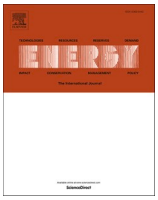
Energy 323 (2025) 135788

Contents lists available at ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy



Optimizing capacity expansion modeling with a novel hierarchical clustering and systematic elbow method: A case study on power and storage units in Spain

Milad Riyahi *, Alvaro Gutiérrez Martín

Higher Technical School of Engineers in Telecommunication, Universidad Politécnica de Madrid, Spain

ARTICLE INFO

Handling editor: Neven Duic

Keywords:
Capacity expansion model
Hierarchical clustering
Euclidean distance
Elbow method
Stopping criterion
K-medoids
K-means

ABSTRACT

To reduce the computational complexity of Capacity Expansion Models, the planning horizon must be simplified into representative time-periods. Also, to accurately model the expansion of power and storage units, these representative time periods must reveal the mid-term dynamics of the planning horizon. In this paper, a novel hierarchical clustering algorithm is presented that retains the chronology of the original data in creating representative time periods. The proposed algorithm, first, determines the optimal number of clusters with a modified elbow method, enhanced with a stopping criterion to prevent it from running uselessly. The designed stopping criterion works based on percentage variance and runtime to determine the number of clusters systematically. Then, the proposed clustering algorithm employs a novel selection strategy based on the Euclidean distance, k-Medoid, and k-Means to determine the most proper representative vector in each cluster. In this way, it reduces the computational time of capacity expansion models while maintaining the accuracy of final answers. To evaluate its performance, the proposed algorithm is tested on energy data, including demand, photovoltaic, wind, and hydrogen generation, across hourly, daily, and weekly time periods. Also, the performance of the proposed clustering algorithm in selecting the number of clusters and clustering is compared with the results of some well-known methods on accuracy and runtime metrics. Numerical results show that the proposed clustering method selects a more appropriate number of clusters in less computational time than other systematic approaches. Moreover, findings on clustering show that the proposed algorithm achieves the highest accuracy on weekly and daily time periods compared to well-known clustering methods, with the error rate of 118 % and 52 %, respectively. Furthermore, implementation results show that the proposed clustering reduces the computational time of capacity expansion models by 84.81 % and 55.91 % on weekly and daily time periods. Additionally, this study assesses the robustness of the clustering methods through a sensitivity analysis, which shows that the proposed algorithm outperforms the others in this metric, as well.

Nomenclature

A. Indexes and Sets	
g	Generation technology index
G^r	Renewable generation technologies
s	Storage technology index
S	Storage technologies
t	Time period index
B. Parameters	
η_s	Energy capacity of storage technology s
ξ_s	Round-trip efficiency of storage technology s
ρ_{gt}	Capacity factor of technology g at time t

(*continued on next column*)

(*continued*)

τ_t	Duration of time-period t
a_n	Yearly availability of hydrogen energy
$\overline{b}_s^0$	Initial capacity of storage s (MW)
$\overline{d}_t$	Demand level at time-period t (MW)
$\overline{h}^0$	Initial hydrogen capacity (MW)
$\widehat{h}$	Maximum hydrogen power capacity (MW)
i_g^G	Investment cost of technology g (€ /MW)
i^H	Investment cost of hydrogen power capacity (€ /MW)
i_s^S	Investment cost of storage s (€ /MW)
$\overline{p}_g^0$	Initial capacity of technology g (MW)

(*continued on next page*)

* Corresponding author.
E-mail address: milad.riyahi@alumnos.upm.es (M. Riyahi).

https://doi.org/10.1016/j.energy.2025.135788
Received 30 August 2024; Received in revised form 16 March 2025; Accepted 22 March 2025
Available online 24 March 2025

(*continued*)

$\widehat{p}_g$	Maximum capacity of technology g (MW)
sc	Load shedding cost (€ /MWh)
ω_t	Weight of time-period t
y_g^G	Expected lifetime of technology g ($years$)
y^H	Expected lifetime of hydrogen power units ($years$)
y_s^S	Expected lifetime of storage s ($years$)
C. Variables	
b_{st}^+	Charge of storage s at time-period t (MW)
b_{st}^-	Discharge of storage s at time-period t (MW)
b_{st}	Energy level of storage s at time-period t (MWh)
$\overline{b}_s$	Additional capacity of storage s at time-period t (MW)
d_t	Satisfied demand at time-period t (MW)
h_t	Hydrogen power generation at time-period t (MW)
$\overline{h}$	Additional Hydrogen capacity (MW)
p_{gt}	Generation of technology g at time-period t (MW)
$\overline{p}_g$	Additional capacity of technology g at time-period t (MW)

1. Introduction

Based on European Energy Roadmap [1], current energy systems must be substituted with carbon free energy sources. To this end, clear policies along with a precise investment overview are needed. Capacity Expansion Models (CEMs) are practical tools to determine the properties of future energy plants, satisfying the demand of the future with optimized cost [2]. The properties of energy plants make CEMs long-term problems that extend to many years [3]. Since the elements of an energy plant are normally represented on an hourly basis, computing CEMs would be intractable for long-term extensions. Subsequently, a stylized planning horizon must be employed to diminish their computational complexity [4]. Selecting some limited blocks of the full-time horizon would be an immediate answer to address this problem [5,6]. However, despite the computational efficiency of this strategy, it does not consider the chronology of the dataset. Therefore, it fails to fulfil the goal of CEMs. Selecting Representative Time Periods (RTPs) is an alternative approach where a set of limited but representative data is used to model the full-time horizon [3]. With this strategy, a full-time horizon dataset (e.g. one year of demand) is processed, and the most appropriate RTPs are selected to diminish the computational complexity of the CEM. Although RTPs can be defined in different formats such as hours, days, and weeks, selecting a reduced set of days or weeks is a more reliable strategy because the chronology of the parameters is typically preserved within these periods [7,8]. In the case of days and weeks, the original dataset is grouped in 24-h and 168-h periods, respectively and then proper RTPs are selected.

Fig. 1 exemplifies the electricity demand for two weeks and two representative days. As illustrated in the figure, the first day of the demand is represented by one representative day (green line), while the demand for the following thirteen days is represented by the other representative day (blue line). Therefore, instead of solving the CEM on the full two weeks (336 h) only two representative days (48 h) are used.

One solution for the RTPs' finding is making use of clustering algorithms [9]. Clustering algorithms split the input data into groups called clusters in a way that the elements of a cluster are more similar to each other than the elements of the other clusters [10]. After clustering the data, one element of each cluster is selected to represent the corresponding cluster [11]. Therefore, time periods (e.g. hour, day, or week) are gathered into some clusters and an element of each cluster is selected as RTP, condensing the full-time horizon into a reduced number of representative data. However, there are some major challenges that must be addressed. First, the user must decide about the clustering algorithm. The literature shows a strong variety of clustering options that are used, where K-Means, k-Medoids, and hierarchical clustering are the most common clustering methods to select RTPs [9]. However, there is no methodology to select the best option [12]. While k-Means is an accurate and easy-to-implement algorithm, it is limited by its need for predefined number of clusters, and it misses peak values in the data. K-Medoids, on the other hand, is more robust to peak values but does not fully capture the influence of all the elements within a cluster and limited to a predefined number of clusters. In contrast, hierarchical clustering is not limited to a predefined number of clusters; However, it potentially overlooks peak values in the data because it always considers the average of the elements within the cluster to represent it.

Determining the optimum number of clusters is the next challenge that must be addressed. The main problem is that there is no precise method to determine this optimum number beforehand [9]. Increasing the number of clusters gradually to measure the optimization error is one common strategy that is employed in Refs. [13,14]. Elbow methods and average normalized root mean square error are other approaches used by Refs. [15–18], respectively. Another well-known method is silhouette coefficient, which determines the number of clusters based on the Euclidean distance between samples within each cluster [19]. However, running the algorithm for all the possible numbers of clusters is the common drawback of the mentioned methods. To tackle this problem, an algorithm is designed in Ref. [20] to systematically select the optimum number of clusters. This method gradually splits and merges the clusters according to the inner variance and similarity between the centroids of the clusters. The process is continued until reaching the optimum number of clusters that is determined by three stopping criteria to prevent the algorithm from running unnecessarily. This method suffers from the same problem as k-Means and hierarchical clustering, where it relies just on the centroids to merge two similar

