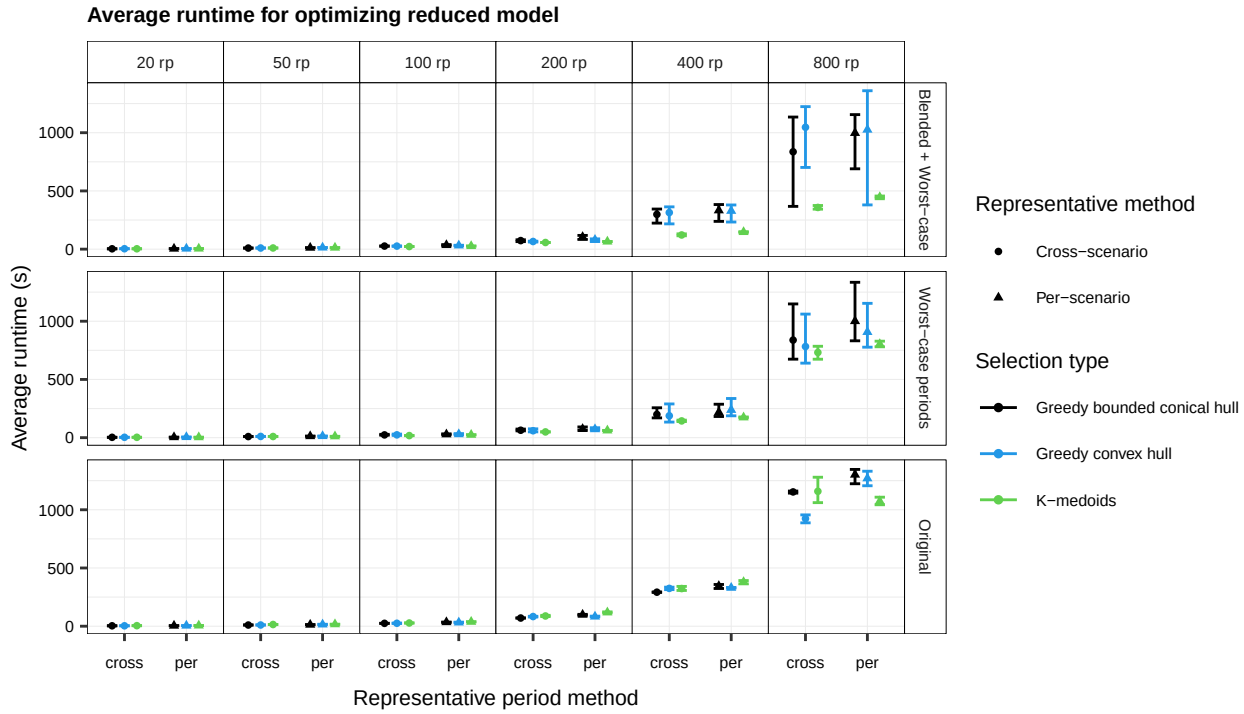

Figure C.4: Average total runtime spent on optimizing the reduced model. The combination of colour and shape indicates the method used. The error bars show the 95% confidence intervals for the mean, based on bootstrapping.

D

Code and input data

As part of this thesis, the code implementing all methods is publicly available on GitHub [15]. Due to file size limitations, some data files (both input and output) are stored in two separate online data repositories as zipped files [16, 17]. Instructions for downloading these files are provided in the GitHub repository, and direct links to both the code and data repositories are listed in the bibliography.

The hardware specifications for the laptop on which all experiments were run are as follows: AMD Ryzen 7 PRO 5850U processor, 8 cores, 16 threads, 1.90 GHz base clock speed, integrated Radeon Graphics. 16 GB RAM. Windows 11 Enterprise 24H2, 64-bit. Only the full stochastic problem for the European case study was ran on a separate TNO computer due to storage limitations of the laptop.

Bibliography

[1] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer, 2001.

[2] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[3] Tom Brown, Marta Victoria, Elisabeth Zeyen, Fabian Hofmann, Fabian Neumann, Martha Frysztacki, Johannes Hampp, David Schlachtberger, Jonas Hörsch, Amos Schledorn, Caspar Schauß, Koen van Greevenbroek, Markus Millinger, Philipp Glaum, Bobby Xiong, and Toni Seibold. PyPSA-Eur: An open sector-coupled optimisation model of the European energy system.

