Real Worst-Case Period Selection in Time Series Aggregation for Energy System Planning

An Experimental Comparison Using the Tulipa Energy Model

Naman Choudhary[1]
Supervisor(s): Germán Morales España[1]**, Maaike Elgersma**[1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Naman Choudhary
Final project course: CSE3000 Research Project
Thesis committee: Germán Morales España, Maaike Elgersma, Jasmijn Baaijens

Abstract

Large-scale energy system planning requires solving capacity expansion problems over complete hourly time series, which is computationally intractable. Selecting a small set of representative periods compresses the input, but standard clustering methods miss rare extreme events and smooth over many local peaks, producing investment plans that underestimate the true optimal cost. A construction from Elgersma [5] addresses this by building an artificial worst-case period, but the result can be more extreme than any historically observed day. This work investigates whether selecting a real observed period to represent the worst case can match this quality while preserving the temporal coherence of each selected day, meaning its demand and availability values reflect conditions that genuinely occurred together in the same 24-hour window. Three real-period selection strategies are proposed and compared against two artificial worst-case variants and the standard k-medoids baseline using the Tulipa Energy Model [4]. All non-fractional methods approach near-zero regret by approximately $k = 400$ periods; no single method is consistently fastest, and the differences between methods become negligible beyond that point. No real worst-case method consistently improves reliability over plain k-medoids, but this result is explained by a dataset calibration issue rather than a fundamental failure of the approach. Two lessons emerge regardless of the dataset: global weight scaling in the fractional-weight variant introduces a persistent non-zero regret plateau that cannot be corrected by adding more periods, and model-guided period selection doubles computation cost without benefit when the planning model is not sensitive to reliability shortfalls. The weight-scaling issue is structural, and the computational cost of the model-guided method follows directly from its two-solve design. A realistically calibrated dataset is still needed to fully evaluate the reliability benefit of real worst-case selection.

1 Introduction

Across Europe, governments and network operators must make large-scale decisions about which energy assets to develop and their appropriate capacities to ensure reliable demand fulfilment in the coming decades [10]. These decisions are represented as capacity expansion problems: linear programmes that find the cheapest investment plan that satisfies demand at every hour. The computational difficulty of these problems grows directly with the number of time steps in the model. A single year of hourly data already implies 8760 time steps [14], and practical planning models must consider multiple years and multiple climate scenarios simultaneously, making a full-resolution solve intractable.

The standard approach is to compress the input time series by selecting a small set of representative periods that represent a compressed version of the full year of hourly profiles [14]. This reduces the number of variables and constraints dramatically, making the problem solvable. However, clustering methods used to select these periods, such as k-means and k-medoids, smooth out rare but essential extreme events. A day with very high demand and very low renewable output is unlikely to become a cluster centroid or medoid, because it resembles too few other days. Recent work shows formally that the investment plan produced by standard clustering is a lower bound on the optimal cost of the full model [13]: It looks cheaper than the full-resolution plan because the worst conditions are never represented. In practice, this plan may be infeasible or severely suboptimal when re-evaluated against the complete time series.

Several approaches address this gap. Li et al. [9] propose selecting representative days that explicitly include extreme net-load conditions, using both input-based and cost-based criteria, and show that doing so tightens the gap between the reduced model and the full model. Yeganefar et al. [18] show that incorporating the extreme days of the net-load signal into the representative set prevents underestimating the dispatchable power needed to cover periods of high demand and low

renewable output. Teichgraeber et al. [16, 15] present a framework for including extreme events in representative period selection, with an iterative extension that uses slack variables to detect reliability shortfalls and adds extreme periods until a feasible solution is found. Hilbers et al. [7] take a different direction and propose a posteriori selection: aggregation is adapted to the specific planning model by identifying which periods cause the largest reliability shortfall in model output. In an unreleased paper from Elgersma [5], has proven that building an artificial period through combining peak demand with the worst availability-to-demand ratio across all real periods provides a feasible investment plan. However, all of these approaches either construct synthetic periods that can be more extreme than any day that has historically occurred, or iterate by running one model solve per added period and repeating until no reliability shortfall remains, which multiplies the computational cost with the number of extreme periods required. The question of whether selecting a real, historically observed period to represent the worst case can achieve comparable quality at lower cost has not yet been studied.

This work addresses that gap directly. Three strategies are proposed for selecting real extreme periods from the historical data: one ranks periods by a capacity-weighted net-load score; one applies the Elgersma scoring logic to real periods to find the one most similar in character to the artificial construction; and one is guided by the solved model's own energy-not-served (ENS) output to identify the representative period with the highest total ENS and then selects the most demanding real period among those originally mapped to it, but with the downside that the model has to be solved twice. All three are compared against the Elgersma artificial worst-case variants and the standard k-medoids baseline, using the Tulipa Energy Model on a three-scenario Tulipa energy dataset.

This work has three main contributions beyond the experimental results: First, it provides the first systematic comparison of real-data period selection against artificial worst-case construction on a common benchmark. Second, it identifies a structural failure in global weight scaling that produces a persistent non-zero regret plateau regardless of how many representative periods are added, and shows that this failure is not dataset-specific. Third, it quantifies the computational cost of model-guided period selection and establishes the conditions under which that cost is not justified.

The remainder of this paper is structured as follows. Section 2 formalises the problem and describes the bound properties of existing methods. Section 3 describes all six methods. Section 4 presents the experiments and results. Section 5 presents the conclusions and future work. The appendix contains full-range plots and the responsible research statement.

2 Representative Period Selection in Energy System Planning

2.1 Time series aggregation procedure

The general process of selecting representative periods follows four steps [5]:

1. **Choose** the number k of representative periods and the time frame they represent.
2. **Select** which k periods will serve as representatives.
3. **Assign weights**, mapping each original period to a representative, and computing how many original periods each representative period stands for.
4. **Solve** the reduced investment model built from the representatives and their weights.

All six methods compared in this work use the same number of periods k, the same period length of 24 hours, and the same Tulipa energy model in step 4. They mainly differ in step 2, namely in how the representative periods are chosen. The artificial variants also differ in how the appended worst-case period is weighted.

Two broad families of methods exist for step 2. The first constructs *artificial* representatives, such as cluster centroids in k-means, that do not correspond to any actual day in the data. The second selects *real* representatives, such as medoids in k-medoids, that are actual days from the historical data. Both families have been studied in the context of capacity expansion planning [9, 18, 12]. An important distinction is that artificial representatives can combine features from different days, creating synthetic profiles that never occurred in reality, whereas real representatives preserve the internal consistency of each day.

2.2 Formal problem statement

Let n denote the total number of original periods and k the number of representative periods. Each period spans 24 hours. For each hour t, let D_t denote the demand and $A_{g,t} \in [0, 1]$ the normalised availability factor of generator g. Each representative period r is assigned a weight W_r, describing how many original periods it stands for. The reduced investment model then minimises investment cost plus a weighted sum of operational costs over the k representatives, instead of over all n original periods. The accuracy of the reduced model depends entirely on how well the chosen representatives and their weights capture both typical operating conditions and extreme events from the original data.

