Clustering methods to find representative periods for the optimization of energy systems: an initial framework and comparison

Holger Teichgraeber[a,*], Adam R Brandt[a]

[a]*Department of Energy Resources Engineering, Stanford University, Green Earth Sciences Building 065, 367 Panama St., Stanford, California, USA*

Abstract

Modeling time-varying operations in complex energy systems optimization problems is often computationally intractable, and time-series input data are thus often aggregated to representative periods. In this work, we introduce a framework for using clustering methods for this purpose, and we compare both conventionally-used methods (*k*-means, *k*-medoids, and hierarchical clustering), and shape-based clustering methods (dynamic time warping barycenter averaging and *k*-shape). We compare these methods in the domain of the objective function of two example operational optimization problems: battery charge/discharge optimization and gas turbine scheduling, which exhibit characteristics of complex optimization problems. We show that centroid-based clustering methods represent the operational part of the optimization problem more predictably than medoid-based approaches but are biased in objective function estimate. On certain problems that exploit intra-daily variability, such as battery scheduling, we show that *k*-shape improves performance significantly over conventionally-used clustering methods. Comparing all locally-converged solutions of the clustering methods, we show that a better representation in terms of clustering measure is not necessarily better in terms of objective function value of the optimization problem.

Keywords: Energy systems, Clustering, Operational optimization, Time-series aggregation, Representative day, k-means

1. Introduction

Computational optimization is a key part of the design and evaluation process for energy industries. The design of energy supply, generation, and conversion systems typically involves complex tradeoffs in capital costs, energy efficiency, system reliability, and system flexibility. These complexities lend themselves well to optimization approaches wherein tradeoffs can be modeled with high fidelity and many cases can be explored. The current state of the art involves applying complex mixed integer linear or nonlinear programming (MILP or MINLP) models that include discrete choices about technologies along with realistic representations of technology physics and chemistry.

Treating time rigorously is a large challenge when optimizing energy systems. Optimal technology design and operation requires understanding of impacts over multiple time scales; design decisions last for years or decades, while operational decisions occur on the order of hours or minutes. Importantly, these different time scales are coupled: how a system is designed often strongly affects the way in which it can be operated in any given hour. Such couplings between time scales are only becoming more important as renewable energy integration increases the need for flexible energy system operation.

In many cases, design-level optimization problems for energy systems are computationally expensive, highly constrained, and in the case of nonlinear optimization, subject to multiple locally-optimal solutions. It is often computationally intractable to analyze such problems for the full set of temporal input data that the facility might encounter.

For example, electricity prices vary at scales of minutes to hours, but it is computationally intractable to represent electricity prices with this fidelity when optimizing, for example, the design of a battery storage system. Many energy system optimization inputs have this characteristic: vehicle charging schedules, industrial facility operations and power usage, heating and cooling demands, renewable power generation, or microgrid design and operation. To include short time scales phenomena in long-term planning, time-series data inputs are often aggregated to representative periods. Such aggregation can reduce computational times by 1-2 orders of magnitude. For example, a system might be optimized for a set of 5 representative days instead of an entire year of 365 days.

Representative periods can be created using time-series clustering methods. Time-series clustering is concerned with grouping periods (in many cases days) into groups that are similar. The clusters are then represented by one representative period each. Multiple clustering methods have been conventionally used to find representative periods, including: *k*-means, *k*-medoids, and hierarchical clustering (Table 1 shows for a list of papers in which such methods have previously been applied).

Examples of using these common methods to find representative periods for energy systems optimization are numerous. These applications range from capacity expansion planning of power systems with and without unit commitment (1–8), to the design of local energy supply systems (9–13), to the design of

*Corresponding author. Tel: +1-650-725-0851

Email addresses: hteich@stanford.edu (Holger Teichgraeber), abrandt@stanford.edu (Adam R Brandt)

building energy systems (14), to the design of combined heat and power plants (15), to the design of carbon capture facilities (16–19). Methods have been developed to include seasonal storage (8; 13; 20), bounding the error in the objective function that was introduced by clustering (9), and including clustering methods in developing advanced solution strategies (5; 11).

Other methods are popular in general time-series clustering but – to the best of our knowledge – have not been widely applied to find representative periods for energy systems optimization (21). These methods include dynamic time warping (DTW) barycenter averaging (DBA) clustering (22) and *k*-shape (23). These methods extract shape-based information, DBA by stretching or contracting the time axis, and *k*-shape by maximizing the cross correlation between shifted versions of the time-series. DTW is widely used in time series classification applications (24) and in clustering applications (22). More recently, *k*-shape has been proposed and was shown to perform well clustering a wide range of time series (23).

Some applications exist of shape-based clustering methods related to energy systems. Teeraratkul et al. (25) apply a clustering method that uses DTW as the distance measure to cluster residential electricity load profiles into similar groups (25), and show that the method outperforms *k*-means clustering. Blanco and Morales (26) apply *k*-shape to find wind power production scenarios for a stochastic unit commitment optimization problem.

Three studies compare different clustering methods. Schuetz et al. (14) evaluate *k*-means, *k*-medians, *k*-centers, and *k*-medoids clustering applied to the optimal design of building energy systems. They find that in the case of small demand variability, all methods perform well in terms of objective function value, and in the case of large demand variability, *k*-medoids captured the variability best in terms of objective function value of the optimization problem. Pfenninger (6) evaluates a variety of combinations of clustering methods (*k*-means, *k*-medoids, hierarchical clustering) and extreme value selection methods on a generation capacity expansion problem. He shows that results can vary significantly among methods, and points out that results can also vary significantly among model years, thus pointing to the need to incorporate several years of input data. Kotzur et al. (12) evaluate *k*-means clustering, *k*-medoids clustering, and hierarchical clustering with the medoid as the representation and formulate several two-stage optimization problems that are concerned with the design of energy supply systems: a CHP based energy supply system, a residential energy supply system, and an island electricity supply system. They conclude that the performance of representative days depends strongly on the system to be optimized. Further, they conclude that generally no clustering method outperforms all others, but that the medoid as representation captures variability in the data better than the centroid because it does not smooth the data as much as the centroid.

The above-mentioned energy systems optimization studies jointly investigate the representation in the operational and design domain of these two-stage problems. These problems are complex and include different characteristics such as storage, load shifting, and power generation. Little formal comparison has been performed across the suite of methods so as to understand which methods perform well on which problem characteristics.

In this work we aim to remedy this gap by introducing an analysis framework and performing structured intercomparison of seven time series clustering methods. We analyze the impact of clustering on the optimization results obtained for two operations optimization problems. We analyze only the operational optimization of systems and do not consider design decisions in order to decompose its effects. We compare traditionally-used clustering methods, as well as the shape-based methods that have not been widely applied. We formulate two operational optimization problems that are simple enough to be solvable deterministically for a full year of electricity price time series data. By applying various implementations of clustering, we can then directly compute the objective function error resulting from clustering. We then compare the clustering methods in the domain of the objective function of these optimization problems.

The aims of this paper are to: (1) develop a framework in which clustering methods for finding representative periods can be described, (2) analyze the operational representation of different clustering methods, especially centroid- vs. medoid-based clustering, (3) investigate the usefulness of shape-based clustering methods, and (4) investigate how well the clustering method maps into the objective function space of the optimization problem by investigating the performance of different clustering methods.

This paper proceeds as follows. In section 2, we introduce a framework for clustering methods to find representative periods. In sections 3, 4, and 5, we introduce the clustering methods that are compared in this paper, the optimization problem formulations that they are applied to, and the data, respectively. In section 6, we present and analyze the results, and we conclude in section 7.

2. Clustering Framework

Generally, the clustering problem aims to find representative periods and associated weights that represent the full time series in the optimization problem. The ideal clustering method would represent all of the information in the full dataset with a small fraction of the input data. However, all real clustering application suffer from some error due to data reduction. This error can be persistent in direction (bias) or random (noise). Importantly, in optimization we care about error introduced into the optimization objective function and decision variables, not the underlying clustering error. These two types of error are related but not equivalent: the best-fitting cluster assignment may not result in the most accurate objective function value (see Section 6.3 for additional details).

Clustering involves multiple steps, some of which are conflated or not made explicit in the prior literature. Figure 1 illustrates our framework for time-series clustering methods. In this framework, there are three general steps: (1) Normalization, (2) Assignment, and (3) Representation. All time series clustering

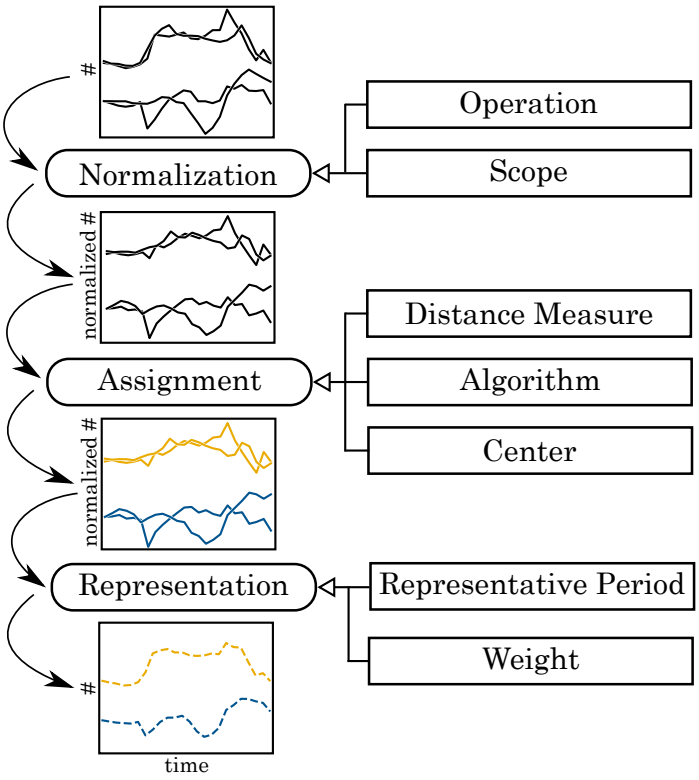


Figure 1: Framework of clustering methods for finding representative periods for the optimization of energy systems. # stands for an attribute that is to be clustered (e.g. electricity price).

methods applied to date can be classified by their methodological choices in these steps.

The first step in the framework is *normalization* of the data. Normalization is commonly used and necessary when using multiple input data sets (often called attributes) with different units because magnitudes of attributes can differ greatly. Normalization methods vary with the operation applied (e.g., division by largest value) and the scope of normalization (e.g., day-wise). Normalization operations include: (1) no normalization; (2) normalizing observations by the largest value (0-1 normalization); or (3) normalizing to mean zero and standard deviation 1 ($\mu = 0$ and $\sigma = 1$, also known as z-normalization).

Normalization scope can also differ. Figure 2 illustrates different normalization scopes in the case of a year of hourly data (here shown normalized with z-normalization for simplicity). For the full normalization scope (left), the time series as a whole is in the scope, so that the entire year of data is normalized using a single μ and σ. For the element-based normalization scope (center), each time step can be normalized individually by computing the μ and σ for each data element (hour) across all observations (days). This will result in 24 values for μ and σ. For the sequence-based normalization scope (right), each observation (day) can be normalized independently, resulting in 365 values of μ and σ.