[4] Kenneth L Clarkson. Applications of random sampling in computational geometry, ii. In *Proceedings of the fourth annual symposium on Computational geometry*, pages 1–11, 1988.

[5] Laurent Condat. Fast projection onto the simplex and the l 1 ball. *Mathematical Programming*, 158(1):575–585, 2016.

[6] Shahab Dehghan, Nima Amjady, and Ahad Kazemi. Two-stage robust generation expansion planning: A mixed integer linear programming model. *IEEE Transactions on Power Systems*, 29(2):584–597, 2013.

[7] ENTSO-E. TYNDP 2022 Scenarios. `https://2022.entsos-tyndp-scenarios.eu/download/`, 2022. Accessed: 2025-04-24.

[8] ENTSO-E. TYNDP 2024 Scenarios Report. `https://tyndp.entsoe.eu/resources/tyndp2024-scenarios-report`, 2024. Accessed: 2025-05-03.

[9] William Gandulfo, Esteban Gil, and Ignacio Aravena. Generation capacity expansion planning under demand uncertainty using stochastic mixed-integer programming. In *2014 IEEE PES general meeting| conference & exposition*, pages 1–5. IEEE, 2014.

[10] Zhi Gao. Input data files for fully flexible temporal resolutions, 2025. Available at `https://github.com/gzclarence/Fully-Flexible-Temporal-Resolution`.

[11] Zhi Gao, Matteo Gazzani, Diego A. Tejada-Arango, Abel Siqueira, Ni Wang, Madeleine Gibescu, and G. Morales-España. Fully flexible temporal resolution for energy system optimization. *SSRN preprint*, 2024. Available at `https://ssrn.com/abstract=5214263`, `http://dx.doi.org/10.2139/ssrn.5214263`.

[12] Esteban Gil, Ignacio Aravena, and Raúl Cárdenas. Generation capacity expansion planning under hydro uncertainty using stochastic mixed integer programming and scenario reduction. *IEEE Transactions on Power Systems*, 30(4):1838–1847, 2014.

[13] Nikolaos E Koltsaklis and Athanasios S Dagoumas. State-of-the-art generation expansion planning: A review. *Applied energy*, 230:563–589, 2018.

[14] Leander Kotzur, Peter Markewitz, Martin Robinius, and Detlef Stolten. Impact of different time series aggregation methods on optimal energy system design. *Renewable energy*, 117:474–487, 2018.

[15] Lotte Kremer. generation-expansion-planning [code implementation]. `https://doi.org/10.5281/zenodo.15638708`, 2025. Accessed: 2025-06-11.

[16] Lotte Kremer. Input and output data thesis - part 1 [data set]. `https://doi.org/10.5281/zenodo.15584244`, 2025. Accessed: 2025-06-11.

[17] Lotte Kremer. Input and output data thesis - part 2 [data set]. `https://doi.org/10.5281/zenodo.15637403`, 2025. Accessed: 2025-06-11.

[18] Can Li, Antonio J Conejo, John D Siirola, and Ignacio E Grossmann. On representative day selection for capacity expansion planning of power systems under extreme operating conditions. *International Journal of Electrical Power & Energy Systems*, 137:107697, 2022.

[19] Dharik S Mallapragada, Dimitri J Papageorgiou, Aranya Venkatesh, Cristiana L Lara, and Ignacio E Grossmann. Impact of model resolution on scenario outcomes for electricity sector system expansion. *Energy*, 163:1231–1244, 2018.

[20] Giovanni Micheli, Maria Teresa Vespucci, Marco Stabile, Cinzia Puglisi, and Andres Ramos. A two-stage stochastic milp model for generation and transmission expansion planning with high shares of renewables. *Energy Systems*, pages 1–43, 2020.

[21] Paul Nahmmacher, Eva Schmid, Lion Hirth, and Brigitte Knopf. Carpe diem: A novel approach to select representative days for long-term power system modeling. *Energy*, 112:430–442, 2016.