2.3 Lower and upper bounds on solution quality

Standard clustering methods, including k-medoids, produce a reduced model that is a relaxation of the original problem. The reduced model disregards the most extreme conditions, so the investment plan it recommends looks cheaper than it truly is. Formally, when the reduced model is a relaxation of the full problem, meaning that it optimises over a subset of the original time steps, its optimal cost is a lower bound on the optimal cost of the full model [13]. This matters in practice: the investment plan found by standard clustering may be infeasible or suboptimal when evaluated against the complete time series, by disregarding worst-case days (high demand, low availability).

The worst-case construction described in Section 3.2 addresses this by replacing each cluster's representative with an artificially constructed period that is at least as demanding as any real period in that cluster. Under this construction, it can be shown that the reduced model provides an upper bound on the optimal cost and that the resulting investment plan is feasible for the full model [8]. This plan is conservative but guaranteed to work. However, the artificial period can be more extreme than any day that has actually occurred in the dataset, because it combines peak demand from one day with the worst renewable availability from a different day. This combination is deliberate: by constructing a period that dominates every real period in demand and availability ratio, the resulting investment plan is guaranteed to be feasible for all of them, which is what produces the upper-bound property.

2.4 Research question

The lower and upper bounds show the lowest and highest possible values for the true optimal cost. However, both bounds are based on either ignoring extreme conditions or creating artificial ones. The central question this work addresses is whether a real period from the historical data, one whose demand and availability values reflect conditions that genuinely occurred together, can close this gap without the use of artificial construction. The proposed methods are described in Section 3.

3 Real Worst-Case Period Selection for Time Series Aggregation

This section describes the six clustering methods. Three algorithms serve as reference points: standard clustering and two variants that add an artificial worst-case period. The other three are the contribution of this work: they select *real* periods from the original data to represent worst-case conditions, instead of constructing a single artificial one.

3.1 Shared algorithmic structure

All six methods follow the four-step time series aggregation procedure described in Section 2.1. The number of periods k, the period length of 24 hours, and the Tulipa energy model used in step 4 are fixed across all methods. The methods differ in *how* the k periods are selected, and for the artificial variants also in how the appended worst-case period is weighted; the rest of the procedure is identical.

Method	**Period selection strategy**	**Weight rule**
k-Medoids (baseline)	k medoids of the original periods	Number of periods in each cluster
Artificial Worst-Case (unit weight)	$k-1$ medoids plus one constructed worst-case period	Cluster sizes, with the worst-case period weight equal to 1
Artificial Worst-Case (fractional weight)	$k-1$ medoids plus one constructed worst-case period	Cluster weights scaled by $(1-\alpha)$, while the worst-case period weight equals $\alpha\, n$
Real Net-Load Extreme	$k-1$ medoids plus the real period with the highest net-load score	Cluster sizes, where the selected period keeps its own weight
Real Elgersma-Score	$k-1$ medoids plus the real period with the highest Elgersma score	Cluster sizes, where the selected period keeps its own weight
ENS-Guided Real	$k-1$ medoids plus the real period chosen from the worst-performing cluster after the first solve	Cluster sizes, where the selected period keeps its own weight

Table 1: Overview of the six time series aggregation methods compared in this work. All methods produce k representative periods that are passed into the Tulipa energy model. Each real worst-case method selects a single global extreme period, mirroring the single artificial worst-case period used in the Elgersma construction it is designed to compare against.

3.2 Baseline and artificial worst-case methods

k-Medoids baseline. The baseline is standard k-medoids clustering, a widely used clustering algorithm in which every representative period is an actual period from the data, namely the medoid (most central member) of its cluster [12]. Each representative is weighted by the number of original periods assigned to its cluster. This method captures average conditions well, but tends to smooth out rare extreme events, because such events are absorbed into larger clusters.

Construction of the artificial worst-case period. The two artificial methods rely on the worst-case construction of Elgersma [5]. A naive worst-case period would take the maximum de-

mand and the minimum availability of every energy source across all hours. This is overly pessimistic, because the lowest demand hour and the lowest availability hour rarely occur together. The Elgersma construction avoids this by linking availability to demand through a ratio. For a set of periods, let D_t be the demand at hour t and $A_{g,t}$ the available fraction of generator g at hour t. The artificial worst-case period is built per local hour as

$$D_r = \max_t D_t, \tag{1}$$

$$\gamma_{g,r} = \min_t \frac{A_{g,t}}{D_t}, \tag{2}$$

$$A_{g,r} = D_r \, \gamma_{g,r}. \tag{3}$$

Equation (1) takes the highest demand, and equation (2) takes the lowest availability-to-demand ratio for each generator. Multiplying the two in equation (3) yields an availability that is low precisely when demand is high. The weight of such a period is computed as $W_r = \sum_t (D_t/D_r)$, the sum of demand ratios of the periods it represents. A key property of this construction is that the demand of the artificial period is at least as high, and its availability ratio at most as low, as that of any real period it represents. This is what allows the reduced model to provide an upper bound and a feasible plan, rather than the lower bound given by standard clustering [8].

Note that the upper-bound guarantee from [8] applies when the Elgersma construction is applied to every cluster, not only when a single artificial period is appended. The two artificial variants tested here do not satisfy this condition.

Artificial Worst-Case (unit weight). This method runs k-medoids on $k-1$ clusters and appends one artificial worst-case period as the k-th representative, with a weight of 1. The artificial period therefore counts as a single extra day added on top of the clustered data.

Artificial Worst-Case (fractional weight). This variant gives the artificial period a larger share of the total weight. The weights of the $k-1$ medoid clusters are scaled down by a factor $(1-\alpha)$, and the artificial period receives a weight of $\alpha\, n$, where n is the total number of original periods and $\alpha = 0.1$ is a fixed fraction controlling how much influence the worst-case period has on the investment objective. The total weight is preserved, so the artificial period contributes a fixed fraction of the represented time. The motivation is to give the worst case more influence on the investment decision.

3.3 Real Net-Load Extreme Period Selection

The first contributed method replaces the constructed worst-case period with a real period taken from the data. The idea is simple: instead of building a synthetic stress period, find the real period that places the most stress on the system, and keep it as one of the representatives.

Stress is measured by a *net-load score* that can be computed directly from the input data, without solving the investment model. For a candidate period p, the score sums the residual demand over all 24 hours:

$$\text{score}_{\text{net}}(p) = \sum_h \Big[D(p,h) - \sum_g \text{cap}_g \, A_g(p,h) \Big]. \tag{4}$$

Here $D(p,h)$ is the demand in hour h of period p, $A_g(p,h) \in [0,1]$ is the normalised availability factor of generator g, and cap_g is its installed capacity. The term $\text{cap}_g \, A_g(p,h)$ is the power deliverable by generator g in that hour, so the bracketed expression is the net load: the residual demand that dispatchable sources must cover. A high net-load score marks a period with high demand and low renewable output, which is exactly the kind of period that stresses the grid.

The capacity values cap_g are the initial installed capacities of each generator as specified in the dataset input, not the result of any investment optimisation. After the model invests, actual capacities will be larger, so the net-load score is a heuristic approximation of system stress based on the pre-investment state. This avoids the circular dependency of needing to solve the model before selecting representative periods. Unlike earlier work that uses net load as a clustering criterion [9, 18], the method here uses it only to rank periods and keep the single most stressful one as a representative.

The algorithm is as follows. All n periods are scored using equation (4) and sorted from highest to lowest. The single highest-scoring period is set aside. The remaining periods are clustered into $k-1$ medoids in the usual way, and the set-aside period is added as the k-th representative, with a weight of 1 (representing itself). Unlike the artificial period, the selected period is a real day, so its pattern of demand and availability is consistent.