After normalization, the *assignment* step assigns similar periods to clusters. Assignment can be performed using a variety of clustering methods. Clustering methods differ in three ways. First, each clustering method requires a distance measure to represent goodness of fit of an observation to a cluster (e.g., Euclidean distance). Second, clustering methods use an algorithm to assign observations to clusters (e.g., partitional or hierarchical). Lastly, they vary in the cluster center used in the calculation (e.g., centroid or medoid).

The third step in the framework is *representation*. Each cluster must be represented by an observation and an associated weight, allowing for the desired data reduction. In existing methods, both the cluster centroid and cluster medoid have been used to represent the cluster in the optimization problem. Finally, the representative period must be expressed in the units of the original data, thus the initial normalization must be undone by denormalization.

Note that the assignment and representation steps are often viewed as connected, though in general they need not be. For example, the cluster center $\boldsymbol{c}$ used in the algorithm is often, but not always, chosen as the cluster representative $\boldsymbol{r}$. For example, the k-means clustering algorithm uses the cluster centroid as its center, and a natural choice is to represent the cluster with the resulting centroid. However, there are also approaches that use k-means clustering, and then choose the medoid of each resulting cluster as the cluster representative (4).

3. Clustering Methods

In this section, we introduce the clustering methods that are compared in this paper for the purpose of finding representative periods, which we from here on refer to as representative days.

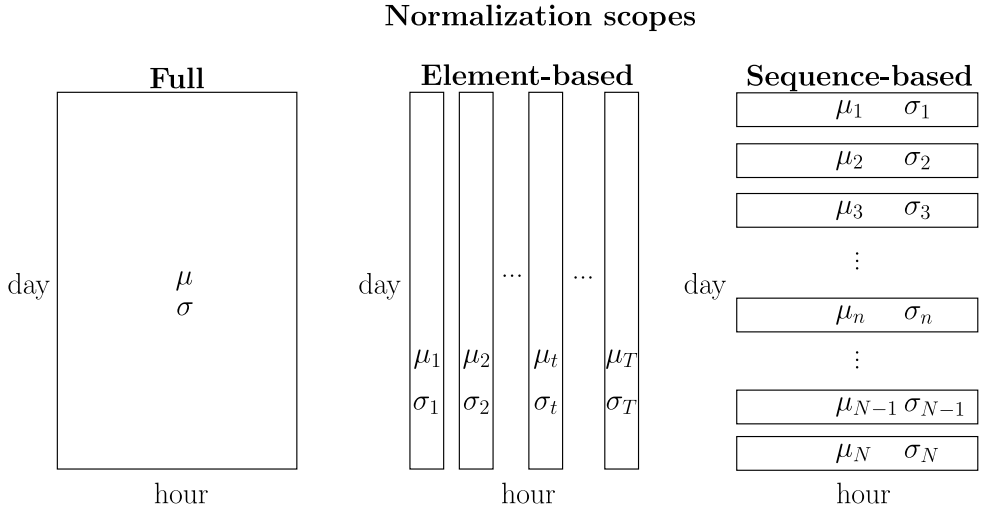


Figure 2: Illustration of the normalization scopes: full normalization, element-based normalization, sequence-based normalization.

We aggregate a time-series consisting of N daily price vectors $\boldsymbol{p}_1 ... \boldsymbol{p}_N$ (each is a day of T hourly prices) into K clusters. We introduce the following sets for notation: $\mathcal{N} = \{1, ..., N\}$ is the set of daily indices, $\mathcal{T} = \{1, ..., T\}$ is the set of hourly time-step indices within each day, $\mathcal{K} = \{1, ..., K\}$ is the set of cluster indices, and C_k with $k \in \mathcal{K}$ are the disjoint sets of daily indices belonging to the respective clusters k. It holds that $\bigcup_{k \in \mathcal{K}} C_k = \mathcal{N}$.

We first introduce the different normalization methods. We then describe the clustering methods by their assignment step (distance measure dist(), clustering algorithm, center $\boldsymbol{c}$). In our comparison, we consider partitional clustering methods (k-means, k-medoids, DBA clustering, k-shape), and hierarchical clustering methods (agglomerative hierarchical clustering using Ward's algorithm). An overview of the methods considered in this work is shown in Table 1.

3.1. Normalization

We use the z-normalization in all cases below. Z-normalization is commonly applied in statistics before clustering algorithms are applied (28). It shifts the mean to zero and the standard deviation to one. The three normalization scopes (full normalization, element-based normalization, sequence-based normalization) considered are shown in Figure 2. All clustering methods are used with z-normalization on all normalization scopes (except for k-shape, which only works with sequence-based normalization). Normalization transforms each daily price vector $\boldsymbol{p}_i \in \boldsymbol{P}$, $\boldsymbol{P} \in \mathbb{R}^{N \times T}$, to the normalized daily price vector $\hat{\boldsymbol{p}}_i \in \hat{\boldsymbol{P}}$, $\hat{\boldsymbol{P}} \in \mathbb{R}^{N \times T}$.

Z-normalization is applied as follows:

$$\hat{\boldsymbol{P}} = \frac{1}{\sigma}(\boldsymbol{P} - \mu) \tag{1}$$

where μ is the mean and σ is the standard deviation. Using the full normalization scope, μ and σ are scalars. For element-based normalization scope, they are vectors $\boldsymbol{\mu}_{elem}, \boldsymbol{\sigma}_{elem} \in \mathbb{R}^T$, while for sequence-based normalization scopes they are vectors $\boldsymbol{\mu}_{seq}, \boldsymbol{\sigma}_{seq} \in \mathbb{R}^N$ (please refer to the SI for implementation details). After applying clustering, we obtain K normalized cluster representations $\hat{\boldsymbol{r}}_k \in \hat{\boldsymbol{R}}$, $\hat{\boldsymbol{R}} \in \mathbb{R}^{K \times T}$. We obtain the denormalized cluster representations $\boldsymbol{r}_k \in \boldsymbol{R}$, $\boldsymbol{R} \in \mathbb{R}^{K \times T}$ by

$$\boldsymbol{R} = \sigma \hat{\boldsymbol{R}} + \mu \tag{2}$$

For time-series clustering using shape-based methods such as DBA clustering and k-shape, sequence-based normalization is commonly used (29). However, these methods are generally used in contexts where only the assignment matters in the analysis. Our methods require the use of the denormalized representation. Because we obtain a mean and standard deviation for each day, but only obtain K representations, we cannot utilize the means and standard deviations directly. We thus introduce denormalization for sequence-based clustering based on averaged means and standard deviations of elements of the cluster:

$$\boldsymbol{R} = \hat{\boldsymbol{R}} \ \mathrm{diag}(\tilde{\boldsymbol{\sigma}}_{seq}) + \mathbf{1}\tilde{\boldsymbol{\mu}}_{seq}^T \tag{3}$$

where $\tilde{\boldsymbol{\mu}}_{seq} = \frac{1}{K} \sum_{i \in C_k} \mu_{seq,i}$ and $\tilde{\boldsymbol{\sigma}}_{seq} = \frac{1}{K} \sum_{i \in C_k} \sigma_{seq,i}$.

3.2. Distance measures

Distance measures (signified dist) are used to compute the dissimilarity between two time-series vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^T$. In this work, we consider three distance measures: Euclidean distance ED($\boldsymbol{x}, \boldsymbol{y}$), dynamic time warping DTW($\boldsymbol{x}, \boldsymbol{y}$), and shape based distance SBD($\boldsymbol{x}, \boldsymbol{y}$).

Euclidean distance computes the distance between two time-series vectors (in our case days) based on the l_2-norm

$$\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{ED}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{t \in \mathcal{T}} (x_t - y_t)^2} \tag{4}$$

It compares each hour t of one day with the respective hour t of the other day (see Figure 3).

Dynamic time warping (DTW) (30) takes into consideration that the shape of two time-series vectors can be similar but

Table 1: Overview of clustering methods considered in this work.

Algorithm	Distance measure dist()	Center $\boldsymbol{c}$	Cluster representation $\boldsymbol{r}$	Name	Previously applied in
partitional	ED	centroid	centroid	k-means	(1; 5; 7; 12–14) (9; 10; 16–19; 27)
partitional	ED	centroid	medoid		(4)
partitional	ED	medoid	medoid	k-medoids	(8; 11; 20) (12; 14; 15)
partitional	DTW	centroid	centroid	DBA	
partitional	SBD	centroid	centroid	k-shape	(26)
agglomerative	ED	centroid	centroid	hierarchical - centroid	
agglomerative	ED	centroid	medoid	hierarchical - medoid	(2; 3; 12)

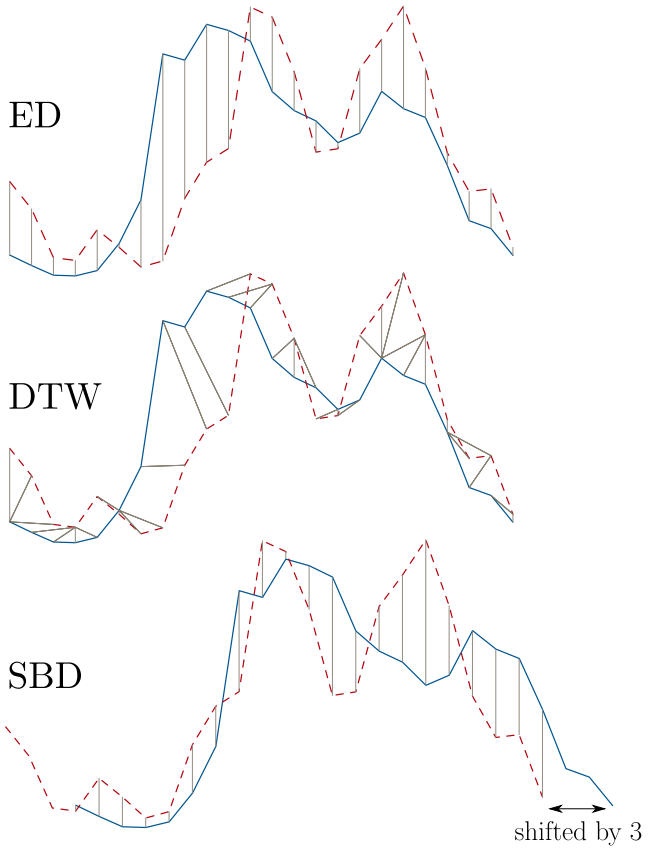


Figure 3: Visualization of the distance measures Euclidean distance (ED), dynamic time warping (DTW), and shape based distance (SBD). Electricity price shapes from Germany, January 23, 2015 (blue-solid) and January 24, 2015 (red-dashed). ED aligns each hour of one day with the respective hour of the other day, whereas DTW is able to align any hour of one day to any hour of another day within a certain bandwidth b (here, $b = 2$). Both use an l_2-norm minimization based measure. SBD finds similarities in shape by sliding one day (blue-solid) compared to the other. It uses a cross-correlation maximization based measure.