[22] Netherlands Authority for Consumers and Markets. The value of lost load for electricity in the netherlands. Report, Netherlands Authority for Consumers and Markets, 2022. URL `https://www.acm.nl/system/files/documents/bijlage-bij-besluit-rapport-the-value-of-lost-load-for-electricity-in-the-netherlands.pdf`. Accessed: 2025-05-27.

[23] Grigory Neustroev, Diego A. Tejada-Arango, Lauren Clisby, and Germán Morales-España. Tulipa Clustering. URL `https://github.com/TulipaEnergy/TulipaClustering.jl`.

[24] Heejung Park and Ross Baldick. Multi-year stochastic generation capacity expansion planning under environmental energy policy. *Applied energy*, 183:737–745, 2016.

[25] Shruthi Patil, Leander Kotzur, and Detlef Stolten. Advanced spatial and technological aggregation scheme for energy system models. *Energies*, 15(24):9517, 2022.

[26] Stefan Pfenninger. Dealing with multiple decades of hourly wind and pv time series in energy models: A comparison of methods to reduce time resolution and the planning implications of inter-annual variability. *Applied energy*, 197:1–13, 2017.

[27] Stefan Pfenninger, Adam Hawkes, and James Keirstead. Energy systems modeling for twenty-first century energy challenges. *Renewable and sustainable energy reviews*, 33:74–86, 2014.

[28] Kris Poncelet, Hanspeter Höschle, Erik Delarue, Ana Virag, and William D'haeseleer. Selecting representative days for capturing the implications of integrating intermittent renewables in generation expansion planning problems. *IEEE Transactions on Power Systems*, 32(3):1936–1948, 2016.

[29] Line A Roald, David Pozo, Anthony Papavasiliou, Daniel K Molzahn, Jalal Kazempour, and Antonio Conejo. Power systems optimization under uncertainty: A review of methods and applications. *Electric Power Systems Research*, 214:108725, 2023.

[30] Ian J Scott, Pedro MS Carvalho, Audun Botterud, and Carlos A Silva. Clustering representative days for power systems generation expansion planning: Capturing the effects of variable renewables and energy storage. *Applied Energy*, 253:113603, 2019.

[31] Ian J Scott, Pedro MS Carvalho, Audun Botterud, and Carlos A Silva. Long-term uncertainties in generation expansion planning: Implications for electricity market modelling and policy. *Energy*, 227:120371, 2021.

[32] Ebrahim Shayesteh, Benjamin F Hobbs, and Mikael Amelin. Scenario reduction, network aggregation, and dc linearisation: which simplifications matter most in operations and planning optimisation? *IET Generation, Transmission & Distribution*, 10(11):2748–2755, 2016.

[33] Holger Teichgraeber and Adam R Brandt. Clustering methods to find representative periods for the optimization of energy systems: An initial framework and comparison. *Applied energy*, 239:1283–1293, 2019.

[34] Holger Teichgraeber and Adam R Brandt. Time-series aggregation for the optimization of energy systems: Goals, challenges, approaches, and opportunities. *Renewable and Sustainable Energy Reviews*, 157:111984, 2022.

[35] Diego A Tejada-Arango, Maya Domeshek, Sonja Wogrin, and Efraim Centeno. Enhanced representative days and system states modeling for energy storage investment analysis. *IEEE Transactions on Power Systems*, 33(6): 6534–6544, 2018.

[36] Diego A Tejada-Arango, German Morales-Espana, and Juha Kiviluoma. Speeding-up large-scale lp energy system models: Using graph-theory to remove the overhead cost of flexible modeling. *arXiv preprint arXiv:2407.05451*, 2024.

[37] Sonja Wogrin, Pablo Duenas, Andrés Delgadillo, and Javier Reneses. A new approach to model load levels in electric power systems with high renewable penetration. *IEEE Transactions on Power Systems*, 29(5):2210–2218, 2014.

[38] Shengfei Yin and Jianhui Wang. Generation and transmission expansion planning towards a 100% renewable future. *IEEE Transactions on Power Systems*, 37(4):3274–3285, 2020.

[39] Marilena Zampara, Daniel Ávila, and Anthony Papavasiliou. Capacity expansion planning under uncertainty subject to expected energy not served constraints. *arXiv preprint arXiv:2501.17484*, 2025.