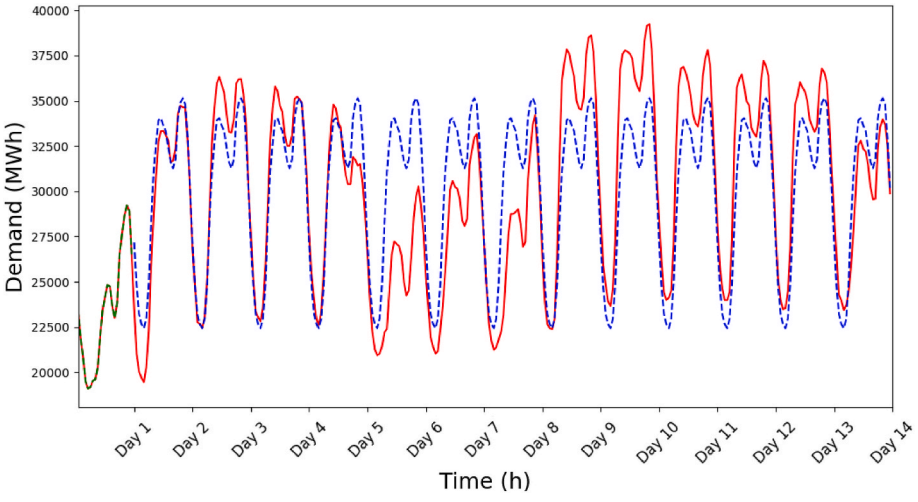


Fig. 1. Spain electricity demand (red line) and two representative days (first day green and second day blue).

clusters. Consequently, it may merge clusters that are not very similar, leading to an incorrect determination of the number of clusters. In addition, there are some other methods such as [3], where the authors select a number as the optimum cluster with no explanation.

Finally, the duration of each PTR is a challenge, where an hourly, daily, and weekly basis are typically selected. The literature shows that selecting representative days is the most common methodology [9]. However, in some studies, such as [3], the proposed hourly clustering is compared with daily and weekly versions in terms of accuracy and computational time and the results show that it outperforms them.

According to those challenges, this paper proposes a novel agglomerative hierarchical clustering approach to retain the chronological information of the full-time horizon. This paper selects hierarchical clustering over the other methods because it does not depend on initialization [3]. Moreover, to address the issue of the number of clusters, this study proposes an improved elbow methos to determine it systematically. Unlike the previous methods that must run for all the possible number of clusters, this study presents Elbow Method with Stopping Criterion (EMSC) to prevent the algorithm from running when it is not needed. Moreover, the proposed hierarchical clustering method uses a novel strategy to select the most appropriate vector to represent each cluster. In contrast to preceding hierarchical approaches such as [3,7], the proposed clustering strategy pays attention to both medoid and mean vectors of a cluster. It measures the inner Euclidean distance between the mean and medoid vectors of the cluster with other members within the cluster, selecting the one with the minimum distance as the representative vector. In this way, the proposed method captures the chronological information of the input data as the selected RTP is more similar to other members of the cluster. Finally, the paper runs the proposed Euclidean based Agglomerative Hierarchical Clustering Method (EAHCM) on an hourly, daily, and weekly basis to analyze which of the duration options works better in terms of computational time and error rate. To extract the level of accuracy, the results of solving CEM based on representative hours, days, and weeks are compared with full-time horizon data. Therefore, the contributions of this paper are as follows.

- Proposing a stopping criterion to enhance elbow method for determining the optimal number of clusters in energy consumption profiles.
- Developing a novel agglomerative hierarchical clustering approach to improve energy load pattern identification, using Euclidean distance for representative selection.
- Combining medoid and mean-based representative selection to preserve the chronological structure of energy patterns.
- Comparing hourly, daily, and weekly clustering to assess the impact of temporal resolution on energy system modeling in Spain.

The rest of this paper is organized as follows. The state of the art is reviewed in Section 2. Section 3 describes the process of EMSC to determine the optimum number of clusters. Section 4 proposes the EAHCM. The CEM problem is detailed in Section 5. Numerical results are detailed in Section 6, while Section 7 is devoted to its comparison. Finally, conclusions are presented in Section 8.

2. State of art

As aforementioned, k-Means, k-Medoids, and hierarchical clustering are the main RTPs selection techniques found in the literature. The traditional k-Means method [21] is purposed to partition the input data into a number of clusters while minimizing within-cluster variances. Regarding the performance of this method, Li et al. integrate it into a cost-based representative day selection algorithm [9]. Rather than solving CEM on the full time-horizon, this study solves CEM separately for each day to determine the optimal solutions. The days are then clustered based on their optimal CEM results, grouping those with similar answers. Afterwards, CEM is solved based on the clustered days. In this study, the authors use k-Means with three clusters to reduce the dataset, but they do not explain why this number is chosen. In addition, the computational time introduced by running the optimization for all possible days is a significant challenge for this study, especially for long-term CEMs. Mallapragada et al. investigate the impacts of adding temporal and operational details on the outcomes of CEM problem [13]. In this study, k-Means is employed to select suitable RTPs in the format of days to reduce the full time-horizon. The main reason for choosing k-Means over other methods is the structure of the data. As the authors explain, since their data does not have an islanded structure, k-Means outperforms other clustering approaches. However, the study has a key drawback: the number of clusters is not selected systematically. Li et al. propose a mixed-integer linear programming to solve transmission expansion planning [22]. Given the problem's complexity, some representative days are selected by k-Means. Moreover, to find the best number of clusters, this study gradually increases the number of representative days and selects the best number of clusters based on the change in trial-and-error of the optimization. Therefore, this method requires multiple runs of the optimization model, increasing computational time.

In addition to the particular weaknesses of clustering methods based on classic k-Means, these methods suffer from one common problem that is capturing peak values in the original data. This issue arises because k-Means always uses the clusters' mean, overlooking peak values. To overcome the limitations of k-Means, particularly its inability to consider peak values in the data, García-Cerezo et al. introduce a modified k-Means, providing the clustering in two stages [23]. In this research, initially, days are clustered into k_1 clusters by means of k-Means. Subsequently, k-Means is run on each cluster separately and divides them into k_2 clusters. Therefore, the input data is categorized into k clusters, where $k = k_1 \times k_2$. Although this two-stage clustering approach improves clustering quality, it does not fully resolve the issue, as the primary problem stems from selecting the mean of each cluster as the RTPs. Additionally, the study does not provide any explanation for choosing the number of clusters, which is a significant weakness. Munoz and Mills analyze how time resolution and solar PV penetration affect resource adequacy using a modified k-Means clustering approach [24]. In this study, k-Means is constrained to include 10 peak loads, improving the clustering of representative days. Also, to determine the best number of clusters, this study examines 5 to 2000 number of clusters, finding that at least 50 days are required for accurate optimization. While forcing k-Means to include certain number of peaks improves the clustering quality, running the clustering method based on different number of clusters increases the computational time, which is a weakness of the method.

The k-Medoid is another clustering approach that is used in different studies [9]. The classic k-Medoids method [25] operates like k-Means, with the key difference that it seeks to minimize the distances from the medoids of each cluster. Given that a medoid is an actual element of the corresponding cluster, k-Medoids can retain the chronological order of the original data, and it can improve the presentation of peak values. Regarding this characteristic, k-Medoids has been employed in different studies. Scott et al. use k-Medoids to introduce a novel RTP selection strategy [26]. In this strategy, there are two sets of medoids, namely preselected and random. The preselected medoids are the ones that are selected based on prior knowledge, ensuring the selection of peak values, while the rest of the medoids are selected randomly. This study reports all the results based on nine clusters, but it is not described why it is selected as the number of clusters. Bahl et al. use k-Medoids to optimize the synthesis of energy systems, aiming to determine the optimal investment and operation strategies for power systems [27]. First, this study investigates the best period length for RTPs, selecting days and then, k-Medoids is employed to select suitable RTPs that runs based on five different numbers of clusters, which are 1, 2, 4, 6, and 12. In this study, the selection of the RTPs depends on the number of clusters,

where a RTP in spring is selected in the case of 1 cluster and 1 RTP per month is selected in the case of 12 clusters. Unfortunately, this study does not explain why these numbers of clusters are selected, which is a weak point for it. Maiz et al. use k-Medoids to select representative days for solving the virtual power plant expansion problem [28]. Due to the importance of market prices and renewable energy generation, this study runs k-Medoids on these datasets to reduce them. Also, in this study, the number of clusters is arbitrarily set to four without explanation, which is a drawback for it.