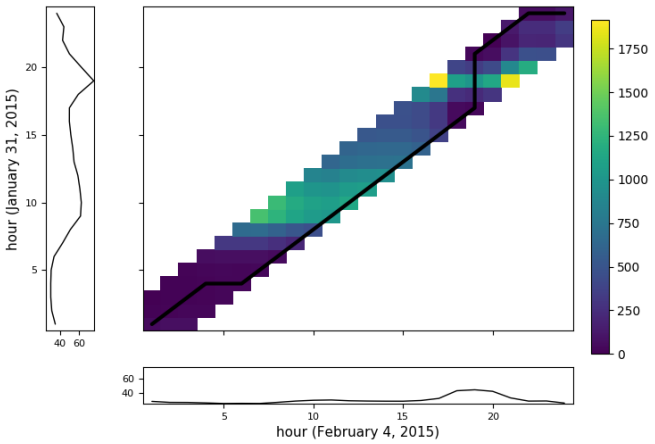


Figure 4: Dynamic time warping with bandwidth $b = 2$ distance matrix visualization. Two exemplary price profiles are shown, and the corresponding distance matrix $\boldsymbol{D}$ in color. The path of minimum cost through the matrix aligns the two time-series.

shifted. It can be seen as an extension of the Euclidean distance, where it compares any hour of one day to any hour of the other day, instead of only the same hours. This is illustrated in Figure 3, where hours with similar prices are aligned.

One can imagine a multitude of possible alignments between two time-series vectors. The optimal alignment is calculated as follows and illustrated in Figure 4. The squared distances of all hours of $\boldsymbol{x}$ to all hours of $\boldsymbol{y}$ are stored in a matrix $\boldsymbol{D} \in \mathbb{R}^{T \times T}$, where every entry D_{ij} of the matrix is the squared error between x_i and y_j (illustrated by the color in Figure 4. Any path through the matrix starting at $D_{1,1}$ and ending at $D_{T,T}$ describes an alignment of the two time series. This path is called the warping path and can formally be described as $W = \{w_1 ... w_L\}$ of elements $w_l \in W$, $w_l = D_{ij}$, $l \in \{1, ..., L\}$, $L \geq T$ through the matrix D starting at $D_{1,1}$ and ending at $D_{N,N}$ such that i, j, or both i and j increase for each step l of the path W. The cost of a particular warping path is the square root of sum of its entries $\sum_{l=1}^{L} w_l$, and the DTW distance is the path of

minimum cost amongst all possible paths:

$$\text{dist}(\boldsymbol{x}, \boldsymbol{y}) = \text{DTW}(\boldsymbol{x}, \boldsymbol{y}) = \min_{W} \sqrt{\sum_{l=1}^{L} (w_l)} \tag{5}$$

Finding the path of minimum cost through the matrix is done recursively.

Furthermore, the warping path can be constrained to remain within a so-called "warping window," where only a subset of the matrix $\boldsymbol{D}$ with bandwidth b is evaluated, i. e., elements of the warping path $w_l = D_{ij}$ are constrained by $|i - j| \leq b$, to find the path between the time-series vectors (30). This reduces computational time significantly and has been shown to improve clustering performance (24). Note that the Euclidean distance is a special case of dynamic time warping, where the warping window $b = 0$, and therefore the cost path is the diagonal through the matrix $\boldsymbol{D}$.

DTW is the major distance measure used in classification tasks of time-series data and has been shown to do well in clustering tasks (24).

The shape based distance (SBD) has recently been proposed by Paparrizos and Gravano (23). It compares the shape of two time-series vectors by sliding one of them against the other as illustrated in Figure 3. One can measure the distance between two slided time series in terms of cross correlation, which is calculated through the inner product between two time series. If no sliding occurs, cross correlation is the inner product $\boldsymbol{x} \cdot \boldsymbol{y}$, and if sliding occurs, cross correlation is the inner product between the two portions of the vectors that overlap. Higher similarities in shape lead to higher cross correlation. Thus, SBD is the maximum cross correlation [1] that occurs among all possibilities of sliding the two vectors amongst each other.

3.3. Partitional clustering algorithms

All methods based on partitional clustering algorithms compared in this work minimize the within-cluster sum of squared distances (SSD) between the cluster members and the cluster center $\boldsymbol{c}$:

$$C_1^* ... C_K^* = \underset{C_1 ... C_K}{\text{argmin}} \sum_{k \in \mathcal{K}} \sum_{i \in C_k} \text{dist}(\boldsymbol{p}_i, \boldsymbol{c}_k)^2 \tag{6}$$

These methods differ in their choice of distance measure dist and cluster center $\boldsymbol{c}$, but they can be solved using a similar general algorithm. The partitional clustering algorithm is a generalization of *Lloyd's* algorithm (31) and the k-means algorithm (32; 33). To begin, the cluster centers are initialized randomly. Then, the algorithm iteratively performs the following two steps until it converges (i. e., there are no changes in cluster assignments) or reaches a maximum number of iterations: In the first step (assignment), each daily price vector $\boldsymbol{p}_i$ is assigned to the closest cluster center $\boldsymbol{c}$ based on the distance measure dist (Equation 7 with fixed $\boldsymbol{c}_k$), and in the second step (refinement), the cluster centers $\boldsymbol{c}$ are updated in order to reflect the changes in cluster assignments. Each center is updated to the $\boldsymbol{z}$ that minimizes the within-cluster distance:

$$\boldsymbol{c}_k = \underset{z}{\text{argmin}} \sum_{i \in C_k} \text{dist}(\boldsymbol{p}_i, \boldsymbol{z})^2 \tag{7}$$

Cluster centers can be based on the centroid or medoid. The centroid is any artificial vector $\boldsymbol{z} \in \mathbb{R}^T$, whereas the medoid is an actual vector $\boldsymbol{z} \in \{\boldsymbol{p}_1, ..., \boldsymbol{p}_N\}$.

The algorithm finds the cluster assignments C_k. Unless noted otherwise, we initialize the algorithms below with k-means++ (34). The algorithm yields locally converged solutions based on the random cluster initializations. In this work, we run all clustering algorithms until convergence by testing 10,000 local starting points and do not consider any non-converged solutions unless noted otherwise.

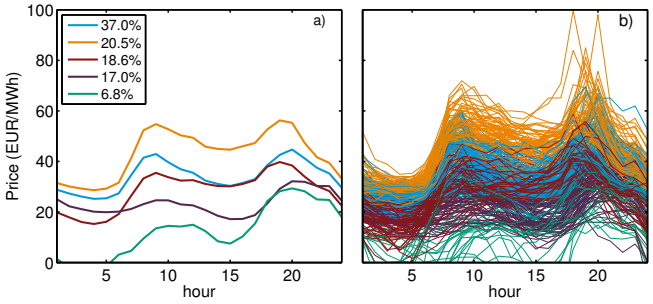


Figure 5: k-means on Germany data (k=5): (a) example cluster representations and respective weights, and (b) the respective assignments of the original data.

3.3.1. k-means clustering

k-means clustering (32; 33) minimizes the Euclidean distance ($\text{dist}(\boldsymbol{p}_i, \boldsymbol{z}) = \text{ED}(\boldsymbol{p}_i, \boldsymbol{z})$). k-means clustering uses the centroid as cluster center, which can be computed with the arithmetic mean. It is commonly used with the centroid as the representation, but has also been used with the medoid as the representation in the context of energy systems optimization. In this work, we consider both representations. Figure 5 shows exemplary centroid representations and the underlying data from k-means clustering.

3.3.2. k-medoids clustering

Similarly to k-means clustering, k-medoids clustering uses the Euclidean distance as distance measure ($\text{dist}(\boldsymbol{p}_i, \boldsymbol{z}) = \text{ED}(\boldsymbol{p}_i, \boldsymbol{z})$). However, the cluster center is its medoid.

Besides being solved with the partitional clustering algorithm outlined initially, the k-medoids clustering problem can be formulated as a Binary Integer Program (BIP) as introduced by (35) and used by (15). Please refer to the SI for the full formulation. The BIP formulation is generally NP-hard, but it can be solved to near global optimality for the number of days considered in this study (optimality gap $\leq 0.01\%$). In this work, we

[1] Because SBD as a distance measure by definition measures dissimilarity, SBD is formally defined as the negative of the cross correlation CC: $\text{dist}(\boldsymbol{x}, \boldsymbol{y}) = \text{SBD}(\boldsymbol{x}, \boldsymbol{y}) = 1 - \max_s \frac{CC_s(\boldsymbol{x}, \boldsymbol{y})}{\sqrt{R_0(\boldsymbol{x}, \boldsymbol{x}) R_0(\boldsymbol{y}, \boldsymbol{y})}}$, which finds the sliding between the two vectors that maximizes cross correlation CC. R is the autocorrelation function and the denominator ensures proper normalization. For a more detailed description of SBD, see Paparrizos and Gravano (23).

use both the partitional clustering algorithm and the BIP formulation.

3.3.3. DBA clustering

Dynamic time warping barycenter averaging (DBA) clustering (22) takes into consideration that the shape of two time series can be similar stretched and contracted. It is a centroid-based partitional clustering algorithm that uses Dynamic Time Warping (DTW) (30) as its distance measure (dist($\boldsymbol{p}_i, \boldsymbol{z}$) = DTW($\boldsymbol{p}_i, \boldsymbol{z}$)). This means that the iterative procedure is similar to the one of the k-means clustering method, but instead of Euclidean distance, the DTW distance is used to assign different days to given cluster centers, and the cluster centers are updated using barycenter averaging instead of arithmetic mean.

The DBA cluster update of centroid $\boldsymbol{c}_k$ using barycenter averaging was introduced by Petitjean et al. (22) and to date is the most efficient and accurate way to find a centroid for the DTW distance measure (23). It is an iterative process and works as follows: First, the DTW distances between all time-series in cluster k and the current centroid $\boldsymbol{c}_k$ are calculated, and the DTW paths are saved. Then, each element c_{ki} is updated as the arithmetic mean of all elements in the time series in cluster k that are connected to element c_{ki}. This is repeated until convergence. For more details on the implementation of the algorithm, see Petitjean et al. (22).

3.3.4. k-shape clustering

k-shape clustering has recently been proposed as an alternative to DTW as a shape-based clustering method (23). It is a centroid-based partitional clustering method and uses the shape-based distance (dist($\boldsymbol{p}_i, \boldsymbol{z}$) = SBD($\boldsymbol{p}_i, \boldsymbol{z}$)). In the iterative procedure of partitional clustering, SBD is used to assign different days to given cluster centers, and the centers are updated according to the center definition in Equation 7, where $\boldsymbol{z} \in \mathbb{R}^T$. Paparrizos and Gravano rewrite the optimization problem stated in Equation 7 with dist($\boldsymbol{p}_i, \boldsymbol{z}$) = SBD($\boldsymbol{p}_i, \boldsymbol{z}$) such that it is in the form of the maximization of the Raleigh Quotient, which can be solved analytically. This makes the center update computationally efficient.

3.4. Hierarchical clustering algorithm

We consider the most common type of hierarchical clustering: agglomerative hierarchical clustering. We start with $k = N$ clusters and proceed by merging the two closest observations into one cluster, obtaining $k = N - 1$ clusters. The process of merging two clusters to obtain $k - 1$ clusters is repeated until we reach the desired number of clusters K. This produces a hierarchy of cluster assignments, where each level of the hierarchy (k clusters) is created by merging clusters from the next lower hierarchy ($k + 1$ clusters) (36). Which clusters to merge is decided by minimizing the total within-cluster variance. This is done using Ward's algorithm (37). In each step, the centroid of each cluster k is calculated according to Equation 7. Then, for all combinations of clusters i and j, the Euclidean distance ED($\boldsymbol{c}_i, \boldsymbol{c}_j$) is calculated and the two clusters i and j that yield the minimum Euclidean distance are merged. After the assignments for the chosen number of clusters is obtained, we need to find a representation. In this work, we compare the centroid and the medoid as cluster representations.