3.4 Real Elgersma-Score Period Selection

The second contributed method selects a real period whose internal structure most closely resembles what the artificial Elgersma construction would produce. Instead of using the Elgersma formula to build a synthetic period, it uses the same logic to *score* real periods and pick the most extreme one.

For each period p, the score mirrors equations (1)–(3), but restricted to the hours within that single period:

$$D_{\max}(p) = \max_h D(p,h), \tag{5}$$

$$\gamma_g(p) = \min_h \frac{A_g(p,h)}{D(p,h)}, \tag{6}$$

$$\text{score}_{\text{elg}}(p) = D_{\max}(p) - \sum_g D_{\max}(p)\,\gamma_g(p) = D_{\max}(p)\Big(1 - \sum_g \gamma_g(p)\Big). \tag{7}$$

A high score marks a period that combines a high peak demand with low availability-to-demand ratios, which is the same combination the artificial construction targets. The highest-scoring real period is selected and appended in the same way as the algorithm in Section 3.3.

There is an important difference between this score and the artificial construction. The construction in equations (1)–(3) may draw its peak demand from one day and its worst availability ratio from a completely different day, so the artificial period can be more extreme than any real period. The score in equation (7) stays within one real period, so it reflects only conditions that occurred simultaneously. This restriction is deliberate: the goal is to find the real period closest in character to the artificial one, while keeping the temporal consistency that the artificial construction loses. Using the Elgersma formula to score and select real periods, rather than to construct synthetic ones, has not yet been documented in prior studies.

3.5 ENS-Guided Real Period Selection

Unlike the two methods above, which select a period from the input data before solving, this method uses the solved model's own output to decide which real period matters most. It is therefore an a posteriori method.

The signal used is energy not served (ENS). Tulipa already includes an ENS asset, which supplies demand at a high penalty cost when no other source can. A representative period with high ENS is one where the model could not meet demand, so it points to a reliability weak spot. The method works in two phases, shown in Algorithm 1.

The zero-ENS fallback in line 2 handles cases where the clustered model already meets all demand, as adding an extreme period would only increase computation without improving reliability.

Algorithm 1 ENS-Guided Real Period Selection

Cluster the data into k medoids and solve the reduced model (first solve)
if total ENS $= 0$ **then**
    return the first result ▷ system already reliable
end if
Find the representative period r^* with the highest total ENS
Among the original periods mapped to r^*, select p^* with the highest net-load score (Eq. 4)
Cluster the data into $k - 1$ medoids
Append p^* as the k-th representative and solve again
return the final result

This method is related to two ideas from the literature. Teichgraeber et al. [15] iteratively add extreme periods that cause a reliability shortcoming, until none remain, using slack variables to detect the shortfall. Hilbers et al. [7] argue that aggregation should be adapted to the model and select extreme periods after inspecting model output. The method here shares the model-guided spirit of both, but differs in two ways. First, it adds a within-cluster step: rather than adding the failing period directly, it first locates the worst cluster and then searches inside it for the most stressful real period. Second, it runs a single iteration rather than iterating to convergence, which keeps the cost to one extra solve per value of k. That extra solve is the main drawback of the method, roughly doubling the solve time. Section 4 examines whether the reliability gain is worth it.

4 Experimental Evaluation

4.1 Experimental setup

Dataset. All experiments use the tutorial-9 dataset from the TulipaEnergy repository [3]. The dataset covers three scenarios (years 1995, 2008, and 2009), all mapped to a single planning milestone year of 2030. Each scenario contains 8760 hourly timesteps, producing 365 periods of 24 hours each. With three scenarios, the total number of periods is $n = 1095$. Five profile types are included: solar availability, offshore wind availability, onshore wind availability, electricity demand, and hydro inflow. The system contains 14 assets, including dispatchable generators (CCGT, OCGT), renewable producers (solar, onshore wind, offshore wind), three storage assets (battery, hydrogen storage, hydro reservoir), and an energy-not-served (ENS) asset that supplies unmet demand at a commodity price of 0.18 k€/MWh (180 €/MWh).

Evaluation metrics. The primary metric is *regret*, which measures how much more expensive the investment plan from the reduced model is when re-evaluated at full temporal resolution. The procedure is illustrated in Figure 1. First, the investment model is solved at reduced resolution to obtain an investment plan. Second, those investments are fixed and the operational model is re-solved at full resolution ($k = 1095$). Relative regret is then defined as

$$\text{Regret}(\%) = \frac{C^{\text{inv}}_{\text{agg}} + C^{\text{op}}_{\text{full}} - C^{\text{full}}}{C^{\text{full}}} \times 100, \tag{8}$$

where $C^{\text{inv}}_{\text{agg}}$ is the investment cost from the reduced solve, $C^{\text{op}}_{\text{full}}$ is the operational cost from the full-resolution re-solve with fixed investments, and C^{full} is the benchmark total cost. A regret of zero means the reduced model recovered the optimal investment plan. Positive regret means the plan is suboptimal.

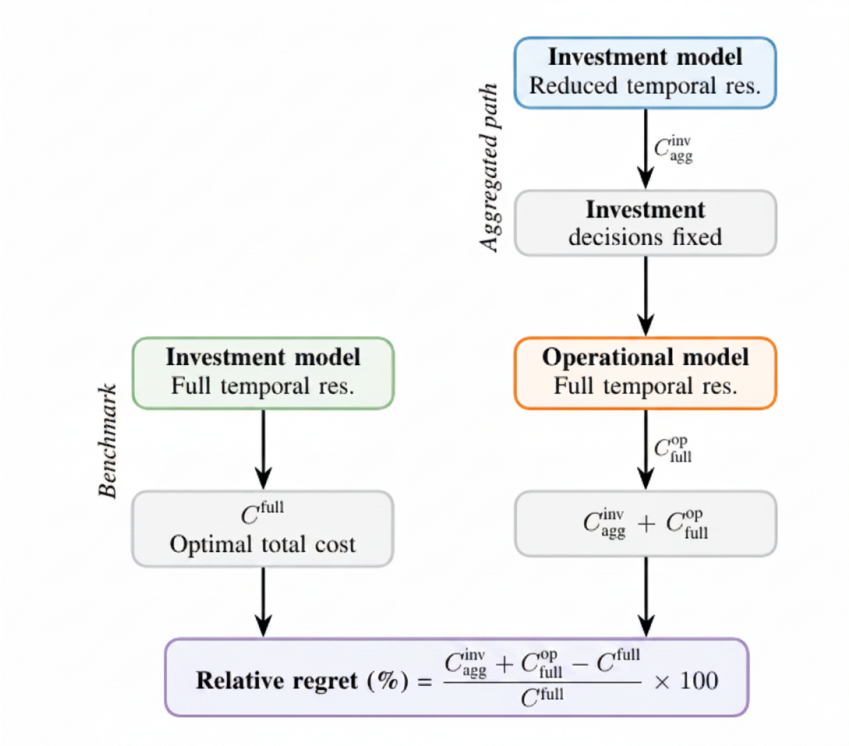

Figure 1: Regret analysis procedure. Investments are fixed from the reduced model and re-evaluated at full temporal resolution. The relative regret compares the resulting total cost to the full-resolution benchmark, adapted from [2].