Although k-Medoids outperforms k-Means in terms of retaining peak values, it does not consider the effects of all the elements within the cluster that is an inherent weakness for all the clustering method based on classic k-Medoids. To address the problems of k-Medoids, Arnold et al. introduce a modified version to solve CEM [29]. In this research, Pearson correlation coefficient and Hamming similarity are combined with k-Medoids to retain the effects of the elements in each cluster. Moreover, this research considers eight clusters and pays attention to weekday and weekend in running the clustering method. However, the number of clusters is not determined systematically, that is a weakness for this method. Anderson et al. combine k-Means with k-Medoids to reduce the dimensionality of the input data for solving CEM [30]. In this approach, the days are first clustered using k-Means, and then the medoid of each cluster is selected as the corresponding RTP. Although combining k-Means with k-Medoids helps mitigate the issue of peak values, it does not fully resolve it. This is because the medoid is chosen only after clustering, meaning k-Means has already ignored the peak values. Additionally, this study determines the number of clusters by running numerical optimization on different numbers of clusters, which is a drawback because running the optimization process multiple times increases the computational time. Li et al. propose a novel k-Medoids that clusters the input data in the format of weeks [31]. To measure the distance between the medoids accurately, this study employs dynamic time warping combined with Euclidean distance. In addition, it sets the number of clusters equal to five; However, it does not describe why this number of clusters is selected, which is a weakness for this study. Theodorakos et al. modify k-Medoids by using Bayesian optimization, which selects the data dimensionality reduction method, the distance metric for comparing different days, and the k-Medoid hyperparameters [32]. Additionally, the study runs the clustering method with different number of clusters to find the best number of clusters. However, running the clustering method for multiple number of clusters and then comparing the optimization results increases computational time, which is a drawback of this study.

Unlike k-Means and k-Medoids, where the merging process is predefined, hierarchical clustering is a method where the merging section can include additional conditions [3]. The classic Hierarchical Clustering [33] works based on dissimilarity of the elements within the clusters, where two members with the lowest dissimilarity are put in one cluster. Pineda and Morales propose a modified hierarchical clustering, named chronological clustering to solve CEMs [3]. In this study, the authors consider one more condition for grouping two members in one cluster. Not only the RTPs of the clusters must be similar, but also, they must be adjacent. After categorizing RTPs, the authors utilize the means of the clusters to construct the final representative vectors. Furthermore, this study tests the clustering method on hourly, daily, and weekly RTPs, considering 672 h, 28 days, and 4 weeks, finding that hourly clustering provides the highest accuracy. However, considering the means of the clusters is the main drawback as it loses the peak values. Also, this study does not explain how the number of clusters is selected, which is another limitation. To address this problem, García-Cerezo et al. introduce recursive hierarchical clustering [34]. This research involves two sets: the first set includes all the possible RTPs, while the second one consists of the selected RTPs. At each iteration, the algorithm measures the dissimilarity between all the elements in the first set with the elements in the second set. Subsequently, an RTP with the highest dissimilarity is transferred to the selected RTPs set. To make sure that the peak values are not overlooked, the RTPs containing the peak values are transferred as the first members of selected RTPs set. Computational time, particularly for long-term data, is the main drawback of this algorithm because many comparisons must be made at each iteration. Moreover, this study does not employ any strategy to determine the best number of clusters, which is another weakness. To detect the seasonality impacts of the original data, Domínguez and Vitali propose repetitive chronological clustering [1]. In this study, the process starts with the full dataset, where the first seasonal parameter is detected by running chronological clustering. Then, it runs the chronological clustering on the representative vectors, resulting from the first step, to detect the second seasonality parameter. In this study, the process continues to reach the desired seasonality parameters; However, this number is not determined systematically and that is a weakness for this study. Marcy et al. compare the performance of hierarchical clustering for selecting time periods with sequential and categorial time selections [35]. This research shows that hierarchical clustering reaches the lowest root-mean-square-error among the methods, which prove its quality.

Determining the best number of clusters is a major challenge for all the clustering-based studies including CEMs. Regarding the importance of this factor in the clustering's performance, different methods have been designed to determine the number of clusters. Comparing the optimization results for each number of clusters is a methodology that is used in some studies. However, it increases the computational time, especially for long-term CEMs. Elbow method, in contrast, determines the number of clusters with lower computational time because it uses Sum of the Squares of the point to the centroid Distance (SSD) in each cluster instead of the results of optimization. Section 3 provides a detailed discussion of the elbow method. However, this method has two main drawbacks: (1) it is visually dependent, leading to subjective cluster selection, and (2) it requires running the algorithm for all possible number of clusters. The former issue is addressed by silhouette method, which introduces a clear metric to select the best number of clusters [19]. Silhouette is a popular metric that determines the quality of the clusters by assessing the distance between cluster elements and the elements of different clusters. To determine the quality of clusters, silhouette coefficient for member i must be determined as follows [19]:

$$s(i) = \frac{b(i) - a(i)}{\max\left(b(i), a(i)\right)} \tag{1}$$

where $a(i)$ is the average distance between member i and the other members in the same cluster and $b(i)$ shows the smallest average distance between member i and the members from the other clusters. According to Equation (1), the silhouette coefficient for each member falls within the interval [-1,1], where 1 indicates that the member is well clustered, and -1 shows that the member is in a wrong cluster. Furthermore, the values for $a(i)$ and $b(i)$ are calculated using Equations (2) and (3), respectively. In these equations C_i shows the cluster that contains member i while C_k denotes the other clusters. Additionally, $d(i,j)$ represents the Euclidean distance between members i and j.

$$a(i) = \frac{1}{|C_i| - 1} \sum_{\substack{j \in C_i \\ j \neq i}} d(i,j) \tag{2}$$

$$b(i) = \min_{C_k \neq C_i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j) \tag{3}$$

After measuring the silhouette coefficient for all the members, the cluster silhouette coefficient is calculated by Equation (4). In this equation, N_C denotes the number of members in cluster C.

$$S(C) = \frac{1}{|C|} \sum_{i=1}^{N_C} s(i) \tag{4}$$

To find the best number of clusters, the silhouette coefficient for all

Table 1
Properties of clustering methods.

	RTP Duration Analysis		Number of Clusters			Clustering	
	Accuracy	Time	Selection		Complexity	Peak Effects	All Effects
			Arbitrary	systematic			
Ref. 9	×	×	✓		Low		✓
Ref. 13	×	×	✓		Low		✓
Ref. 22	×	×		✓	High		✓
Ref. 23	×	×	✓		Low		✓
Ref. 24	×	×		✓	High	✓	✓
Ref. 26	×	×		✓	Low	✓	
Ref. 27	✓	×		✓	Low	✓	
Ref. 28	×	×	✓		Low	✓	
Ref. 29	×	×	✓		Low	✓	✓
Ref. 30	×	×		✓	High	✓	✓
Ref. 31	×	×	✓		Low	✓	✓
Ref. 32	×	×		✓	High	✓	
Ref. 3	✓	✓	✓		Low		✓
Ref. 34	×	×	✓		Low	✓	✓
Ref. 1	×	×	✓		Low	✓	

the possible number of clusters is measured and then the one with the highest silhouette score will be selected as the best number of the clusters. Although this approach tackles the visuality problem of elbow method, still it needs to run on all the numbers of the clusters to find the best one, which means additional computational time.

Selecting the duration of RTPs is another research scope in the field of clustering for CEMs. To the best of our knowledge, day-based RTPs are the most used methodology for solving CEMs in various studies such as [9,36]. In contrast, some studies such as [31,37] consider weeks for RTPs. However, literature review shows that few studies investigate what duration is the best for clustering by considering different metrics.

Table 1 summarizes the key properties of the reviewed literature. As shown in the table, no existing study simultaneously addresses all the essential aspects of an effective clustering method. To bridge this gap, this paper proposes a novel clustering algorithm that considers all these aspects comprehensively.

3. Elbow method with stopping criterion

Selecting the optimum number of clusters is a crucial aspect of all the clustering methods. Despite the importance of it, there is no qualified method to precisely select the optimum number [12,38]. The classic elbow method is a visual toolbox to determine the optimum number of clusters that increasing the number of clusters gradually and measuring the quality of the clusters is the core idea of this method [38,39]. To determine clusters' quality, the classic elbow method uses the percentage variance for all of clusters, which is the Sum of the Squares of the point to the centroid Distance in each cluster. Nonetheless, some other studies utilize other factors like the result of optimization instead of percentage variance [40,41]. However, using the result of optimization increases the computational time because for each number of clusters the problem must be solved.

Elbow method starts by considering one cluster that includes all the data points and increases the number of clusters until it reaches the maximum possible number, where each cluster has only one member. Then, this method plots the number of clusters versus the percentage variance to find the best number of clusters. As the number of clusters increases, more similar data points are grouped together, leading to reduction in the within-cluster variance and subsequently the percentage variance. To this end, the plot has an overall decreasing trend. After a certain number of clusters, the plot does not change significantly, and the reducing trend becomes minimal. It indicates that increasing the number of clusters beyond that point does not improve the quality in terms of reducing percentage variance. This certain number of clusters is called the elbow point which is selected as the optimum number of the clusters [4,38].