Note that choosing the medoid as cluster representation does not mean that this algorithm is a greedy version of the above introduced k-medoids algorithm. The k-medoid algorithm uses medoids as cluster centers for the minimization of the Euclidean distance. In contrast, the hierarchical clustering algorithm uses centroids as cluster centers for the minimization of the Euclidean distance, and chooses the medoid only as a cluster representation.

Hierarchical clustering is deterministic, which means it is reproducible. However, it is also greedy, which means that it yields local solutions with sets of cluster assignments $C_1 ... C_K$ that do not necessarily satisfy optimality in Equation 6.

3.5. Representation

Each cluster is represented in the optimization problem by a representative period and an associated weight. The representative period is often, but not always the same as the choice of cluster center in the assignment step. For representative periods based on the medoid, the weighted mean of the representative periods $\boldsymbol{r}_1 ... \boldsymbol{r}_k$ may be different than the mean of the full time series $\boldsymbol{p}_1 ... \boldsymbol{p}_N$. Thus, we rescale the medoid-based representations by a scaling factor s as introduced in (2; 12):

$$s = \frac{\sum_{n \in \mathcal{N}} \sum_{t \in \mathcal{T}} p_{n,t}}{\sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} |C_k| \, r_{k,t}} \tag{8}$$

4. Optimization Problems

In order to test the impacts of the above clustering choices, we generate simple operational problems formulated as linear programs that can be solved directly for 365 days of hourly data. This solution gives us the true value of the objective function without error introduced by clustered input data, against which the value of the objective function with clustered input data can be compared. We create two optimization problems: (1) an electricity storage problem for optimal energy price arbitrage; (2) a gas turbine power generation problem for optimal power dispatch.

The electricity storage problem is formulated as follows:

$$\begin{aligned}
&\max_{\underline{E}, \bar{E}, S} \sum_{k \in \mathcal{K}} N_k \sum_{t \in \mathcal{T}} (\underline{E}_{k,t} - \bar{E}_{j,t}) p_{k,t} \\
&\text{s.t.} \\
&0 \le \underline{E}_{k,t} \le P_{max} \Delta t && \forall\, k \in \mathcal{K},\ t \in \mathcal{T} \\
&0 \le \bar{E}_{k,t} \le P_{max} \Delta t && \forall\, k \in \mathcal{K},\ t \in \mathcal{T} \\
&0 \le S_{k,t} \le E_{max} && \forall\, k \in \mathcal{K},\ t \in \mathcal{T} \\
&S_{k,t+1} = S_{k,t} + \eta_{in} \bar{E}_{k,t} - \frac{\underline{E}_{k,t}}{\eta_{out}} && \forall\, k \in \mathcal{K},\ t \in \mathcal{T} \\
&S_{k,1} = S_{k,T+1} \wedge S_{1,1} = S_{k,1} && \forall k \in \mathcal{K}
\end{aligned} \tag{9}$$

Here, $\underline{E}_{k,t}$ and $\bar{E}_{k,t}$ is hourly electric energy that flows out or into the battery from the market — they can also be seen as average power —, $p_{k,t}$ is the hourly electricity price, and $N_k = |C_k|$ is the number of days in cluster k. P_{max} is the maximum power, and

E_{max} is the maximum storage capacity of the battery. S_{kt} is the amount of energy stored in the battery, and η_{in} and η_{out} are the charge and discharge efficiencies. Here we assume maximum power of 100 MW and maximum storage capacity of 400 MWh, with charge and discharge efficiencies at 95%. Note that when we solve this problem for the full representation, we solve with $k = N = 365$ days. This assumes that the storage level at the beginning of each day is the same.

The gas turbine dispatch problem is formulated as follows:

$$\begin{aligned} &\max_{\underline{E}} \sum_{k\in\mathcal{K}} N_k \sum_{t\in\mathcal{T}} \underline{E}_{k,t} p_{k,t} - \frac{\underline{E}_{k,t}}{\eta_{gt}} p_{gas} \\ &\text{s.t.} \\ &0 \le \underline{E}_{k,t} \le P_{max}\Delta t \qquad\qquad \forall\, k \in \mathcal{K},\ t \in \mathcal{T} \end{aligned} \tag{10}$$

Here, $\underline{E}_{k,t}$ is the hourly electric energy generated from the gas turbine, and p_{gas} is the gas price. We assume a gas price of 4 \$/GJ for California and 6.8 \$/GJ for Germany, and assume a combined cycle efficiency η_{gt} of 60%. Using a single-cycle gas turbine with 40% efficiency does not change the findings qualitatively.

5. Data

The above-described clustering methods are evaluated using two electricity price data sets from 2015. Both are hourly data from the day-ahead market, one from a price node in Northern California and the other from the German country-wide price. The mean price is 30.4 \$/MWh in California and 31.6 EUR/MWh in Germany, and the standard deviation is 13.0 \$/MWh in California and 12.7 EUR/MWh in Germany. A visual example of the data can be found in (18) and a visual example of cluster outcomes on the data is shown in Figure 5.

6. Results

We calculate the resulting objective function value of the respective optimization problem for the full representation (N=365 days of electricity prices) and for each clustering option. We then compare the results in objective function value between the full representation and the clustered representation.

6.1. Clustering cases explored

The clustering cases we explore include all of the methods outlined in Table 1. For each case we perform clustering for $k = 1 \ldots 9$ representative days. The results shown in this range show all the indicative characteristics of optimization results with higher number of clusters k. We ensure in each case that the solutions obtained for a particular k are close to the global optimum in clustering distance measure by re-starting each clustering operation with 10,000 randomly initialized cluster assignments and taking the clustering result with the lowest locally converged distance measure of all of these trials.

6.2. Clustering method comparison

Figure 6 shows the objective function value for a given k ($k \ll N$) and clustering method, normalized by the objective function value for the full representation. This is performed for both the battery problem and the gas turbine problem using electricity price data from both Germany (GER) and California (CA). Centroid-based (first row), medoid-based (second row), and shape-based clustering methods (third row) are compared in the figure.

The figure presents the methods using the normalization scope that they are conventionally used with: The methods using centroid and medoid as representation use full normalization scope, and the shape-based methods use sequence-based normalization scope. The results comparing all clustering methods on all normalization scopes are presented in the SI.

Figure 6 allows several observations, taking as an example the battery optimization problem and Germany electricity price data (column 1). First, centroid-based methods (top row) perform similarly: they increase in objective function value with increasing k, and one single cluster captures approximately 75% of the objective function value of the full representation. The objective function value of both k-means clustering and hierarchical clustering with the centroid always underestimate the objective function value of the full representation. This can be explained by Theorem 1. The proofs for Theorems 1-3 can be found in the SI.

Theorem 1. *For an LP, if data is clustered in the objective function coefficient vector* $\boldsymbol{c}$*, the constraints are structurally the same for all periods, and the cluster representation is centroid based, then the optimization problem with full input data is a relaxation of the optimization problem with clustered input data.*

Similarly, the following holds for linear programs with data clustered in the constraint coefficient vector (such as heating or cooling demand).

Theorem 2. *For an LP, if data is clustered in the constraint coefficient vector* $\boldsymbol{b}$*, the corresponding constraints are structurally the same, and the cluster representation is centroid based, then the optimization problem with clustered input data is a relaxation of the optimization problem with full input data.*

In Figure 6, we further observe that hierarchical clustering with the centroid as its representation increases monotonically in objective function value with an increasing number of clusters. This also is a general property.

Theorem 3. *For an LP, if data is clustered in the objective function coefficient vector* $\boldsymbol{c}$ *or in the constraint coefficient vector* $\boldsymbol{b}$ *with structurally similar constraints for all periods, and if the cluster representation is centroid based and the clustering algorithm is hierarchical clustering using Ward's algorithm, then the objective function value* z_k *is a monotonic function of the number of clusters* k*.*

k-medoids and hierarchical clustering with the medoid as its representation perform less predictably. As k increases, the objective function either increases toward or decreases away from

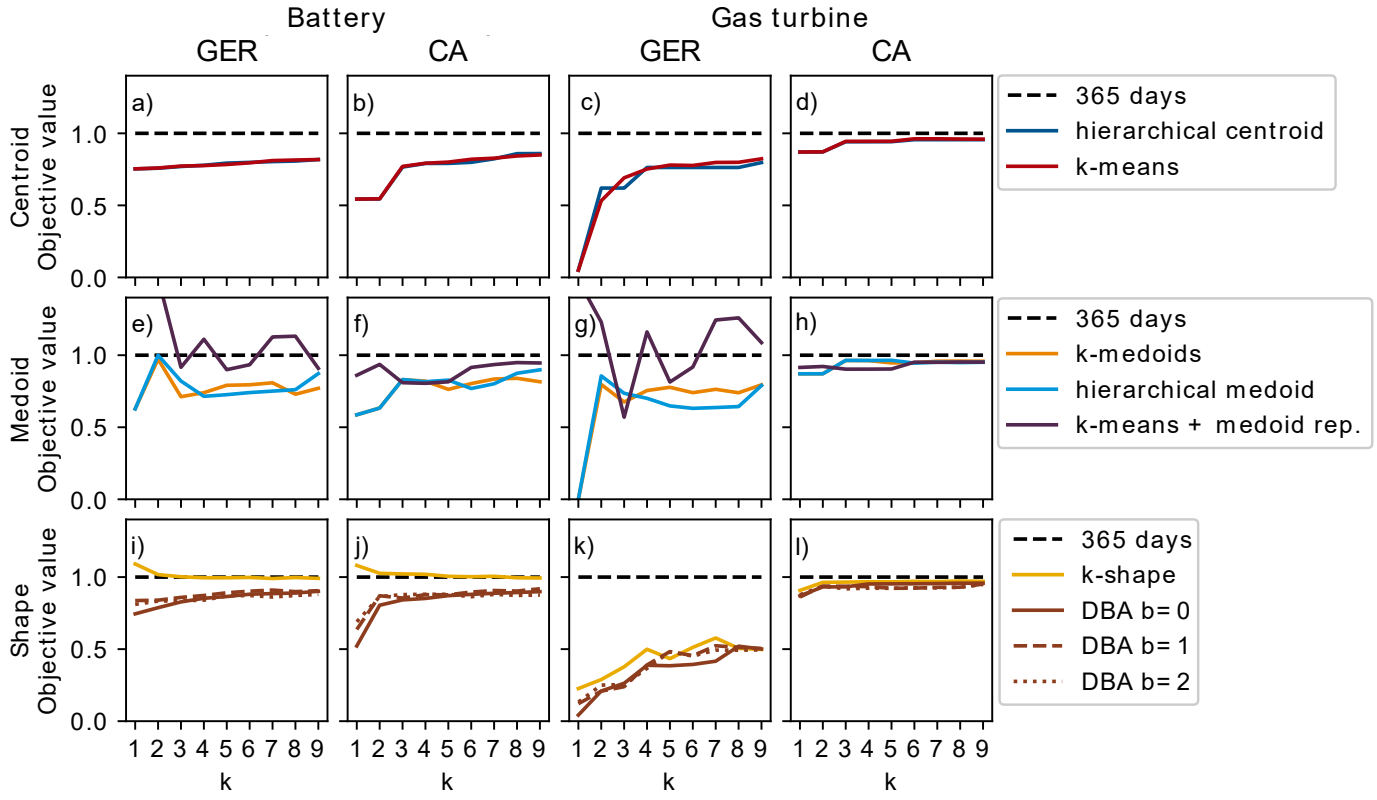


Figure 6: Objective value (revenue) as a function of the number of clusters for the different clustering methods. The methods using centroid and medoid as representation use the full normalization scope, and the shape-based methods use the sequence-based normalization scope.

the full representation, with no clear pattern emerging. If the full representation is unknown, choosing an appropriate k in a specific case study is challenging. k-means clustering with the medoid as representation performs even less predictably, showing inconsistent operational representation as the number of clusters increases.