Two secondary metrics capture the reliability of the investment plan when re-evaluated at full resolution. Loss of Load Expectation (LOLE, h/yr) sums the weighted hours in which any energy is not served (ENS > 0). Expected Energy Not Served (EENS, MWh/yr) additionally weights those hours by the amount of unmet demand. Solve time measures the time to run clustering and model solving for each method at each k-value.

A limitation of the dataset that affects the interpretation of LOLE and EENS results is that the ENS commodity price in the tutorial data (0.18 k€/MWh) is low relative to investment costs, so the model finds it cheaper to tolerate energy not served rather than investing in additional capacity. This produces a full-resolution LOLE of 336 h/yr, far above the TenneT standard of 4 h/yr [17]. The cost-optimal investment plan for this dataset therefore has high LOLE by design, and method comparisons should be made against that benchmark rather than against the TenneT target. Absolute LOLE values are not comparable to real-world standards, but relative comparisons between methods remain meaningful.

Benchmark. The benchmark runs k-medoids at $k = 1095$ so every period is its own representative, giving reference values: $C^{\text{full}} = 512.44$, investment $= 647.23$, LOLE $= 336$ h/yr, EENS $=$ 25.71 MWh/yr.

Implementation. All experiments are implemented in Julia [1] using the TulipaClustering [11] and TulipaEnergyModel packages with Gurobi [6] as the optimisation solver. Each k-value run uses an isolated in-memory DuckDB connection to prevent any shared state between runs. The

representative period selection algorithms described in Section 3 are implemented on top of TulipaClustering's cross-scenario layout, which shares representative periods across all three climate scenarios rather than making them scenario-specific. Each k-value is evaluated over five random seeds (n_seeds = 5), and all reported values are averages across those seeds, which reduces but does not eliminate sensitivity to k-medoids initialisation at very low k. The k-range spans 2 to 1095, with step size 2 for $k \leq 50$, step size 5 for $k \leq 202$, step size 25 for $k \leq 402$, and step size 150 beyond that. Full-range plots for each individual method comparison are provided in Appendix A.

4.2 Overall convergence

Throughout this section, *convergence* means that regret approaches zero as k increases, that is, the reduced model recovers the full-resolution benchmark cost as more representative periods are added.

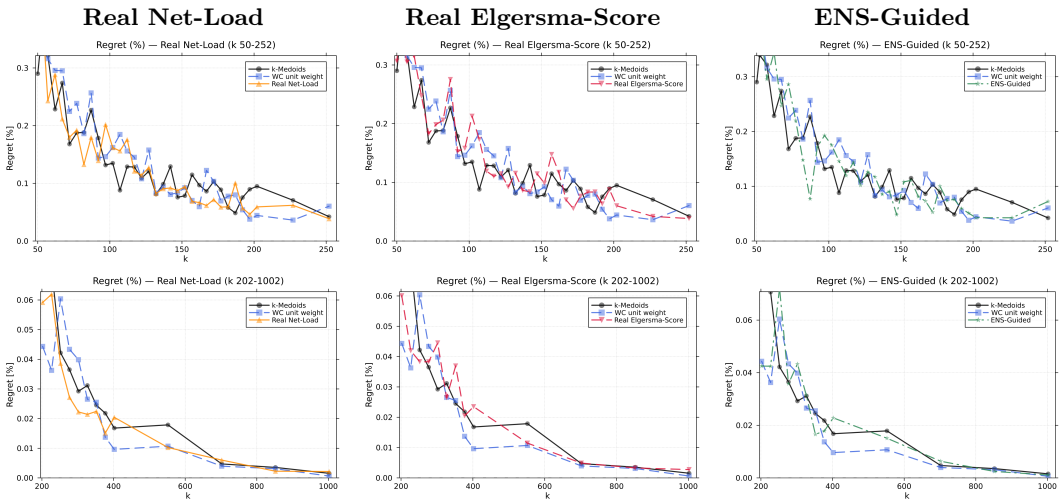

Figure 2: Regret (%) for each real worst-case method against the two baselines (k-medoids and WC unit weight), averaged over five seeds. Each column shows one contributed method; both baselines are identical across columns and serve as the common reference. Top row: $k = 50$–252, the separation zone where method differences are most visible. Bottom row: $k = 202$–1002, showing all methods approaching near-zero regret. The dashed line marks zero regret. Full-range plots are in Appendix A.

The central question is whether the real worst-case methods can match plain k-medoids on solution quality. Figure 2 shows regret averaged over five seeds for each contributed method against the two baselines; the fractional-weight variant is examined separately in Section 4.3.

For $k \leq 50$, all methods show high and variable regret. Five-seed averaging reduces noise relative to a single run, but the basic aggregation error at very low k remains: too few representative periods must cover 1095 original periods, so any method is unreliable in this zone.

The interesting region is $50 \leq k \leq 252$, shown in the top row of Figure 2. At $k = 102$, plain k-medoids has the lowest regret at 0.135%. The real worst-case methods lag slightly behind: Real Net-Load reaches 0.162%, ENS-Guided 0.175%, and Real Elgersma-Score 0.213%, with WC unit weight also at 0.162%. The slightly slower performance of the real worst-case methods at this k is expected: one representative slot is reserved for the extreme period, leaving one fewer slot for the typical operating conditions that drive most of the objective function.

The picture shifts by $k = 202$. At that point, plain k-medoids sits at 0.095%, while WC unit weight (0.044%), ENS-Guided (0.043%), Real Net-Load (0.059%), and Real Elgersma-Score (0.060%) all outperform it. This crossover suggests that once enough representative periods cover ordinary conditions, the dedicated extreme-period slot pays off. Beyond this point the advantage does not persist: by $k = 302$ all methods reach or go below 0.045%, and by $k = 402$ all are at or below 0.025%, becoming essentially indistinguishable, as confirmed by the bottom row of Figure 2.

The practical implication is that no real worst-case method consistently improves on plain k-medoids across the full k-range: they are slightly slower to approach zero regret at low-to-medium k, briefly competitive around $k = 200$, and indistinguishable beyond $k \approx 400$, which is approximately 36% of the full dataset.

4.3 Structural failure of fractional weighting

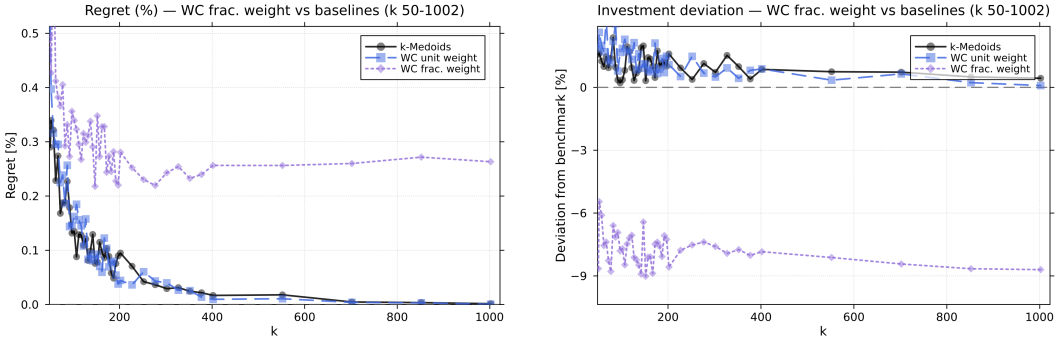

Figure 3: Left: Regret (%) for the WC fractional-weight method against the two baselines ($k = 50$–1002). The fractional-weight variant fails to approach zero regret, plateauing at approximately 0.22–0.28% while both baselines continue falling. Right: Signed investment deviation from the benchmark (%) for the same three methods. Negative values indicate under-investment. The fractional-weight variant maintains a persistent deficit of approximately −8% to −9% across all k-values. The dashed line marks zero.