Simplicity is for sure one of the positive aspects of this method, nonetheless there are two important weak points. The first issue stems from the visuality aspect of the elbow method. In some cases, it is not an easy task to discern the exact point because there is no clear inflexion. The second challenge arises from the computational time of the elbow method. This method must run for all the possible numbers of clusters. For example, if one year is considered to solve CEM on an hourly basis, the classic elbow method must run 8760 times, that is the maximum number of clusters, with each cluster containing just one member.

The problems of the elbow method can be addressed by designing a stopping criterion. Therefore, a modification is introduced in this paper. The designed stopping criterion is inspired by meta-heuristic optimization algorithms, where the process is continued until reaching a certain point. Unlike the preceding methods, where just percentage variance or the result of optimization are considered to determine the number of the clusters, this study considers two factors, runtime and percentage variance, to design the stopping criterion. To put it more clearly, EMSC monitors the changes in the derivative of percentage variance and runtime to find the best number of clusters. Complexity of CEM is the main reason for selecting percentage variance over the results of optimization. In other words, using the result of optimization requires running CEM many times, which is not feasible within the concept of selecting RTPs.

Equation (5) shows the way EMSC calculates the change in percentage variance (Δ_{PVa}^{cn}). EMSC employs the ratio between the percentage variances with cn clusters (PVa_{cn}) and $cn-1$ clusters (PVa_{cn-1}). Furthermore, the change in runtime (Δ_{RT}^{cn}) is measured by Equation (6). Again, EMSC uses the ratio between runtimes with cn clusters (RT_{cn}) and $cn-1$ clusters (RT_{cn-1}). Finally, the derivative change of the corresponding number of clusters (DC^{cn}) is calculated by Equation (7). The main reason for using ratios in EMSC is the difference between percentage variances and runtime in terms of quantity. Based on the implementations, while percentage variances are greater than one million, the runtime for each cluster number is less than 1 s. Using ratios can balance the range of values of the mentioned elements.

EMSC measures the derivative change as the number of clusters increases with each iteration. The derivative change decreases after a certain number of clusters because increasing the number of clusters beyond that number does not change the percentage variance and runtime significantly. The stopping criterion in this study is defined according to the derivative change and how it tends to zero. The stopping criterion is reached if the derivative change for some consecutive numbers of clusters (α) is less than a predefined threshold value (β). Then, the first number of clusters (cn) is selected as the optimum number of clusters. The process of EMSC is described in Algorithm 1.

Table 2
Properties of generation sources.

Source	i_g^G ($€/MW$)	y_g^G (*years*)	*Maximum Capacity* (*MW*)	*Availibility* (*p.u.*)
Wind	1.5×10^6	25	366900 ($\widehat{p}_{wind}$)	–
Solar	1×10^6	25	221600 ($\widehat{p}_{solar}$)	–
Hydrogen	1.2×10^6	50	14700 ($\widehat{h}$)	0.25 (a_n)

Table 3
Properties of storage units.

Storage	η_s (h)	ξ_s (*p.u.*)	i_s^S ($€/MW$)	y_s (*years*)
Intraday	6	0.8	1.5×10^6	80
Interday	48	0.7	2×10^6	60

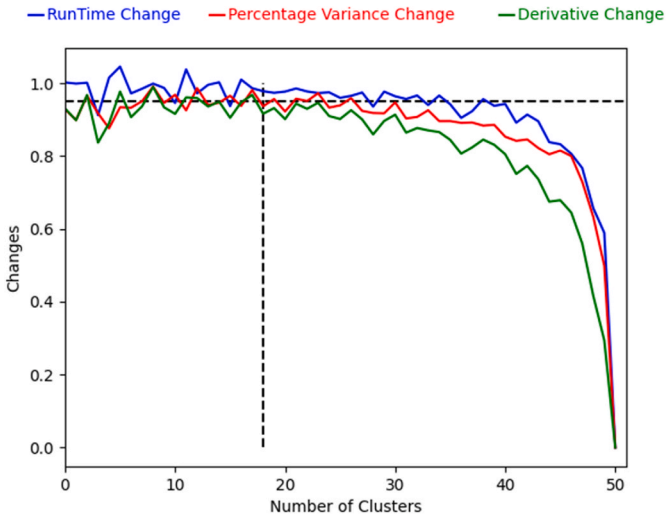


Fig. 2. Result of EMSC.

Table 4
Implementation results of EAHCA.

	Full Data	*Weekly* (18)	*Daily* (126)	*Hourly* (3024)
Overall Cost($10^9€$)	8.57	7.88	7.65	7.63
Wind (*GW*)	76.95	82.09	70.95	71.04
Solar (*GW*)	73.05	50.25	58.53	61.23
Hydrogen (*GW*)	14.7	14.70	14.70	14.70
Intraday (*GW*)	11.87	4.08	11.00	14.04
Interday (*GW*)	13.71	14.84	14.95	9.20

Table 5
Error rate of cluster formats.

	Weekly (18)	*Daily* (126)	*Hourly* (3024)
Overall Cost Error (%)	8	10	10
Wind Error (%)	6	7	7
Solar Error (%)	31	19	16
Hydrogen Error (%)	0	0	0
Intraday Error (%)	65	7	18
Interday Error (%)	8	9	32
Overall Error (%)	118	52	83

$$\Delta_{PVa}^{cn} = \frac{PVa_{cn}}{PVa_{cn-1}} \tag{5}$$

$$\Delta_{RT}^{cn} = \frac{RT_{cn}}{RT_{cn-1}} \tag{6}$$

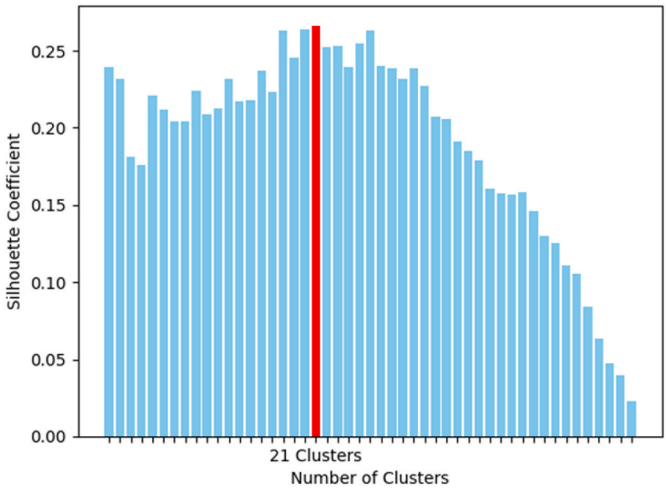


Fig. 3. Silhouette coefficients.

$$DC^{cn} = \Delta_{PVa}^{cn} \times \Delta_{RT}^{cn} \tag{7}$$

Algorithm 1
EMSC

```
Begin:
  Counter = 0
  for cn in the rage of (1, potential number of clusters):
    Start timer = time.time()
    Cluster input data, respecting cn
    End timer = time.time()
    Measure RT_cn based on Start and End timer
    Calculate PVa_cn
    if cn = = 1:
      DC^cn = PVa_cn × RT_cn
    else:
      Δ^cn_PVa = PVa_cn / PVa_cn−1
      Δ^cn_RT = RT_cn / RT_cn−1
      DC^cn = Δ^cn_PVa × Δ^cn_RT
    if DC^cn ≤ β:
      Counter + = 1
    if Counter = = α:
      Break for
  Optimum Number of Clusters = Counter − (α − 1)
end
```

4. Euclidean based hierarchical clustering

In [3], an agglomerative hierarchical clustering algorithm with the purpose of maintaining the chronological order of the members in each cluster is presented. The clustering method proposed in this paper is a modified version, where a representative vector is selected according to both mean and medoid. The process of EAHCM is outlined in Algorithm 2. EAHCM is composed of two main steps, selecting the representative vector and cluster merging, both are explained in the following subsections.

Table 6
EMSC and Silhouette comparison.

Number of Clusters	Weekly		Daily		Hourly	
	silhouette (21)	EMSC (18)	silhouette (147)	EMSC (126)	silhouette (3528)	EMSC (3024)
Overall Cost Error (%)	**7**	8	16	**10**	12	**10**
Wind Error (%)	**3**	6	**6**	7	**5**	7
Solar Error (%)	**28**	31	27	**19**	61	**16**
Hydrogen Error (%)	**0**	**0**	**0**	**0**	**0**	**0**
Intraday Error (%)	**57**	65	12	**7**	23	**18**
Interday Error (%)	25	**8**	10	**9**	45	**32**
Overall Error (%)	120	**118**	71	**52**	146	**83**

Table 7
Results of solving CEM.