The error due to data reduction that we observe with centroid-based and medoid-based methods can be persistent in direction (bias) or random (noise). Overall, we observe that centroid-based methods have low noise, but underestimate the objective function value of the full representation (bias). On the other hand, we also observe bias but additionally noise for medoid-based methods. We also observe that bias and noise are not only method dependent, but also depend on the data: k-means with the medoid as its representation results in low bias but high noise on problems with Germany data (Figures 6e and 6g), but lower noise and higher bias on problems with California data (Figures 6f and 6h).

k-shape performs well on the battery optimization problem, representing the full representation very well with only two representative days. Because the objective function value of the battery optimization problem depends on the arbitrage potential between hours, the characteristics of this problem are based on the difference of the electricity price between hours and does not depend on the overall level of the price itself. These are characteristics that the k-shape method prioritizes when forming clusters and thus it performs well. DBA clustering does not perform as well and does not show significant differences when used with different bandwidths. Even though DBA clustering is also shape based, this performance may be because DTW can assign hours in ways that distort physical meaning.

The DTW alignment of two example days in Figure 3 shows that the time series is stretched and contracted, whereas k-shape shifts time series relative to each other. Shifting may better retain the information relevant to storage optimization problems than stretching and contracting. Note that DBA clustering with a bandwidth of zero is equivalent to k-means clustering with sequence-based normalization.

The battery optimization problem evaluated on the California data yields results similar to those of the Germany data for all clustering methods.

On the gas turbine optimization problem for the Germany data, clustering methods do not in general yield good objective function values, but they improve as the number of clusters increases. For the medoid-based methods the gas turbine problem exhibits behavior similar to the behavior in the battery problem, performing much less predictably. This is significant because in problems evaluating design and operations jointly, medoid based approaches have been suggested to capture more variability in the data. However, we show that this comes at the trade-off of less predictable representation of the operational domain.

Shape-based methods do not perform well on the gas turbine problem with Germany electricity price data because they extract characteristics from the data not relevant to that problem. However, all methods perform well on the gas turbine problem with California data. This is because the natural gas price in California is lower than in Germany, and the turbine is profitably generating power for most of the time. This shows that even on the same optimization problem, different input data can yield significantly different results and stresses the need to evaluate methods on different data sets.

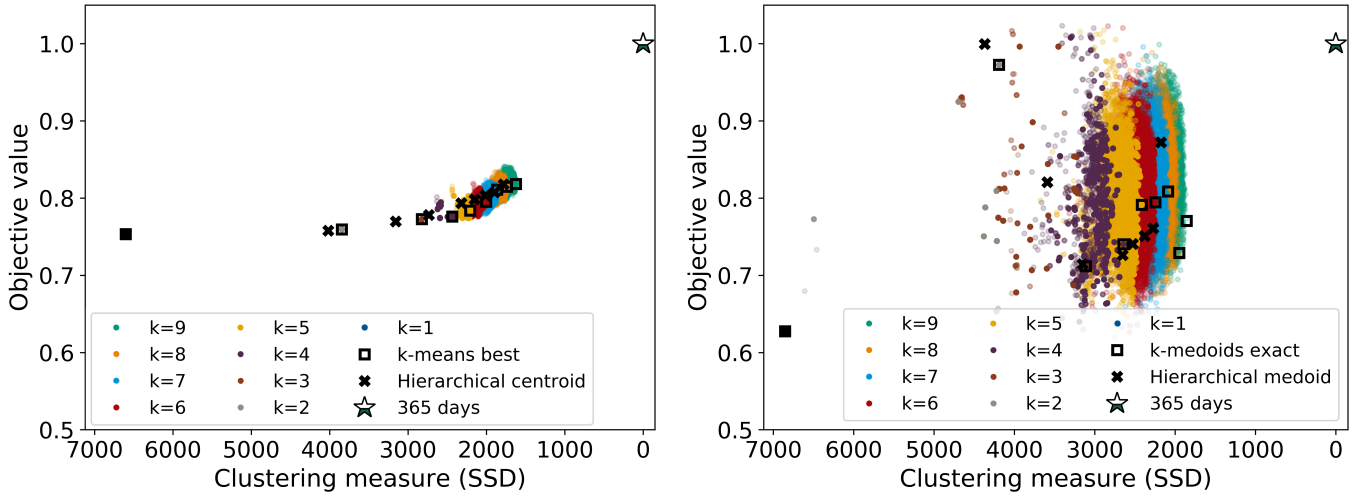
An analysis of the impact of different normalization scopes on the optimization results can be found in the SI. Also note that we tested but do not show fuzzy c-means (38), but the method does not perform well. Due to fuzzy cluster assignments, the clusters are averaged even more strongly and do not exhibit much price variability, which is important to capture for the problems at hand.

6.3. Local convergence of k-means and k-medoids

Figure 7 shows the local convergence of 10,000 initial points for each k for the k-means (a) and k-medoids (b) methods for the battery problem using Germany data, using full normalization. Figure 7a allows for a more detailed analysis of Figure 6a, and Figure 7b allows for a more detailed analysis of Figure 6e. In Figure 7a, hierarchical with centroid representation and best k-means in terms of SSD correspond to Figure 6a, and in Figure 7b, hierarchical with medoid representation and k-medoids exact correspond to Figure 6e. The results shown here look qualitatively similar for both optimization problems (battery and turbine) and for both datasets (CA and GER). On the x-axis, each colored dot is plotted at the value of minimum clustering measure (SSD) obtained after convergence at each initial guess. The y-axis coordinate is the resulting objective function value after running the clusters through the optimization problem.

In the upper right, the star shape at 0 clustering error and 1.0 relative objective function value indicates the full representation (i.e., $k = 365$, or no clustering). Improved clustering methods should aim to get as close to this point with the smallest value of k. The best solution in clustering error measure (most rightward) for each value of k is boxed. The 10,000 locally converged solutions for k-medoids are generated by a partitional clustering algorithm. We see that for this large number of initial points, the best solution in terms of the clustering measure of this algorithm is the same as the solution to the exact k-medoids formulation.

Both the k-means and k-medoids exhibit movement to the right as k increases. That is, clustering error is reduced with increased k. However, the methods exhibit quite disparate performance in recreating the full representation objective function value. k-means solutions reliably increase to the upper right, and in general a reduction in clustering measure error is associated with a more accurate objective value. However, for k-medoids, there is high variance and weak correspondence between clustering measure and objective value. All values of $k \geq 2$ contain assignments with objective value near 1 as well assignments with objective value near 0.7, with little difference in clustering measure between them. That is, one is not guaranteed that the k-medioids solution with minimum clustering error will perform any better in optimization than another randomly chosen k-medoids solution for the same k. This explains



(a) *k*-means

(b) *k*-medoids

Figure 7: Revenue vs. clustering measure (SSD) for the battery optimization problem on Germany data of (a) the centroid-based and (b) the medoid-based clustering methods. 10,000 initial points for (a) *k*-means and (b) *k*-medoids for each number of representative days *k*. (a) Hierarchical with centroid representation and best k-means in terms of SSD correspond to Figure 6a, and (b) hierarchical with medoid representation and k-medoids exact correspond to Figure 6e. We observe the difference in variance in objective function value between centroid- and medoid-based representations.

why the objective functions resulting from medoid-based clustering methods perform less predictably than the centroid-based methods. A similar observation can be made for the hierarchical clustering with representation as a centroid or medoid (note as well that we observe that hierarchical clustering does not satisfy optimality in terms of clustering measure as described in Section 3.4).

When we compare the locally converged solutions of *k*-means (Figure 7a) and *k*-medoids (Figure 7b), we observe that the *k*-means solutions span a lower range in terms of objective function value than the *k*-medoids solutions. Furthermore, we observe that the *k*-means objective function values are always below the objective function value (as shown in Proof 1), whereas this is not guaranteed for the objective function based on *k*-medoids clustering.

This large spread in locally converged solutions is especially significant for the *k*-medoids clustering, where the objective function values span a significant range: the objective function values of all locally converged solutions do not generally increase with increasing number of clusters, and no clear pattern emerges as to what the best solution of the *k*-medoids method represents in terms of objective function value.

7. Conclusion

It is often computationally intractable to solve design and operations optimization problems of energy systems for a full set of time series input data. In this work, we presented a framework for clustering methods for finding representative periods for these optimization problems that classifies aggregation methods by normalization method, clustering method, and cluster representation. By describing the choices for each of the parts of the framework, future studies that employ clustering will enhance understanding and evaluation of modeling results. We furthermore presented a comparison of conventionally used and shape-based clustering algorithms, and how they affect the operational representation in terms of the objective function value of the full representation.

Our analysis shows that centroid-based clustering methods represent the operational part of the optimization problem more predictably than medoid-based approaches. They can also be shown to reliably underestimate objective value. Whereas it has previously been suggested that medoid based approaches capture variability important to the design part of complex optimization problems, this comes at the trade-off of less predictable representation of the operational domain.

Comparing different normalization scopes, we found that the conventionally used clustering methods perform best with full normalization or hourly normalization, whereas shape-based clustering methods perform best when used with a shape-based normalization scope. On certain problems that exploit intra-daily variability, such as battery problems, we showed that *k*-shape improves performance significantly over conventionally used clustering methods.

Comparing all locally converged solutions of the clustering methods, we showed that a better representation in terms of clustering measure is not necessarily better in terms of objective function value of the optimization problem. This underscores that clustering performance should be evaluated in the domain of the objective function value of the optimization problem instead of evaluating the clustering error itself.

This paper provides an initial framework for clustering methods and a comparison on two operational optimization problems. Future research could include the investigation of extreme period selection. While representative periods obtained solely from clustering are often smooth, extreme periods can

add variability that is contained in the original time series. Furthermore, it would be interesting to compare the clustering methods on multiple additional design and operations problems.

Acknowledgment

This work was supported by the Wells Family Stanford Graduate Fellowship for HT. The Stanford Center for Computational Earth and Environmental Science (CEES) provided the computational resources used in this work.