The Artificial Worst-Case (fractional weight) method fails to approach zero regret at any tested k-value, as shown in Figure 3. Regret plateaus at approximately 0.22–0.28% for all $k > 200$, with no improvement as k increases. At $k = 1002$, regret is still 0.263%, essentially identical to $k = 202$ at 0.281%. The right panel gives a clear explanation: the investment deviation for this method sits at approximately −8% to −9% across all k-values, reflecting a consistent under-investment of roughly 50–55 model cost units relative to the benchmark of 647.

The mechanism is the weight scaling that defines this method. By multiplying all existing cluster weights by $(1 - \alpha) = 0.9$, every original period contributes 10% less to the operational objective. This permanently reduces the apparent cost of under-investing in capacity, regardless of how many clusters are used. Adding more representative periods does not fix this, because the distortion is applied uniformly to all of them. The artificial worst-case period itself cannot compensate, as its weight $\alpha n = 109.5$ covers a single representative in a model of 1095 original periods. The result is a persistent non-zero regret plateau: the method asymptotically approaches a wrong value rather than the full-resolution benchmark. Importantly, this failure is not specific to this dataset. The weight scaling applies uniformly regardless of which periods are selected or how many clusters are used, so the same plateau would arise in any setting where global weight distortion is applied. This serves as a general warning: any method that scales all existing period weights by a fixed factor risks introducing a bias that additional representative periods cannot correct.

4.4 Reliability at full resolution

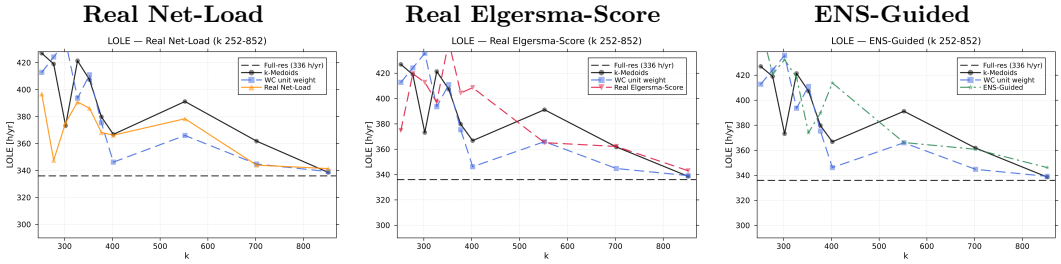

Figure 4: LOLE (h/yr) at full resolution for each real worst-case method against the two baselines (k = 252–852), averaged over five seeds. The dashed line marks the full-resolution benchmark (336 h/yr). The TenneT standard of 4 h/yr is outside the plotted range and is visible in the full-range plots in Appendix A.

Figure 4 shows LOLE at full resolution for the medium-to-high k range, where method comparisons are most meaningful. No real worst-case method consistently achieves lower LOLE than plain k-medoids. At $k = 302$, k-medoids achieves LOLE = 373 h/yr, while Real Net-Load gives 375 h/yr, Real Elgersma-Score 413 h/yr, and ENS-Guided 432 h/yr. At $k = 702$, k-medoids gives 362 h/yr, with Real Net-Load at 344 h/yr, Real Elgersma-Score at 362 h/yr, and ENS-Guided at 361 h/yr. The ordering is not stable across k-values. Explicitly preserving an extreme period does not produce a consistently more reliable investment plan on this dataset.

The reason is a property of the dataset rather than the methods. Because the ENS commodity price is low relative to investment costs, the cost-optimal investment plan for this dataset tolerates high LOLE by design: accepting unserved energy is cheaper than building additional capacity. The full-resolution benchmark itself has LOLE = 336 h/yr, so the cost-optimal plan for this dataset already tolerates a substantial amount of unserved energy. Reliability comparisons between methods are therefore valid, but none of the methods can approach the TenneT standard of 4 h/yr on this dataset because the objective function does not value reliability at that level. A proper evaluation of whether extreme period selection improves reliability would require an ENS penalty calibrated to a realistic standard, on a dataset where the full-resolution LOLE is near that target.

At low k, instability is visible across all methods even after five-seed averaging. Real Elgersma-Score shows the most pronounced instability: at $k = 4$ the 5-seed average gives regret 4.47% and LOLE 2037 h/yr, substantially worse than plain k-medoids at the same k (regret 2.90%, LOLE 1356 h/yr). The cause is the same as described in Section 4.2: at $k = 4$, only three medoids cover 1094 original periods, and the Elgersma score identifies one highly extreme period that occupies a slot, leaving those three medoids insufficient to represent ordinary operating conditions. This effect diminishes as k grows and the method approaches the same level as other methods by $k \approx 100$. Full-range early-k plots for all methods are provided in Appendix A.

4.5 Computational cost

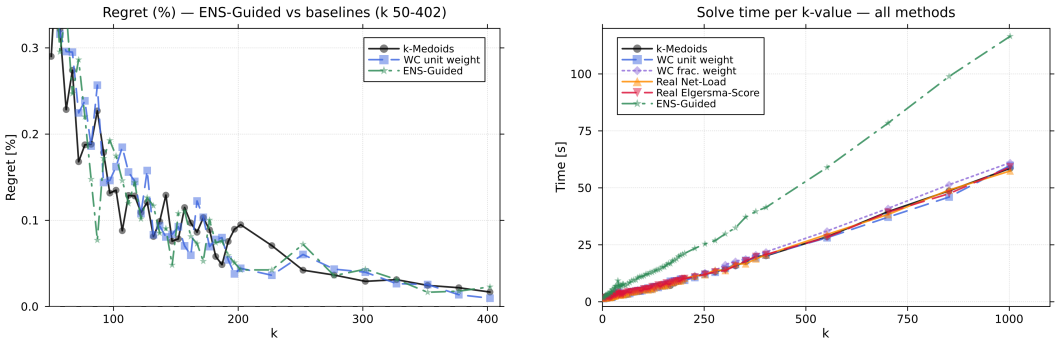

Figure 5: Left: Regret (%) for ENS-Guided selection against the two baselines ($k = 50$–402). Right: Solve time (s) per k-value for all six methods, full range. ENS-Guided costs approximately 2× the solve time of all other methods but does not achieve a corresponding improvement in regret.

Figure 5 shows both the regret and the solve time for ENS-Guided selection. The right panel confirms that ENS-Guided costs approximately 2× the solve time of all other methods across all tested k-values: at $k = 402$, k-medoids takes 20.2 s while ENS-Guided takes 41.4 s; at $k = 702$, 39.6 s versus 78.4 s; at $k = 1002$, 58.6 s versus 116.5 s. The ratio is stable at 1.98–2.05× throughout, consistent with the two-phase design that runs a first solve and a final solve for every k-value. All other methods are comparable in solve time to k-medoids, since the extra computation for scoring and appending a single real period is negligible relative to Gurobi's solve time.