18 Clustered Weeks								
	Full	CTPC	TKM	CKM	CHC	MKMA	MMDA	EAHCM
Overall Cost(10^9€)	8.57	7.79	5.70	6.30	5.85	5.48	6.72	7.88
Wind (*GW*)	76.95	79.36	0.00	0.00	0.00	0.00	0.00	82.09
Solar (*GW*)	73.05	49.64	116.94	130.82	120.62	109.88	142.16	50.25
Hydrogen (*GW*)	14.7	14.70	14.70	14.70	14.70	14.70	14.70	14.70
Intraday (*GW*)	11.87	0.21	0.00	0.00	0.00	0.00	0.18	4.08
Interday (*GW*)	13.71	20.48	20.02	21.42	20.08	21.99	20.51	14.84
126 Clustered Days								
	Full	CTPC	TKM	CKM	CHC	MKMA	MMDA	EAHCM
Overall Cost(10^9€)	8.57	7.53	5.54	5.60	5.61	5.40	6.15	7.65
Wind (*GW*)	76.95	73.50	0.00	0.00	0.00	0.00	0.00	70.95
Solar (*GW*)	73.05	54.29	112.65	111.80	115.32	109.60	125.36	58.53
Hydrogen (*GW*)	14.7	14.70	14.70	14.70	14.70	14.70	14.70	14.70
Intraday (*GW*)	11.87	10.01	0.74	0.00	0.82	0.00	0.00	11.00
Interday (*GW*)	13.71	12.49	20.14	23.50	18.74	19.84	23.72	14.95
3024 Clustered Hours								
	Full	CTPC	TKM	CKM	CHC	MKMA	MMDA	EAHCM
Overall Cost(10^9€)	8.57	7.85	5.22	4.01	5.45	5.14	6.29	7.63
Wind (*GW*)	76.95	78.97	0.00	0.00	0.00	0.00	0.00	71.04
Solar (*GW*)	73.05	54.99	111.65	73.53	115.99	109.75	131.66	61.23
Hydrogen (*GW*)	14.7	14.70	14.70	14.70	14.70	14.70	14.70	14.70
Intraday (*GW*)	11.87	12.35	20.82	8.56	16.75	20.94	8.33	14.04
Interday (*GW*)	13.71	9.90	0.00	16.71	4.06	0.00	15.56	9.20

Algorithm 2. EAHCM

```
Begin:
  Find Optimum Number of Clusters with EMSC
  Cluster Numbers = n
  While Cluster Numbers > Optimum Number of Clusters
    Determine the representative vector of each cluster based on Equation 10
    Calculate dissimilarity between all the adjacent clusters with Equation 11
    Merge to closets adjacent clusters
    Cluster Numbers = Cluster Numbers − 1
end
```

4.1. Selecting representative vector

The mean and medoid vectors of each cluster are the main two approaches to represent the cluster, which have been used in different studies. In the first option, the mean vector can absorb the effects of all the vectors in the cluster. However, it tends to distort the peak values. On the other hand, the medoid can maintain the chronological order of the data because it considers one vector of the cluster and can improve the representation of peak values. However, it disregards the effects of the other members. To this end, EAHCM uses both approaches to take advantage of the two options. To select a proper representative vector, EAHCM calculates the Euclidean distance between the members of the clusters and the mean (RQ_{medoid}) and medoid (RQ_{mean}) vectors (see Equations (8) and (9), respectively).

$$RQ_{medoid} = \sum_{j \in Cluster} \left\| X_{medoid} - member_j \right\| \tag{8}$$

$$RQ_{mean} = \sum_{j \in Cluster} \left\| \overline{X}_{mean} - member_j \right\| \tag{9}$$

where X_{medoid} is the medoid vector and $\overline{X}_{mean}$ is the mean vector of the cluster.

Afterwards, EAHCM chooses the option that has minimum distance with the others. In other words, EAHCM selects the option that is closer to the other members. In this way, the representative vector will be more similar to the other members within the cluster, thus the chronological order of the original data is preserved. Equation (10) shows the selection part of EAHCM where $RV_{cluster}$ is the representative vector of the corresponding cluster.

$$RV_{cluster} = \min(RQ_{medoid}, RQ_{mean}) \tag{10}$$

4.2. Cluster merging

Following the selection of RTPs in the clusters, EAHCM measures the dissimilarity between the representative vectors of all the adjacent clusters. Then, two adjacent clusters exhibiting the highest similarity will be merged. To determine the dissimilarity between two clusters, EAHCM updates Ward's equation [25] according to the introduced method in Section 4.1. Equation (11) finds the dissimilarity of two

Table 8
Error in clustering methods.

18 Clustered Weeks							
	CTPC	TKM	CKM	CHC	MKMA	MMDA	EAHCM
Overall Cost Error (%)	9	33	26	31	36	21	**8**
Wind Error(%)	**3**	100	100	100	100	100	6
Solar Error(%)	32	60	79	65	50	94	**31**
Hydrogen Error (%)	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Intraday Error(%)	98	100	100	100	100	98	**65**
Interday Error (%)	49	46	56	46	60	49	**8**
Overall Error (%)	191	339	361	342	346	362	**118**
126 Clustered Days							
	CTPC	TKM	CKM	CHC	MKMA	MMDA	EAHCM
Overall Cost Error (%)	12	35	34	34	36	28	**10**
Wind Error(%)	**4**	100	100	100	100	100	7
Solar Error(%)	25	54	53	057	050	71	**19**
Hydrogen Error (%)	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Intraday Error(%)	15	93	100	93	100	100	**7**
Interday Error (%)	**8**	46	71	36	44	73	9
Overall Error (%)	64	328	358	320	330	372	**52**
3024 Clustered Hours							
	CTPC	TKM	CKM	CHC	MKMA	MMDA	EAHCM
Overall Cost Error (%)	**8**	39	53	36	40	26	10
Wind Error(%)	**2**	100	100	100	100	100	7
Solar Error(%)	24	52	**1**	58	50	80	16
Hydrogen Error (%)	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Intraday Error(%)	**0**	75	27	41	76	29	18
Interday Error (%)	27	100	21	70	100	**13**	32
Overall Error (%)	**65**	366	202	305	366	248	83

Table 9
Average runtime of Solving CEM based on clustered data.

Format	Runtime(sec.)	CTPC	TKM	CKM	CHC	MKMA	MMDA	EAHCM
Week	*Clustering*	0.19	0.09	0.01	0.01	0.17	0.31	0.15
	Solving CEM	2.24	0.01	2.08	3.43	0.01	5.10	2.03
	Overall	2.43	**0.10**	2.09	3.44	0.18	5.41	2.18
Day	*Clustering*	1.39	0.11	0.02	0.02	0.74	8.39	3.06
	Solving CEM	4.08	0.01	5.44	7.47	0.02	3.27	3.27
	Overall	5.47	**0.12**	5.46	7.49	0.76	11.66	6.33
Hour	*Clustering*	55.23	30.13	60.97	1.03	44.21	22491.52	276.37
	Solving CEM	3.31	16.77	6.26	7.65	0.12	3.65	3.18
	Overall	58.54	46.30	67.23	**8.68**	44.33	22495.14	279.55

adjacent clusters.

$$D_{EAHCM}(I,J) = \frac{2|I||J|}{|I|+|J|}\|RV_I - RV_J\|^2 \tag{11}$$

where RV_I and RV_J are the representative vectors of clusters I and J, respectively. At each iteration, EAHCM merges two clusters, and this process continues until reaching the optimum number of clusters.

5. Capacity expansion model

This section designs a CEM for an isolated energy plant that is powered just by renewable energy sources, such as wind, solar and hydrogen. Since the plant is isolated, it cannot buy or sell energy to the grid and its demand must be met exclusively through its own renewable generation. Additionally, the plant is equipped with two types of batteries, interday and intraday that can be charged when there is excess generation and discharged during periods of energy shortage. The model in this study is an updated version of [3].