Nomenclature

$\bar{E}$	Hourly electric energy inflow
$\boldsymbol{c}$	Cluster center
$\boldsymbol{D}$	DTW distance matrix
$\boldsymbol{P}$	Matrix of daily price vectors
$\boldsymbol{p}_i$	Daily price vector
$\boldsymbol{R}$	Matrix of representative days
$\boldsymbol{r}$	Cluster representative
$\hat{\boldsymbol{P}}$	Matrix of normalized daily price vectors
$\hat{\boldsymbol{p}}$	Daily normalized price vector
$\hat{\boldsymbol{R}}$	Matrix of normalized representative days
$\hat{\boldsymbol{r}}$	normalized representative day
$\underline{E}$	Hourly electric energy outflow
dist()	Distance measure
η_{gt}	Combined cycle efficiency
η_{in}	Battery inflow efficiency
η_{out}	Battery outflow efficiency
C_k	Set of daily indices belonging to cluster k
$\mathcal{K}$	Set of cluster indices
$\mathcal{N}$	Set of daily indices
$\mathcal{T}$	Set of hourly indices
μ	mean
σ	standard deviation
b	DTW bandwidth
E_{max}	Maximum energy capacity
N_k	Number of days in cluster k
p_{gas}	Natural gas price
P_{max}	Maximum power capacity
R	Autocorrelation function
CA	California
CC	Cross correlation
DBA	Dynamic Time Warping Barycenter Averaging
DTW	Dynamic Time Warping
ED	Euclidean distance
GER	Germany
K	Number of clusters
LP	Linear Program
MILP	Mixed Integer Linear Program
MINLP	Mixed Integer Nonlinear Program
S	Battery storage level
SBD	Shape based distance
SSD	Sum of squared distances

References

[1] R. Green, I. Staffell, N. Vasilakos, Divide and Conquer? k-means clustering of demand data allows rapid and accurate simulations of the British electricity system, IEEE Transactions on Engineering Management 61 (2) (2014) 251–260. doi:10.1109/TEM.2013.2284386.

[2] P. Nahmmacher, E. Schmid, L. Hirth, B. Knopf, Carpe diem: A novel approach to select representative days for long-term power system modeling, Energy 112 (2016) 430–442. doi:10.1016/j.energy.2016.06.081.

[3] J. H. Merrick, On representation of temporal variability in electricity capacity planning models, Energy Economics 59 (2016) 261–274. doi:10.1016/j.eneco.2016.08.001.

[4] C. F. Heuberger, I. Staffell, N. Shah, N. M. Dowell, A systems approach to quantifying the value of power generation and energy storage technologies in future electricity networks, Computers and Chemical Engineering 107 (2017) 247–256. doi:10.1016/j.compchemeng.2017.05.012.

[5] C. L. Lara, D. S. Mallapragada, D. J. Papageorgiou, A. Venkatesh, I. E. Grossmann, Deterministic electric power infrastructure planning: Mixed-integer programming model and nested decomposition algorithm, Eurpean Journal of Operational Research 271 (2018) 1037–1054. doi:10.1016/j.ejor.2018.05.039.

[6] S. 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 (2017) 1–13. doi:10.1016/j.apenergy.2017.03.051. URL http://www.sciencedirect.com/science/article/pii/S0306261917302775

[7] A. Almaimouni, A. Ademola-Idowu, J. N. Kutz, A. Negash, D. Kirschen, Selecting and Evaluating Representative Days for Generation Expansion Planning, in: XX Power Systems Computation Conference, Dublin, Ireland, 2018, pp. 1–7.

[8] D. A. Tejada-Arango, M. Domeshek, S. Wogrin, E. Centeno, Enhanced Representative Days and System States Modeling for Energy Storage Investment Analysis, IEEE Transactions on Power Systems 8950 (c) (2018) 1–10. doi:10.1109/TPWRS.2018.2819578.

[9] B. Bahl, A. Kümpel, H. Seele, M. Lampe, A. Bardow, Time-series aggregation for synthesis problems by bounding error in the objective function, Energy 135 (2017) 900–912. doi:10.1016/j.energy.2017.06.082.

[10] B. Bahl, T. Söhler, M. Hennen, A. Bardow, Typical Periods for Two-Stage Synthesis by Time-Series Aggregation with Bounded Error in Objective Function, Frontiers in Energy Research 5 (January) (2018) 1–13. doi:10.3389/fenrg.2017.00035.

[11] B. Bahl, J. Lützow, D. Shu, D. E. Hollermann, M. Lampe, M. Hennen, A. Bardow, Rigorous synthesis of energy systems by decomposition via time-series aggregation, Computers and Chemical Engineering 112 (2018) 70–81. doi:10.1016/j.compchemeng.2018.01.023.
URL https://doi.org/10.1016/j.compchemeng.2018.01.023

[12] L. Kotzur, P. Markewitz, M. Robinius, D. Stolten, Impact of different time series aggregation methods on optimal energy system design, Renewable Energy 117 (2018) 474–487. doi:10.1016/j.renene.2017.10.017.

[13] P. Gabrielli, M. Gazzani, E. Martelli, M. Mazzotti, Optimal design of multi-energy systems with seasonal storage, Applied Energy (May) (2017) 1–17. doi:10.1016/j.apenergy.2017.07.142.

[14] T. Schuetz, M. H. Schraven, H. Harb, M. Fuchs, D. Mueller, Clustering algorithms for the selection of typical demand days for the optimal design of building energy systems, in: Proceedings of ECOS 2016 - The 29th international confernce on efficiency, cost, optimization, simulation and environmental impact of energy systems, Portoroz, Slovenia, 2016, pp. 1–12.

[15] F. Dominguez-Munoz, J. M. Cejudo-Lopez, A. Carrillo-Andres, M. Gallardo-Salazar, Selection of typical demand days for CHP optimization, Energ. Build. 43 (2011) 3036–3043. doi:10.1016/j.enbuild.2011.07.024.

[16] P. G. Brodrick, C. A. Kang, A. R. Brandt, L. J. Durlofsky, Optimization of carbon-capture-enabled coal-gas-solar power generation, Energy 79 (2015) 149–162. doi:https://doi.org/10.1016/j.energy.2014.11.003.

[17] P. G. Brodrick, A. R. Brandt, L. J. Durlofsky, Operational optimization of an integrated solar combined cycle under practical time-dependent constraints, Energy 141 (2017) 1569–1584. doi:10.1016/j.energy.2017.11.059.

[18] H. Teichgraeber, P. G. Brodrick, A. R. Brandt, Optimal design and operations of a flexible oxyfuel natural gas plant, Energy 141 (2017) 506–518. doi:10.1016/j.energy.2017.09.087.

[19] P. G. Brodrick, A. R. Brandt, L. J. Durlofsky, Optimal design and operation of integrated solar combined cycles under emissions intensity constraints, Applied Energy 226 (2018) 979–990. doi:10.1016/j.apenergy.2018.06.052.

[20] L. Kotzur, P. Markewitz, M. Robinius, D. Stolten, Time series aggregation for energy system design: Modeling seasonal storage, Applied Energy 213 (2018) 123–135. doi:10.1016/j.apenergy.2018.01.023.

[21] S. Aghabozorgi, A. Seyed Shirkhorshidi, T. Ying Wah, Time-series clustering - A decade review, Information Systems 53 (2015) 16–38. doi:10.1016/j.is.2015.04.007.

[22] F. Petitjean, A. Ketterlin, P. Gancarski, A global averaging method for dynamic time warping, with applications to clustering, Pattern Recognition 44 (3) (2011) 678–693. doi:10.1016/j.patcog.2010.09.013.

[23] J. Paparrizos, L. Gravano, k-Shape: Efficient and Accurate Clustering of Time Series, Acm Sigmod (2015) 1855–1870doi:10.1145/2723372.2737793.

[24] E. Keogh, H. A. Dau, N. Begum, Semi-Supervision Dramatically Improves Time Series Clustering under Dynamic Time Warping, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, ACM, Indianapolis, Indiana, 2016, pp. 999–1008. doi:10.1145/2983323.2983855.

[25] T. Teeraratkul, D. O'Neill, S. Lall, Shape-Based Approach to Household Electric Load Curve Clustering and Prediction, IEEE Transactions on Smart Grid (2017) 1–13doi:10.1109/TSG.2017.2683461.

[26] I. Blanco, J. M. Morales, An Efficient Robust Solution to the Two-Stage Stochastic Unit Commitment Problem 32 (6) (2016) 4477–4488. doi:10.1109/TPWRS.2017.2683263.

[27] S. Fazlollahi, S. L. Bungener, P. Mandel, G. Becker, F. Marchal, Multi-Objectives, Multi-Period Optimization of district energy systems: I-Selection of typical operating periods, Comput. Chem. Eng. (65) (2014) 54–66. doi:http://dx.doi.org/10.1016/j.compchemeng.2014.03.005.

[28] Todd D. Little, The Oxford Handbook of Quantitative Methods, Volume 2, Oxford Library of Psychology 2: Statist (2013) 551–. doi:10.1017/CBO9781107415324.004.

[29] E. Keogh, J. Lin, Clustering of time-series subsequences is meaningless: implications for previous and future research, Knowledge and Information Systems 8 (2) (2005) 154–177. doi:10.1007/s10115-004-0172-7.

[30] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing 26 (1978) 43–49. doi:10.1109/TASSP.1978.1163055.

[31] S. P. Lloyd, Least Squares Quantization in PCM, IEEE Transactions on Information Theory 28 (2) (1982) 129–137. doi:10.1109/TIT.1982.1056489.

[32] H. Steinhaus, Sur la division des corp materiels en parties, Bull. Acad. Polon. Sci 1 (804) (1956) 801.

[33] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1, Oakland, CA, USA., 1967, pp. 281–297.

[34] D. Arthur, S. Vassilvitskii, K-Means++: the Advantages of Careful Seeding, Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms 8 (2007) 1027–1025. doi:10.1145/1283383.1283494.

[35] H. D. Vinod, Integer Programming and the Theory of Grouping, Journal of the American Statistical Association 64 (326) (1969) 506–519. doi:10.1080/01621459.1969.10500990.

[36] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, 2009. doi:10.1007/978-0-387-84858-7.

[37] J. H. Ward, Hierarchical grouping to optimize an objective function, Journal of the American statistical association 58 (301) (1963) 236–244.

[38] J. C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, Journal of Cybernetics 3 (3) (1973) 32–57. doi:10.1080/01969727308546046.

Supplementary Information
Clustering methods to find representative periods for the optimization of energy systems: an initial framework and comparison

Holger Teichgraeber[a,*], Adam R Brandt[a]

[a]*Department of Energy Resources Engineering, Stanford University, Green Earth Sciences Building 065, 367 Panama St., Stanford, California, USA*

1. Introduction

This document provides additional details concerning the paper "Clustering methods to find representative periods for the optimization of energy systems: an initial framework and comparison".

2. Theorems and Proofs

The following proofs are shown for a period length of 1 for readability purposes, but can be easily generalized to any period length (e.g. 24 as in this paper). Table 1 shows an overview of the results of Theorem 1 and Theorem 2. We restate the theorems prior to the proofs for readability.