The left panel directly addresses whether the higher cost is justified. At $k = 202$, ENS-Guided regret (0.043%) is marginally lower than k-medoids (0.095%), but this advantage disappears quickly: at $k = 302$ ENS-Guided gives 0.043% against k-medoids at 0.029%, and at $k = 402$ the values are 0.023% and 0.017% respectively. ENS-Guided does not perform more than twice better than k-medoids in regret or LOLE at any tested k-value. The 2× computational cost is therefore not justified on this dataset. Whether it pays off in settings where the ENS penalty is calibrated to a realistic reliability standard, making the model genuinely sensitive to extreme events, is an open question and the most important direction for future work on this method.

5 Conclusions and Future Work

5.1 Summary and conclusions

This work investigated whether real observed periods can replace the single artificial worst-case period used in existing methods for representative period selection in energy system planning. The central question was whether such real-period strategies could match or improve on solution quality and system reliability compared to standard k-medoids clustering, while keeping each selected day internally consistent. Three strategies were proposed and implemented: Real Net-Load Extreme selection, which ranks periods by a capacity-weighted residual demand score; Real Elgersma-Score selection, which applies the Elgersma worst-case scoring logic directly to real periods to find the one most similar in character to the artificial construction; and ENS-Guided selection, which uses the solved model's own energy-not-served output to identify the representative period with the highest total ENS and then picks the most demanding real period among those originally mapped to it. All three were compared against two artificial worst-case variants and the standard k-medoids baseline, using the Tulipa Energy Model on a three-scenario European planning dataset.

On solution quality, averaged over five seeds, all non-fractional methods approach near-zero regret by approximately $k = 400$, which is roughly 36% of the full dataset of 1095 periods. At low-to-medium k no single method consistently dominates: plain k-medoids is marginally best at $k = 102$ (0.135% regret), but around $k = 202$ the real worst-case methods and the unit-weight variant briefly outperform it, before all methods converge to below 0.025% regret by $k = 402$. The differences in the medium-k range are small enough that no real worst-case method can be recommended on solution quality grounds alone, and no method can be ruled out either.

On reliability, none of the real worst-case methods consistently achieves lower LOLE than plain k-medoids at medium to high k. This conclusion is influenced by the dataset's low ENS commodity price, which causes the full-resolution model to accept energy not served rather than invest in additional capacity. Under these conditions the cost-optimal investment plan for this dataset tolerates high LOLE by design, and no period selection strategy can change that without changing the objective function itself. A definitive answer to whether real extreme period selection improves reliability requires experiments on a dataset where the model is genuinely sensitive to reliability gaps.

Two further individual results deserve emphasis. The fractional-weight artificial worst-case variant fails to approach zero regret at any tested k-value, plateauing at approximately 0.24–0.28% with a persistent under-investment of approximately 8–9% relative to the benchmark. This failure occurs because the global weight scaling mechanism permanently reduces the contribution of every period to the operational objective by a fixed factor, and no number of additional representative periods can correct this bias. This result is not dataset-specific and serves as a warning when using global weight distortion. The Real Elgersma-Score method shows the most severe instability at low k: in the five-seed average, it reaches LOLE of 2037 h/yr at $k = 4$ and regret of 4.47%, substantially worse than plain k-medoids at the same k (1356 h/yr and 2.90% respectively). The ENS-Guided method costs approximately twice the computation of all other methods, with a stable ratio of 1.98–2.05× throughout. It does not perform more than twice better than plain k-medoids in regret or LOLE at any tested k-value, so the extra computational cost is not justified on this dataset.

5.2 Future work

The most important next step is repeating all experiments with an ENS penalty calibrated to a realistic standard such as the TenneT target of 4 h/yr [17], on a dataset where full-resolution LOLE is near that standard, and with per-seed results stored so that variance bands or confidence intervals can be reported. The ENS-Guided method in particular warrants re-evaluation under these conditions, since its design is specifically motivated by settings where reliability shortcomings are costly and the current dataset does not provide them.

Another direction is combining real period selection with the full Elgersma construction applied to every cluster, rather than only appending a single period. This would restore the upper-bound guarantee proven by Elgersma [5], while the real-period component would narrow the gap between that conservative upper bound and the true optimal cost. Whether this combination produces tighter and more practically useful bounds has not yet been studied.

Finally, all experiments in this work use a single tutorial dataset with a relatively small system. Testing on a larger real-world European planning instance, closer to the scale at which Tulipa is intended to operate, would clarify whether the finding that a few hundred representative periods suffice for near-zero regret holds at practical scale, and whether the relative performance of the six methods changes under a more complex system with more assets and more interdependencies between carriers.

References

[1] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

[2] Sven Butzelaar. Extreme-preserving hierarchical clustering for automated temporal partitioning in energy system optimization. Master's thesis, Delft University of Technology, 2026. Supervisors: Dr. Germán Morales-España and Maaike Elgersma. Jointly conducted with TNO. Unpublished draft.

[3] TulipaEnergyModel.jl Developers. My awesome energy system tutorial 9 dataset. , 2026.

[4] TulipaEnergyModel.jl Developers. TulipaEnergyModel.jl. GitHub repository, 2026.

[5] Maaike Elgersma, Luca Santosuosso, Sonja Wogrin, Germán Morales-España, Mathijs de Weerdt, Greg Neustroev, and Lotte Kremer. Obtaining upper bounds for GEP with storage fast: Performance guarantees for TSA based on the worst case. Working draft, 2026.

[6] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2026.

[7] Adriaan P Hilbers, David J Brayshaw, and Axel Gandy. Reducing climate risk in energy system planning: A posteriori time series aggregation for models with storage. *Applied Energy*, 334:120624, 2023.

[8] Lotte A. A. Kremer. Stochastic programming for energy models: A blended cross-scenario representative periods approach. Master thesis, Delft University of Technology, 2025.

[9] 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.

[10] Energy Model and Uroš Gluščević. Start-up and shut-down costs in an energy system optimisation model with fully flexible temporal resolutions. 2025.

[11] G. Neustroev, D. A. Tejada-Arango, L. Clisby, and G. Morales-Espana. Tulipaclustering.jl, 2026.

[12] John Paparrizos, Fan Yang, and Haojun Li. Bridging the gap: A decade review of time-series clustering methods. *arXiv preprint arXiv:2412.20582*, 2024.

[13] Luca Santosuosso, Bettina Klinz, and Sonja Wogrin. What are we clustering for? establishing performance guarantees for time series aggregation in generation expansion planning. *arXiv preprint arXiv:2510.09357*, 2025.

[14] 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.

[15] Holger Teichgraeber, Lucas Elias Küpper, and Adam R Brandt. Designing reliable future energy systems by iteratively including extreme periods in time-series aggregation. *Applied Energy*, 304:117696, 2021.

[16] Holger Teichgraeber, Constantin P Lindenmeyer, Nils Baumgärtner, Leander Kotzur, Detlef Stolten, Martin Robinius, André Bardow, and Adam R Brandt. Extreme events in time series aggregation: A case study for optimal residential energy supply systems. *Applied energy*, 275:115223, 2020.

[17] TenneT. Electricity supply security under pressure after 2030, May 2025.