The designed CEM determines the optimal generation and storage capacities for the plant to minimize overall costs while ensuring sufficient energy is generated to meet demand. Regarding its objectives, the objective function consists of four components: two measure the cost of energy generation, one accounts for the cost of storage units, and one represents the cost of load shedding. Load shedding refers to the deliberate reduction of electricity consumption when supply is insufficient to meet demand. It is typically used as a last resort to maintain the plant stability and prevent system failures. In the proposed CEM, a penalty is considered for load shedding to discourage energy shortages in the plant. Moreover, the CEM in this study is based on deterministic optimization that ignores the uncertainty of the parameters in the optimization process. The linear optimization model of CEM is formulated such that each variable corresponds to the length of the input data. In other words, if the model runs for one full year, each variable has a length of 8,760, matching the number of hours in a year. The model is represented below.

$$\min \sum_{nt} \tau_t \omega_t sc(\bar{d}_t - d_t) + \sum_g \frac{i_g^G}{y_g^G}\bar{p}_g + \sum_s \frac{i_s^S}{y_s^S} b_s + \sum \frac{i^H}{y^H}\bar{h} \tag{12a}$$

$$0 \le \bar{p}_g \le \hat{p}_g, \forall g \epsilon G^r \tag{12b}$$

$$0 \le \bar{h} \le \hat{h} \tag{12c}$$

$$\sum_g p_{gt} + \sum_s \left(b_{st}^- - b_{st}^+\right) + h_t = d_t \tag{12d}$$

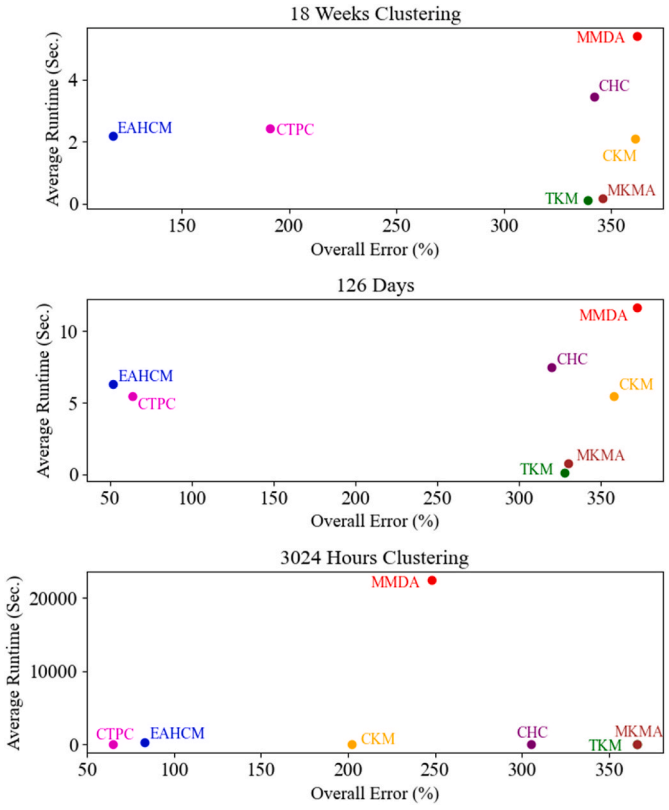


Fig. 4. Overall error versus average runtime.

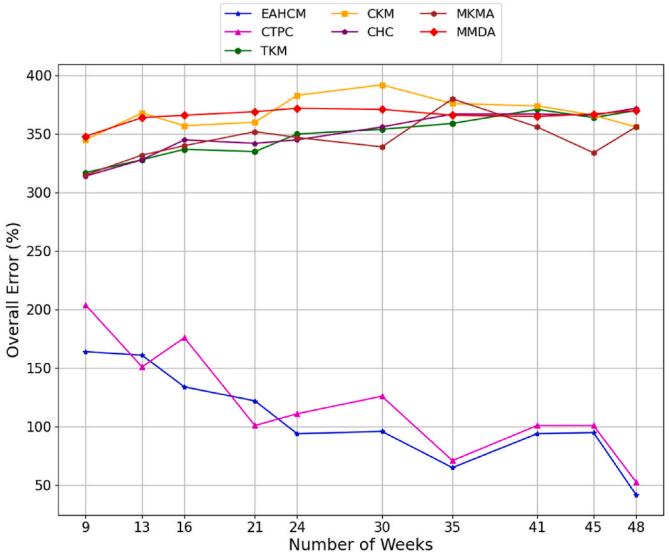


Fig. 5. Overall Error rates of different clustering methods on different weekly basis.

$$0 \le p_{gt} \le \rho_{gt}\left(\overline{p}_g^0 + \overline{p}_g\right), \forall g, t \tag{12e}$$

$$0 \le h_t \le \overline{h}^0 + \overline{h}, \forall t \tag{12f}$$

$$\sum_t \tau_t \omega_t h_t \le a_n \sum_t \tau_t \omega_t \left(\overline{h}^0 + \overline{h}\right) \tag{12g}$$

$$0 \le b_{st}^+ \le \overline{b}_s^0 + \overline{b}_s, \forall s, t \tag{12h}$$

$$0 \le b_{st}^- \le \overline{b}_s^0 + \overline{b}_s, \forall s, t \tag{12i}$$

$$b_{st} = b_{st-1} - \tau_t b_{st}^- + \tau_t \xi_s b_{st}^+, \forall s, t \ne 1 \tag{12j}$$

$$0 \le b_{sn} \le \eta_s \left(\overline{b}_s^0 + \overline{b}_s\right) \tag{12k}$$

$$0 \le d_t \le \overline{d}_t \tag{12l}$$

As Equation (12a) represents, minimizing the overall cost of the plant is the main objective, which includes load shedding cost ($\sum_{nt} \tau_t \omega_t sc(\overline{d}_t - d_t)$), the cost of investment in renewable energy ($\sum_g \frac{i_g^G}{y_g^G} \overline{p}_g$), storage units cost ($\sum_s \frac{i_s^S}{y_s^S} b_s$), and finally hydrogen power cost ($\sum \frac{i^H}{y^H} \overline{h}$).

All the energy plants have a maximum installable capacity for energy production that cannot be exceeded because ignoring these constraints may lead to unrealistic energy supply and demand forecasts. Therefore, Equations (12b) and (12c) are designed to limit the renewable and hydrogen capacity to a maximum installable capacity. Additionally, energy plants must balance their production with demand at each time step. Because this study considers an isolated plant, the energy generation sources are limited to renewables, storage, and hydrogen that must meet the demand at the corresponding time step. This balance is ensured by Equation (12d). Also, the considered energy sources must be limited to the maximum possible generation that is modeled by Equations (12e) and (12f) for renewable sources and hydrogen, respectively. Moreover, Equation (12g) sets the maximum hydrogen energy available over the planning horizon.

Regarding the utilization of storage units in this study, the constraints related to them must be modeled, too. The first constraint is the charging and discharging levels, that they must not exceed the capacity of the battery. Equations (12h) and (12i) are designed for this issue, while Equation (12j) incorporates round-trip efficiency to determine the storage energy level at each time step. Obviously, the stored energy at each time-period must be less than the capacity. Equation (12k) models this constraint. Finally, Equation (12l) makes sure that the served demand is restricted to the demand level at each time-period.

6. Numerical results

In this paper, EAHCM has been implemented with Pyomo 6.6.1 along with Gurobi solver in Python on a Razer Blade laptop equipped with CPU of AMD Ryzen 9 8945HX, Radeon Graphics 3.30 GHz, and 32 GB of RAM. Pyomo is an open-source Python library for formulating, solving, and analyzing mathematical optimization models such as CEM. It provides a flexible framework for defining linear, nonlinear, integer, and mixed-integer programming problems. Additionally, Gurobi is a commercial optimization solver known for its high performance in solving linear programming, mixed-integer programming, and quadratic programming problems. It is widely used in research and industry due to its speed, robustness, and advanced features like parallel computing and cutting-edge algorithms.

This study examines generation and storage in Spain for a single target year, 2030. As section 5 explains, to run the linear optimization model the properties of solar, wind, and hydrogen generations, including their maximum installable capacities in Spain must be determined. Table 2 details these properties. Additionally, storage capabilities must be defined beforehand to run the linear optimization model, with Table 3 providing the relevant details. Even though CEMs are long-term problems by nature, one single year is considered because CEM needs to be solved on the full-time horizon dataset to measure the accuracy of the results. In other words, in the case of a long-time capacity expansion model with a large dataset the optimization will become intractable. Therefore, the accuracy of clustering strategies cannot be evaluated. Subsequently, the demand in the target year must be

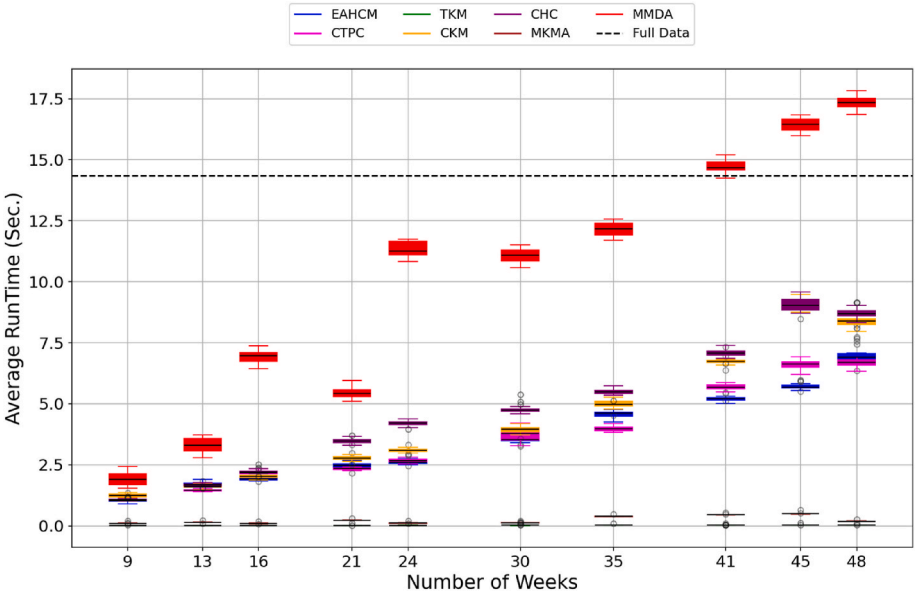


Fig. 6. Average Runtimes of different clustering methods on different weekly basis.