Theorem 1. *For an LP, if data is clustered in the objective function coefficient vector* $\boldsymbol{c}$*, the constraints are structurally the same for all periods, and the cluster representation is centroid based, then the optimization problem with full input data is a relaxation of the optimization problem with clustered input data.*

Proof of Theorem 1. Let the optimization problem with full input data be

$$\max\{\boldsymbol{c}^T\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \wedge \boldsymbol{x} \geq 0\} \tag{1}$$

We cluster coefficients $\boldsymbol{c} = [c_1 \ ... \ c_N]$ into K disjoint clusters $\mathcal{C}_k$ with cluster representations $\boldsymbol{r} = [r_1 \ ... \ r_k]$, $\boldsymbol{r}_k = \frac{1}{N_k}\sum_{i\in\mathcal{C}_k} c_i$, where $N_k = |\mathcal{C}_k|$. The clustered optimization problem is

$$\max\{\bar{\boldsymbol{c}}^T\bar{\boldsymbol{x}} \mid \bar{\boldsymbol{A}}\bar{\boldsymbol{x}} \leq \bar{\boldsymbol{b}} \wedge \boldsymbol{x} \geq 0\} \tag{2}$$

where $\bar{\boldsymbol{c}}^T = [N_1 r_1 \ ... \ N_K r_K]$, $\bar{\boldsymbol{A}}$ is a submatrix of $\boldsymbol{A}$, and $\bar{\boldsymbol{b}}$ is a subvector of $\boldsymbol{b}$. We can then reformulate the clustered optimization problem

$$\begin{aligned}
(2) &= \max\{\sum_{k\in\mathcal{K}} N_k r_k \bar{x}_k \mid \bar{\boldsymbol{A}}\bar{\boldsymbol{x}} \leq \bar{\boldsymbol{b}} \wedge \bar{\boldsymbol{x}} \geq 0\} \\
&= \max\{\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{C}_k} c_i \bar{x}_k \mid \bar{\boldsymbol{A}}\bar{\boldsymbol{x}} \leq \bar{\boldsymbol{b}} \wedge \bar{\boldsymbol{x}} \geq 0\} \\
&= \max\{\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{C}_k} c_i x_i \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \wedge \boldsymbol{x} \geq 0 \ \wedge \ x_j = x_l \quad \forall\, j,l \in \mathcal{C}_k,\ k \in \mathcal{K}\} \\
&= \max\{\boldsymbol{c}^T\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \wedge \boldsymbol{x} \geq 0 \ \wedge \ x_j = x_l \quad \forall\, j,l \in \mathcal{C}_k,\ k \in \mathcal{K}\}
\end{aligned} \tag{3}$$

which is the original optimization problem (Equation 1) with additional constraints. This proves Theorem 1.

*Corresponding author. Tel: +1-650-725-0851
Email addresses: hteich@stanford.edu (Holger Teichgraeber), abrandt@stanford.edu (Adam R Brandt)

Theorem 2. *For an LP, if data is clustered in the constraint coefficient vector* $\boldsymbol{b}$*, the corresponding constraints are structurally the same, and the cluster representation is centroid based, then the optimization problem with clustered input data is a relaxation of the optimization problem with full input data.*

Proof of Theorem 2. Let the optimization problem with full input data be

$$z_0 = \max\{\boldsymbol{c}^T\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \wedge \boldsymbol{x} \geq 0\} \tag{4}$$

and then the associated dual formulation is

$$\hat{z}_0 = \min\{\boldsymbol{b}^T\boldsymbol{y} \mid -\boldsymbol{A}^T\boldsymbol{y} \leq -\boldsymbol{c} \wedge \boldsymbol{y} \geq 0\} \tag{5}$$

where $z_0 = \hat{z}_0$ due to strong duality. We cluster coefficients $\boldsymbol{b} = [b_1 \ ... \ b_N]$ into K disjoint clusters and obtain $\bar{\boldsymbol{b}}$. The optimization problem with clustered input data is

$$z_c = \max\{\bar{\boldsymbol{c}}^T\bar{\boldsymbol{x}} \mid \bar{\boldsymbol{A}}\bar{\boldsymbol{x}} \leq \bar{\boldsymbol{b}} \wedge \bar{\boldsymbol{x}} \geq 0\} \tag{6}$$

The associated dual is

$$\hat{z}_c = \min\{\bar{\boldsymbol{b}}^T\bar{\boldsymbol{y}} \mid -\bar{\boldsymbol{A}}^T\bar{\boldsymbol{y}} \leq -\bar{\boldsymbol{c}} \wedge \bar{\boldsymbol{y}} \geq 0\} \tag{7}$$

By Theorem 1, we obtain

$$\hat{z}_c = \min\{\boldsymbol{b}^T\boldsymbol{y} \mid -\boldsymbol{A}^T\boldsymbol{y} \leq -\boldsymbol{c} \wedge \boldsymbol{y} \geq 0 \wedge y_j = y_l \quad \forall\ j,l \in \mathcal{C}_k,\ k \in \mathcal{K}\} \tag{8}$$

The primal of this reformulation is

$$z_{cr} = \max\left\{ \begin{bmatrix}\boldsymbol{c}^T & \boldsymbol{0}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{s}\end{bmatrix} \,\middle|\, \begin{bmatrix}\boldsymbol{P}\boldsymbol{A} & M\end{bmatrix}\begin{bmatrix}\boldsymbol{P}\boldsymbol{x}\\ \boldsymbol{s}\end{bmatrix} \wedge\ \boldsymbol{x} \geq 0 \right\} \tag{9}$$

where $\boldsymbol{P}$ is a permutation matrix which sorts the rows of the constraint matrix by cluster assignment, $\boldsymbol{s}$ are additional variables, and $\boldsymbol{M}$ is a block diagonal matrix with entries $\boldsymbol{M}_i$,

$$\boldsymbol{M}_i = \begin{bmatrix} -1 & \cdots & -1 \\ & \boldsymbol{I} & \end{bmatrix}_{N_k \times N_k - 1} \tag{10}$$

By strong duality, $z_{cr} = \hat{z}_c = z_c$. Problem formulation 9 is equivalent to problem formulation 6 with additional variables that relax the constraints. Thus, the problem with clustered input data is a relaxation of the problem with full input data, which proves Theorem 2.

Theorem 3. *For an LP, if data is clustered in the objective function coefficient vector* $\boldsymbol{c}$ *or in the constraint coefficient vector* $\boldsymbol{b}$ *with structurally similar constraints for all periods, and if the cluster representation is centroid based and the clustering algorithm is hierarchical clustering using Ward's algorithm, then the objective function value* z_k *is a monotonic function of the number of clusters* k*.*

Proof of Theorem 3. Let $\mathcal{C}_1 ... \mathcal{C}_k$ be the clusters obtained from hierarchical clustering at any number of clusters k. Hierarchical clustering then determines the cluster representations for k-1 clusters by merging two of the k clusters. Let z_k be the objective function of the optimization problem with these k clusters as input data. Let z_{k-1} be the objective function value of the optimization problem with the k-1 clusters as input data.

In the case of clustering in the objective function coefficient vector $\boldsymbol{c}$, the optimization problem with the k clusters as input data can be expressed as Equation 1, and the optimization problem with the k-1 clusters as input data can be expressed as Equation 2. By Theorem 1, the optimization problem with k clusters is a relaxation of the optimization problem with k-1 clusters. Thus, if the optimization problem is a maximization problem, then $z_{k-1} \leq z_k\ \forall\ k = \{2...N\}$, and if the optimization problem is a minimization problem, then $z_{k-1} \geq z_k\ \forall\ k = \{2...N\}$. This proofs Theorem 3 for clustering in $\boldsymbol{c}$.

The proof for optimization problems where data is clustered in the constraint coefficient vector $\boldsymbol{b}$ can be carried out similarly using Theorem 2.

Table 1: Relationship of objective values of minimization and maximization problems when clustering in either c or b occurs. z_o is the optimal objective value of the problem with full input data, and z_c is the optimal objective value of the problem with clustered input data.

Clustering of \ Problem type	min	max
$\boldsymbol{c}$	$z_c \geq z_o$	$z_c \leq z_o$
$\boldsymbol{b}$	$z_c \leq z_o$	$z_c \geq z_o$

3. Normalization Methods - Implementation Details

Z-normalization for the three normalization scopes can be implemented with the following equations:
Full normalization yields

$$\hat{\boldsymbol{P}} = \frac{1}{\sigma}(\boldsymbol{P} - \mu) \tag{11}$$

where μ is the overall mean and σ is the overall standard deviation. After applying clustering, we obtain K normalized cluster representations $\hat{\boldsymbol{r}}_k \in \hat{\boldsymbol{R}}$, $\hat{\boldsymbol{R}} \in \mathbb{R}^{K\times T}$. We obtain the denormalized cluster representations $\boldsymbol{r}_k \in \boldsymbol{R}$, $\boldsymbol{R} \in \mathbb{R}^{K\times T}$ by

$$\boldsymbol{R} = \sigma\hat{\boldsymbol{R}} + \mu \tag{12}$$

Note that in this work, we only consider one type of input data. In case of several types of input data are used (e.g. electricity prices and solar capacity factors), full normalization is applied to each type individually, and the data are merged afterwards.

Element-based normalization yields

$$\hat{\boldsymbol{P}} = \operatorname{diag}^{-1}(\boldsymbol{\sigma}_{el})(\boldsymbol{P} - \boldsymbol{\mu}_{el}\mathbf{1}^T) \tag{13}$$

where $\boldsymbol{\mu}_{elem} \in \mathbb{R}^T$ is the vector of hourly means, $\boldsymbol{\sigma}_{elem} \in \mathbb{R}^T$ is the vector of hourly standard deviations of the full year of price data, and $\mathbf{1}$ is a vector of all ones. The denormalized cluster representations can be obtained from

$$\boldsymbol{R} = \operatorname{diag}(\boldsymbol{\sigma}_{elem})\ \hat{\boldsymbol{R}} + \boldsymbol{\mu}_{elem}\mathbf{1}^T \tag{14}$$

Sequence-based normalization normalizes each daily sample to zero mean and standard deviation one.

$$\hat{\boldsymbol{P}} = (\boldsymbol{P} - \mathbf{1}\boldsymbol{\mu}_{seq}^T)\operatorname{diag}^{-1}(\boldsymbol{\sigma}_{seq}) \tag{15}$$

where $\boldsymbol{\mu}_{seq} \in \mathbb{R}^N$ is the vector of daily means and $\boldsymbol{\sigma}_{seq} \in \mathbb{R}^N$ is the vector of daily standard deviations.

4. Clustering methods - additional details

4.1. k-medoids

The Binary Integer Program (BIP) as introduced by [1] and used by [2] is formulated as follows:

$$\begin{aligned}
&\min \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}} D_{ij}z_{ij} \\
&\text{s.t.} \\
&\sum_{i\in\mathcal{N}} z_{ij} = 1 && \forall\, j \in \mathcal{N} \\
&z_{ij} \leq y_i && \forall\, i \in \mathcal{N} \\
&\sum_{i\in\mathcal{N}} y_i = K \\
&y_i, z_{ij} \in \{0,1\}
\end{aligned} \tag{16}$$

where $D \in \mathbb{R}^{n\times n}$ is the precomputed distance matrix between all observations with $D_{ij} = \mathrm{ED}(\boldsymbol{p}_i, \boldsymbol{p}_j)^2$, y_i signifies whether $\boldsymbol{p}_i$ is a medoid of $\mathcal{C}_i$, and z_{ij} signifies whether $\boldsymbol{p}_j, j \in \mathcal{C}_i$.

4.2. DBA clustering

For the "warping window" in the DTW distance, we use the Sakoe-Chiba bandwidth [3].

5. Additional Results

5.1. Comparing different normalization scopes

While Figure 6 in the main text shows the centroid-based and medoid-based clustering methods with full normalization scope, and the shape-based methods with the sequence-based normalization scope, Figure 1 shows all clustering methods with the full normalization scope, Figure 2 shows all clustering methods with the element-based normalization scope, and Figure 3 shows all clustering methods with sequence-based normalization scope.