[18] Ali Yeganefar, Mohammad Reza Amin-Naseri, and Mohammad Kazem Sheikh-El-Eslami. Improvement of representative days selection in power system planning by incorporating the extreme days of the net load to take account of the variability and intermittency of renewable resources. *Applied Energy*, 272:115224, 2020.

A Full-Range Method Comparison Plots

This appendix contains plots that complement the focused figures in Section 4 by showing the full k-range and the volatile low-k zone ($k < 50$) that is excluded from the main figures. Each plot contains three lines: the contributed method and the two baselines.

Real Net-Load

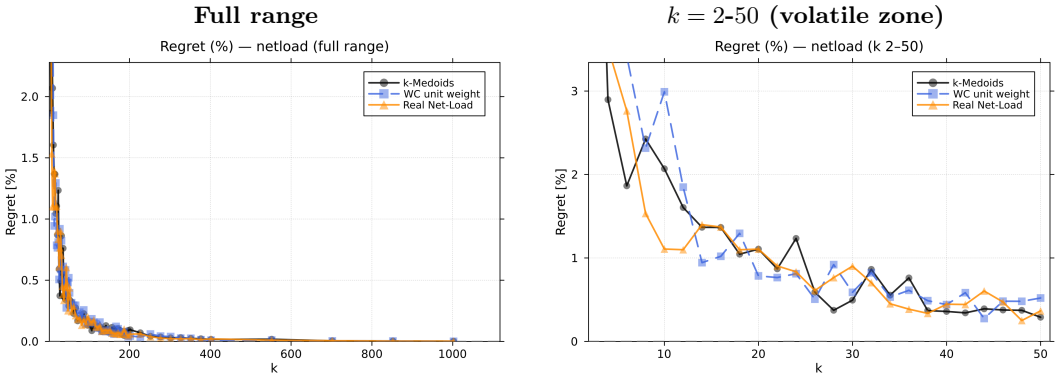

Figure 6: Regret (%) for Real Net-Load vs baselines. Left: full k-range showing overall convergence behaviour. Right: k = 2-50, the volatile zone where five-seed averaging reduces but does not eliminate initialisation noise. Dashed line marks zero regret.

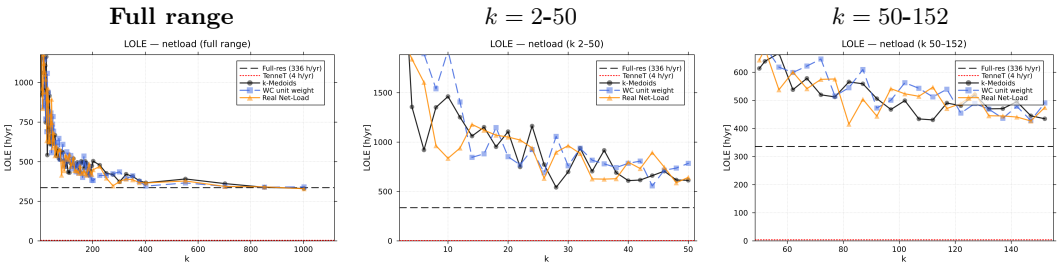

Figure 7: LOLE (h/yr) for Real Net-Load vs baselines. Left: full range. Centre: k = 2-50, low-k instability zone. Right: k = 50-152, early medium-k range not shown in the main paper. Dashed line: full-resolution benchmark (336 h/yr). Dotted red line: TenneT standard (4 h/yr).

Real Elgersma-Score

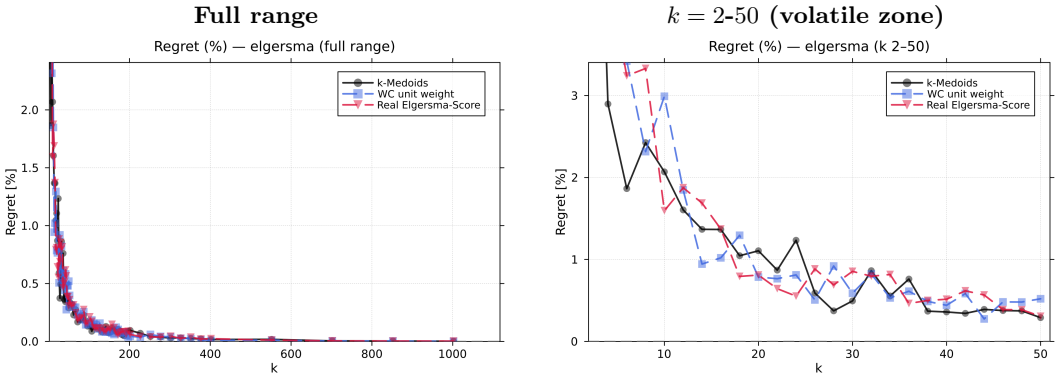

Figure 8: Regret (%) for Real Elgersma-Score vs baselines. Left: full range. Right: k = 2-50, showing the pronounced early instability of the Elgersma-Score method at very low k. Dashed line marks zero regret.

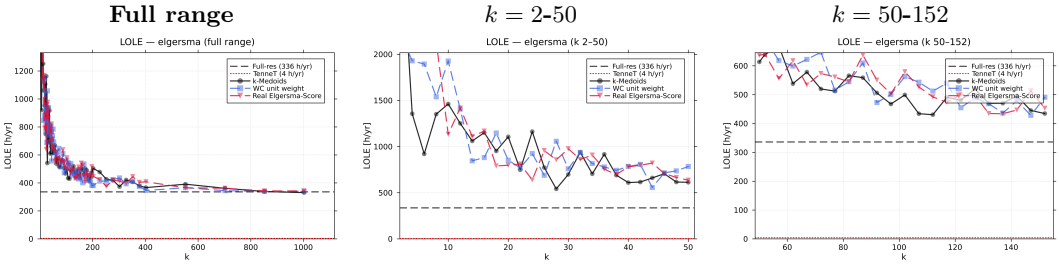

Figure 9: LOLE (h/yr) for Real Elgersma-Score vs baselines. Left: full range; the spike at $k = 4$ (5-seed average 2037 h/yr) is visible here. Centre: k = 2-50 zoom of that spike. Right: k = 50-152, where the method recovers toward the baseline level. Dashed line: full-resolution benchmark (336 h/yr). Dotted red line: TenneT standard (4 h/yr).

ENS-Guided

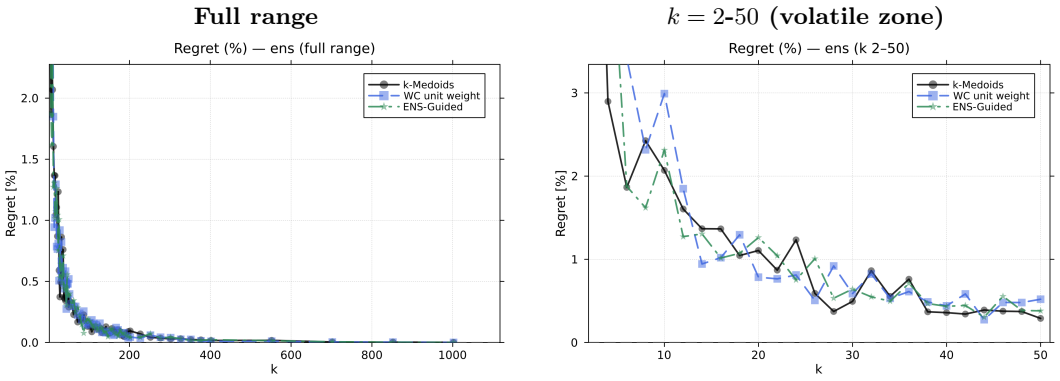

Figure 10: Regret (%) for ENS-Guided vs baselines. Left: full range. Right: k = 2-50, showing the volatile behaviour driven by the sensitivity of the pilot solve to random initialisation at very low k. Dashed line marks zero regret.