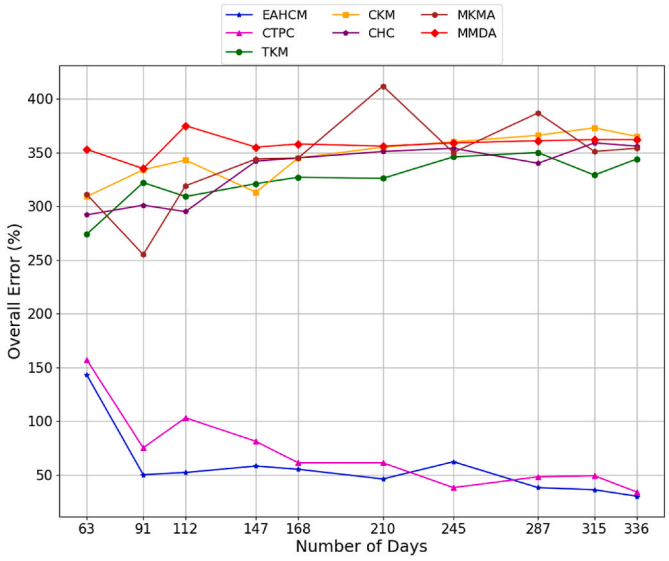


Fig. 7. Overall Error rates of different clustering methods on different daily basis.

predicted. This study considers the Spanish demand with 1 % growth since 2017, which is obtained from red electrica de España, the Spanish grid operator.[1] Finally, this study assumes 1000€/MWh as the load shedding cost.

Runing EMSC to determine the number of clusters is the first step of the proposed clustering method. To run EMSC, demand data that is arranged in the format of weeks, is used. Based on Algorithm 1, two values must be determined in advance to run EMSC: the predefined threshold value (β) and the number of consecutive derivative changes (α). This study uses empirical values to run the EMSC, with the threshold value set to 0.95 ($\beta = 0.95$) and the number of consecutive derivative changes set to 5 ($\alpha = 4$). Fig. 2 demonstrates the changes in derivative, runtime, and percentage variance resulting from running EMSC. As it is depicted, by increasing the number of clusters beyond 18, the derivative change does not cross the threshold. It means that 18 is the optimum number of clusters. Although the process of EMSC does not continue after 22 clusters because the stopping criterion is reached, the full process is shown in Fig. 2 to illustrate the behavior of the derivative change. As expected, the derivative change falls with increasing the number of clusters, because the change in percentage variance (Δ^{cn}_{PVa}) and the change in runtime (Δ^{cn}_{RT}) are decreasing. This means that increasing the number of clusters does not significantly change the percentage variance and runtime, making it unnecessary. According to Fig. 2, Δ^{cn}_{PVa} and Δ^{cn}_{RT} record values greater than one when the number of clusters is small. Therefore, the plot is not monotonous when the number of clusters is less than 18.

Since the proposed method works based on runtime, with potential variation from one iteration to another, the process is repeated 30 times. In all the 30 runs, the general trend of the derivative change plot stays the same. However, the selected optimum number of clusters changes throughout the runs. From 30 runs, EMSC identifies 18 as the optimum number of clusters in twenty-one runs, while 24 is selected in six runs and 16, 27, and 30 are chosen in just one run.

Once the optimum number of clusters is determined, data must be clustered. To solve CEM, other data, like capacity factors of solar, wind, and hydrogen energies are required. Therefore, these datasets are incorporated in the clustering process alongside the demand data. In addition to the clustered datasets, the weight of each cluster must be determined to optimize the CEM objective function. The weight of each cluster is the number of datapoints, hours, days, or weeks that it includes.

Table 4 shows the results of solving CEM on the hourly, daily, and weekly clustered datasets. As the table indicates, this study centers its attention on six key parameters: overall cost, wind, solar, hydrogen, intraday, and interday, which are the fundamental elements of capacity planning model. Considering 18 weeks as the optimum number of clusters implies that the optimum number of clusters on hourly and daily basis are 3024 and 126, respectively. Moreover, to find the accuracy of the clustering formats, the identical problem is solved based on a full year data. It is straightforward that the results of a better clustering method must be closer to the full data outcomes.

As Table 4 shows, the six key elements take values in different ranges, which is a challenge to measure accuracy. To this end, this study employs Equation (13), where the difference of each key element resulting from clustering format with the full data format is calculated

[1] https://www.esios.ree.es/es.

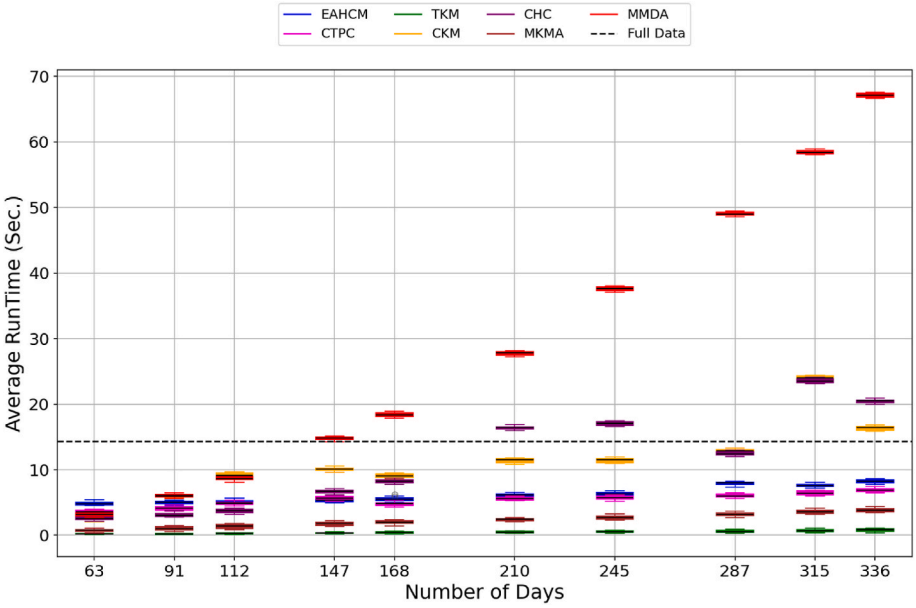


Fig. 8. Average Runtimes of different clustering methods on different weekly basis.

first. Then, the difference value is normalized. Finally, the sum of the mapped values is considered as the error associated with the corresponding cluster format. Table 5 stands for the error of each cluster format. As the table indicates, daily clustering format has the lowest error among the others.

$$Error = \sum_{i} \frac{|full\ data_i - cluster\ format_i|}{full\ data_i} \times 100 \tag{13}$$

7. Comparison

This section draws a comparison between the proposed clustering method and other well-know methods. First, the performance of EMSC is compared with Silhouette method on accuracy to investigate which of them determines a better number of clusters. Then, EAHCM is compared with traditional k-Means (TKM) [21], classic k-Medoids (CKM) [25], classic hierarchical clustering (CHC) [33], Chronological Time-Period Clustering (CTPC) [3], modified k-Means algorithm (MKMA) [23], and modified maximum dissimilarity algorithm (MMDA) [34], on three key metrics: accuracy, runtime, and sensitivity analysis.

7.1. Number of clusters

As discussed earlier, determining a proper number of clusters is a basic challenge for all the clustering approaches. Although some algorithms are developed in different studies, almost all of them such as Silhouette need to cluster the input data for all the possible number of clusters, leading to a high computational time. Therefore, it can be mentioned that the proposed EMSC has better performance in comparison with silhouette method on the runtime because it does not need to run for all the possible number of clusters.

Accuracy is another key comparison metric. To assess this, EAHCM is used to cluster data based on the number of clusters EMSC and Silhouette determine and measure the accuracy of solving CEM for each clustered data. Fig. 3 represents silhouette coefficients for all the possible number of clusters on a weekly basis. As shown, the coefficient reaches its maximum with 21 clusters, indicating that 21 is the best number of clusters.