We observe that the methods with centroid and medoid as their representation perform very similarly to the full normalization scope and the element-based normalization scope. Using the sequence-based normalization scope, these methods perform well on the battery problem because this normalization scope helps to extract shape. The methods using the medoid as their representation perform very close to the objective function of the reference problem but increase and decrease unpredictably with an increasing number of clusters. On the gas turbine problem for the Germany data, sequence-based normalization yields bad performance. This shows that similarly to the shape-based clustering methods, sequence-based normalization alone is highly problem specific. All methods perform well on the gas turbine problem with California input data, similarly to the previously described methods.

K-shape clustering normalizes within the algorithm and is thus not applicable to full normalization scope and element-based normalization scope. However, DBA clustering can be used with the full normalization scope and improves on the battery problem as the Sakoe-Chiba bandwidth increases.

Using the full normalization scope, we compare DBA clustering with a Sakoe-Chiba warping window of 0 to 2. A warping window of 1 increases the objective function for both optimization problems. As the warping window increases further, the objective function increases on average, but it also increases toward and decreases away from the full representation, with less of a pattern emerging. As we increase the warping window size, the clustering algorithm is able to compare hours further and further away, which at some higher warping window size may lead to clustering peaks and valleys in price that may not belong together physically. Increasing the size of the warping window to a number too large can more generally be seen as overfitting.

We can observe improved performance for DBA clustering even with the element-based normalization scope; however, there seems no reasonable explanation for this behavior.

Note that k-means clustering and DBA clustering with a Sakoe-Chiba bandwidth of 0 are equivalent, which we observe in each of the three figures, and which confirms the implementation of our algorithm.

References

[1] H. D. Vinod, Integer Programming and the Theory of Grouping, Journal of the American Statistical Association 64 (326) (1969) 506–519. doi:10.1080/01621459.1969.10500990.
URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1969.10500990

[2] F. Dominguez-Munoz, J. M. Cejudo-Lopez, A. Carrillo-Andres, M. Gallardo-Salazar, Selection of typical demand days for {CHP} optimization, Energ. Build. 43 (11) (2011) 3036–3043. doi:10.1016/j.enbuild.2011.07.024.

[3] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing 26 (1) (1978) 43–49. doi:10.1109/TASSP.1978.1163055.

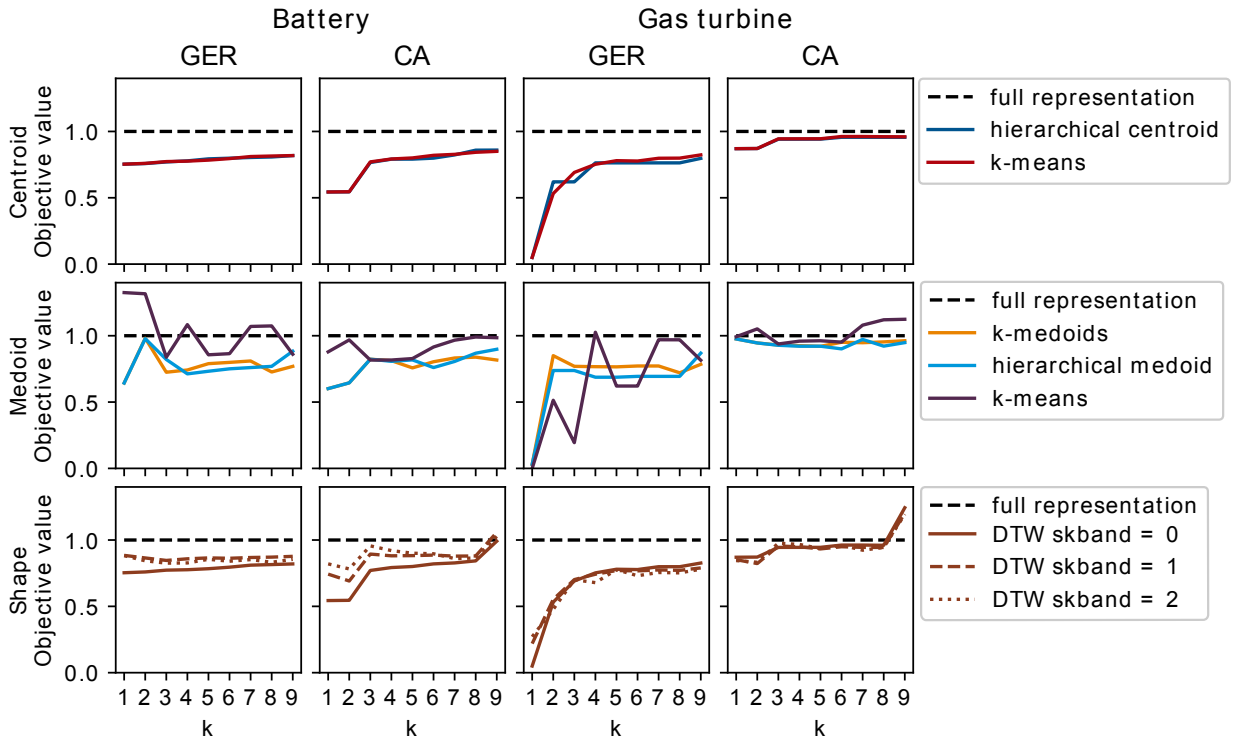


Figure 1: Objective value (revenue) as a function of the number of clusters for the different clustering methods. Normalization scope: full normalization.

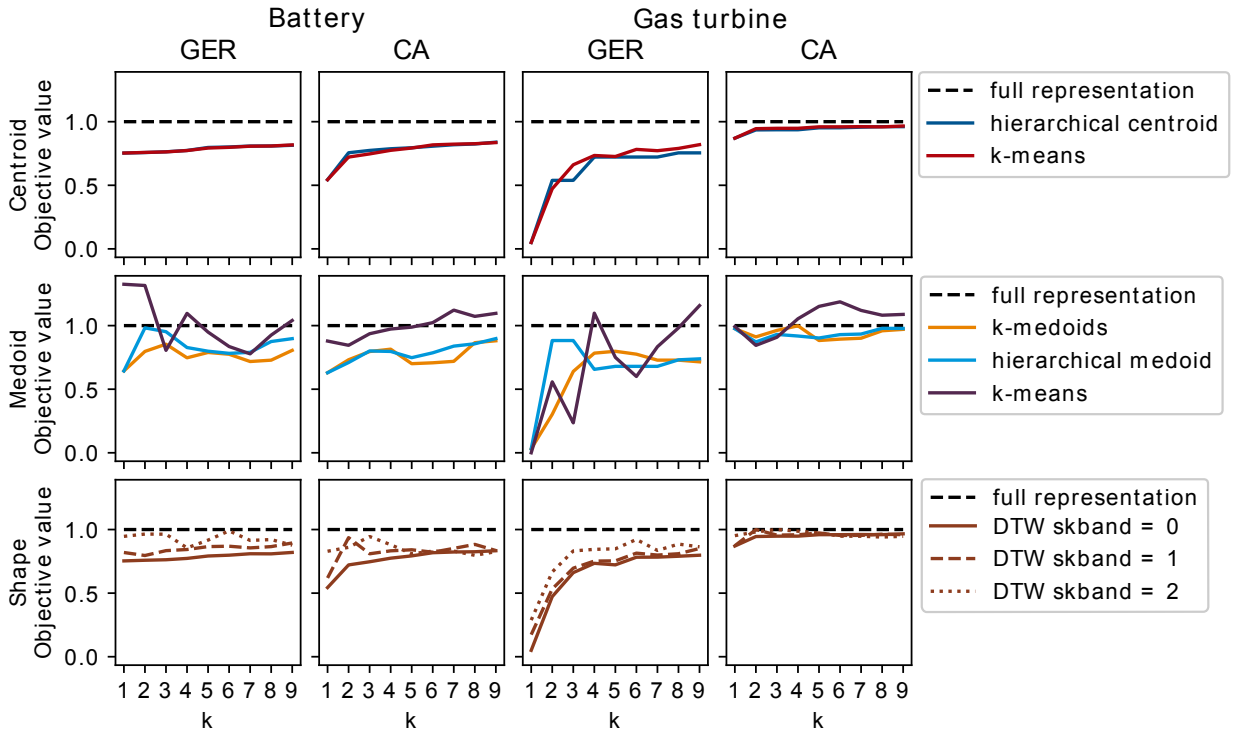


Figure 2: Objective value (revenue) as a function of the number of clusters for the different clustering methods. Normalization scope: element-based normalization.

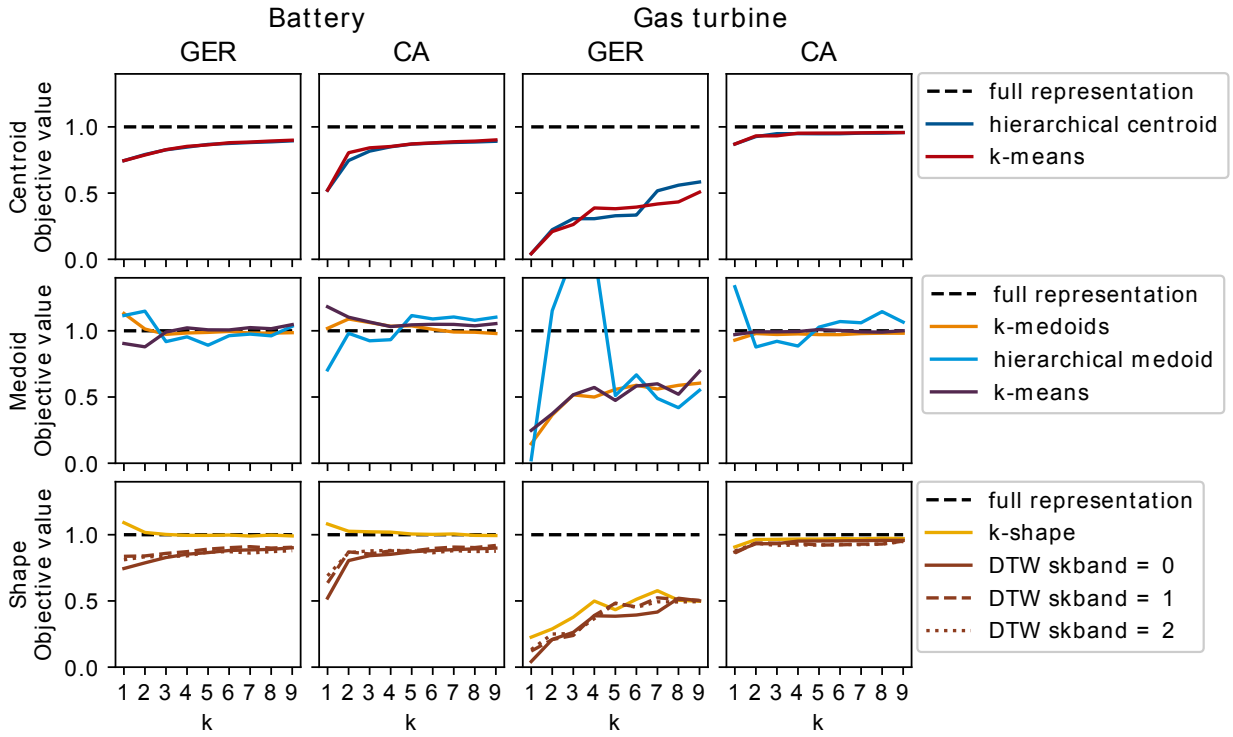


Figure 3: Objective value (revenue) as a function of the number of clusters for the different clustering methods. Normalization scope: sequence-based normalization.