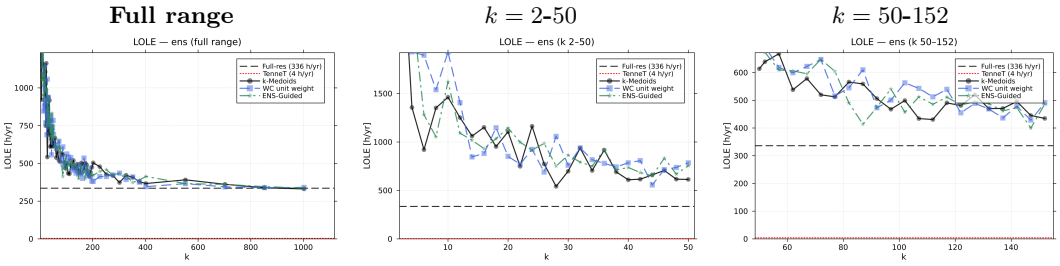

Figure 11: LOLE (h/yr) for ENS-Guided vs baselines. Left: full range. Centre: k = 2-50. Right: k = 50-152. Dashed line: full-resolution benchmark (336 h/yr). Dotted red line: TenneT standard (4 h/yr).

WC Fractional Weight

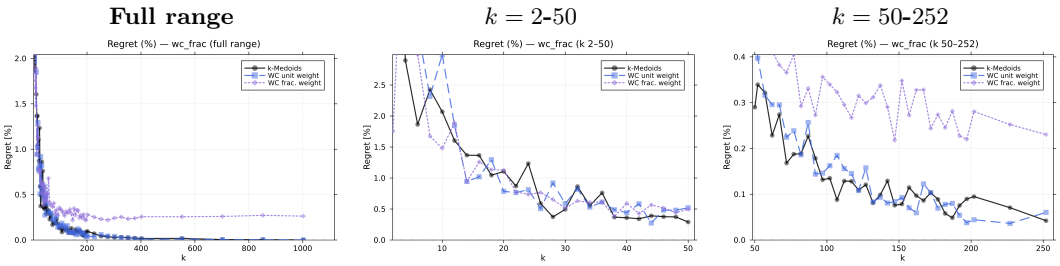

Figure 12: Regret (%) for WC fractional weight vs baselines. Left: full range, showing the plateau at approximately 0.24-0.28% that persists from k = 100 onward. Centre: k = 2-50, volatile zone. Right: k = 50-252, where the plateau first becomes clearly visible against the falling baselines. Dashed line marks zero regret.

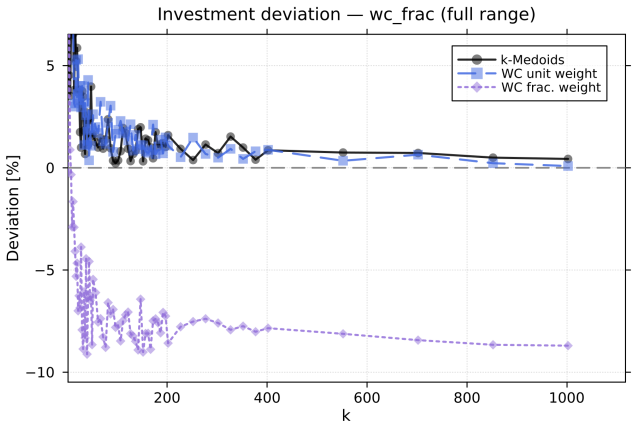

Figure 13: Investment deviation (%) for WC fractional weight vs baselines, full k-range. The persistent deficit of approximately -8% to -9% is visible across the entire range, confirming that the regret plateau in the figure above is driven by systematic under-investment rather than by clustering error. Dashed line marks zero deviation.

B Responsible Research

B.1 Ethical considerations

Energy system planning decisions affect infrastructure that serves millions of people over decades [10]. Investment plans that underestimate capacity requirements risk energy insecurity, while plans that overestimate them waste public resources. The methods studied in this work are therefore evaluated honestly, including results that are negative or inconclusive. No results are selectively omitted: the failure of the fractional-weight method, the absence of a reliability benefit from the real worst-case methods, and the limitations of the dataset are all reported and discussed.

B.2 Reproducibility

All code implementing the six period selection methods, the evaluation pipeline, the regret computation, and the result generation scripts is published in a public repository at `https://github.com/NamanChoudharyDev/Research-project-Tulipa`, so that every numerical result in this paper can be verified. The dataset used is the publicly available tutorial-9 dataset from the TulipaEnergy repository [3], so no proprietary or restricted data is involved and the input can be accessed independently. No heuristic filtering, manual smoothing, or outlier removal was applied to the raw profiles. The artificial worst-case periods are generated by a deterministic construction, and five fixed random seeds are used throughout all clustering runs in a deterministic order, meaning every result is fully reproducible given the same environment.

B.3 Software environment

All experiments are implemented in Julia v1.12.6 using TulipaClustering v0.5.2 and TulipaEnergyModel v0.21.0 as the core frameworks, with Gurobi as the optimisation solver. The repository includes a `Project.toml` and `Manifest.toml` that lock the exact package versions, ensuring the code runs in the same environment regardless of when it is executed. The Gurobi version and solver parameters are documented in the evaluation script.

B.4 Computational environment

All experiments were conducted on an HP laptop equipped with an Intel Core i7-13700H processor and 16 GB of RAM running Windows 11. No high-performance computing infrastructure was required. This means the methods and evaluation pipeline are accessible to researchers and practitioners without special hardware, which is relevant given that one stated motivation for representative period selection is making energy system models tractable on day-to-day hardware.

B.5 Limitations and transparency

Three limitations of the experimental setup are acknowledged. First, each k-value is averaged over five random seeds, which reduces but does not eliminate sensitivity to k-medoids initialisation at very low k. Since only seed-averaged results are stored, no confidence intervals or per-seed variance are reported; conclusions in the low-k region ($k < 50$) should therefore be treated as indicative rather than statistically robust. Second, the tutorial-9 dataset has an ENS commodity price of 0.18 k€/MWh that is low relative to investment costs, so the cost-optimal full-resolution plan already tolerates high LOLE. This limits the conclusions that can be drawn about the reliability benefit of extreme period selection, since the model objective does not penalise energy not served strongly enough for period selection to make a meaningful difference in LOLE. Third, all experiments use a single small dataset, and it is not known how the results generalise to larger, more realistic datasets. The unpublished supervisor draft that defines the Elgersma worst-case construction [5] is cited as the source of that method, for full transparency.

B.6 Use of large language models

Large language models were used in two ways during the project. First, for the writing of the report they were used as a grammar and spell-checking aid and for sentence-structure suggestions. Second, they were used to generate boilerplate Julia code for the plotting the results. No AI was used to produce or derive any of the contributed algorithms, the mathematical formulations, the experimental design, or the scientific analysis and conclusions