Thus, the input data is clustered into 18 and 21 clusters, as determined by EMSC and the Silhouette method, respectively. Table 6 presents the error rate for each cluster on an hourly, daily, and weekly basis. As shown, CEM achieves higher accuracy with the number of clusters determined by EMSC compared to silhouette.

7.2. Clustering accuracy

Accuracy is the first and the most important metric to compare the clustering methods. To measure accuracy, CEM is solved based on the full data and the clustered data generated by the clustering methods. Table 7 shows the results of solving CEM based on weekly, daily, hourly clustering methods. As the table shows, six key factors resulting from solving CEM are taken into consideration to measure accuracy. Closer values to the full data reflect better clustering methodology. To rank the clustering methods, the error rates are analyzed by means of Equation (13). The errors in running the clustering methods on weekly, daily, and hourly basis are presented in Table 8.

As the results show, EAHCM outperforms the other methods in the weekly and daily formats. Also, it achieves the second highest accuracy on an hourly basis. These findings prove the effectiveness of the proposed method in comparison with other well-known clustering strategies. Furthermore, the results confirm that running the proposed method on a daily basis reaches the lowest error rate among the other options.

7.3. Clustering runtime

Reducing the computational burden of CEM is the primary motivation for using clustering methodology. Consequently, this section provides an in-depth analysis of the runtime performance of the clustering methods. Since the runtime changes from run to run, to have an accurate overview of this metric, each clustering method is run 30 times. It is also important to emphasize that all algorithms are implemented on the same hardware to ensure a fair comparison.

According to the implementation results, although the runtime varies slightly across runs, the clustering methods consistently produce the same results. Consequently, the optimization structure operates on identical clustered data which results in the same error rate. Table 9 shows the average runtime of the clustering method on a weekly, daily, and hourly basis. As the table shows, for a more in-depth analysis, the runtime is broken down into two sections: clustering and solving CEM. The overall runtime is then the sum of these two sections. Based on the findings, clustering runtime on a weekly basis is lower than the other intervals. However, the execution time of solving CEM based on the

clustered data remains almost the same on weekly, daily, and hourly basis. Furthermore, based on the overall runtime, TKM demonstrates the fastest methodology on a weekly and daily basis, while CHC exhibits the fastest performance on an hourly basis.

The experimental results show that solving CEM on the full data requires approximately 14.36 s. Regarding the main purpose of using clustering methodology, the overall runtime of the clustering methods must be less than 14.36 s. Therefore, it becomes evident from the results of Table 9 that clustering on an hourly basis is meaningless because clustering methods, except CHC, do not diminish the computational burden of CEM.

In addition, to have a clear overview of the performances, accuracy and runtime metrics must be considered together. Fig. 4 compares the overall error rate versus the average runtime of the clustering methods on weekly, daily, and hourly basis. Obviously, a method which is closer to zero reflects better performance in terms of accuracy and runtime. As the figure illustrates, EAHCM has the lowest error rate among all, on the weekly and daily clustering, while it reduces the computational burden of CEM. Only on an hourly basis clustering, CTPC has better performance than EAHCM. However, none of the methods are suitable for the hourly basis clustering because they do not diminish the computational time of CEM in comparison with the full data.

7.4. Clustering robustness

Sensitivity analysis is a valuable tool for measuring the robustness of the clustering methods. In this analysis, some parameters are systematically changed to observe the performance of the clustering methods. Based on the obtained results in sections 7.2 and 7.3, clustering on an hourly basis can be ignored due to its computational time. Thus, this section undertakes the sensitivity analysis of the clustering methods, focusing just on weekly and daily intervals.

Although selecting the optimum number of clusters is one of the novelties of this study, this section changes the number of clusters for the sake of sensitivity analysis. In this study, 10 number of clusters are selected in a random way for weekly and daily clustering. Clustering methods are executed based on these number of clusters and the error rates and runtimes are analyzed. As mentioned earlier, due to the slight fluctuation in runtime, the clustering methods are run 30 times on each number of clusters. However, repeating the runs is not necessary to assess the error rate performance because it remains constant throughout the runs. The selected values are 9, 13, 16, 21, 24, 30, 35, 41, 45, and 48 weeks, which are equal to 63, 91, 112, 147, 168, 210, 245, 287, 315, and 336 days.

Fig. 5 depicts how the error of the clustering methods fluctuates with changing the number of clusters on a weekly basis. It can be clearly seen that the performances of EAHCM and CTPC are far better than the other implemented methods. Moreover, the proposed method exhibits superior performance compared to CTPC in 8 cases out of 10, which depicts the robustness of EAHCM.

Fig. 6 shows how changing the number of weeks impacts the overall runtime of the clustering methods. This figure uses boxplots demonstrating the variation of 30 implementations where the average of these runs is shown with a black line in each box. Obviously, a clustering method with shorter average runtime will have a better performance in this metric. As the figure illustrates, k-Means based clustering methods, TKM and MKMA, demonstrate faster execution time on average compared to the others. Also, the figure indicates that MMDA is unable to reduce the computational burden of CEM for 41, 45 and 48 clusters.

In general, increasing the number of clusters results in higher runtime. However, the runtime fluctuates for some methods on special number of clusters. The experiments indicate that while the clustering runtime remains stable, the runtime for solving CEM spikes in some cases and it is the main reason for these fluctuations. It highlights how the quality of the final clusters can affect the optimization process.

Regarding the figure, in the worst-case scenario, the average runtime of EAHCM is almost 50 % less than that of the full data. This time complexity reduction is even greater on other number of clusters. Even though the proposed method is not the fastest clustering method, with putting the accuracy alongside runtime, it can easily be concluded that the proposed method stands out as the most robust method for weekly clustering.

Fig. 7 measures the robustness of the clustering methods in response to changes in the number of days. Similar to weekly clustering, EAHCM and CTPC exhibit superior performance in comparison with the other methods. Also, comparing the proposed method with CTPC reveals that EAHCM results in a lower error rate in 9 out of 10 cases. So, the proposed method is the best option for daily clustering among the others in accuracy metric.

The average runtimes of the clustering methods daily along with the variations of the runs are plotted in Fig. 8. It is evident that MMDA is not a proper daily clustering method because with increasing the number of clusters, its computational time increases dramatically. The same problem happens for CHC and CKM in four and two clusters, respectively. Regarding the findings of the figure, k-Means based clustering methods, TKM and MKMA, have the fastest execution times; However, their performances are very poor in terms of accuracy. Similar to sensitivity analysis on weekly clustering, considering the performances of the clustering methods in both metrics will lead to selecting EAHCM as the most robust option for daily clustering.

8. Conclusions

This study presents a novel hierarchical clustering method to diminish the computational burden of capacity expansion modeling. The proposed clustering method, EAHCM, begins by determining the optimum number of clusters. Unlike the preceding approaches, a systematic method called EMSC is introduced to enhance classic elbow method with designing a stopping criterion. Afterwards, EAHCM clusters data by means of merging two adjacent clusters that have the highest similarity between their representative vectors. To determine a proper representative vector for each cluster, the proposed method takes the advantage of both medoid and mean strategies. In this way, the proposed method retains the chronological information of the original data.

To evaluate the performance of the proposed clustering method, first, it is compared with well-known methods for determining the optimal number of clusters, such as the silhouette. The comparison results show that EMSC outperforms other methods in terms of computational time. Unlike traditional methods, which must run for all the possible number of clusters, the proposed method incorporates a stopping criterion to prevent unnecessary computations. Furthermore, data has been clustered based on the number of clusters determined by silhouette and EMSC to assess clustering accuracy. Implementation results show that EMSC reduces clustering errors compared to the silhouette method by 2 %, 19 %, and 63 % for hourly, daily, and weekly clustering, respectively.

Additionally, the performance of the proposed clustering algorithm has been compared with well-known clustering methods, including traditional k-Means, classic k-Medoids, classic hierarchical clustering, Chronological Time-Period Clustering, the modified k-Means algorithm, and the modified maximum dissimilarity algorithm. The comparison is done based on three key metrics: accuracy, runtime, and sensitivity analysis. Numerical results demonstrate that EAHCM on an hourly basis achieves the highest accuracy, 48 %, in comparison to other clustering methods.

To accurately analyze the runtimes, clustering and solving CEM runtimes are measured separately in this study. This investigation reveals that the proposed EAHCM diminishes the runtime of CEM by more than 50 % on daily and weekly clustering. However, it is not the fastest algorithm overall, as k-Means based clustering methods outperform others in this metric. Finally, the robustness of the proposed clustering method is evaluated through sensitivity analysis, where it is tested

across different numbers of clusters. The findings indicate that EAHCM outperforms the others in terms of accuracy for almost all the number of clusters.

Although EAHCM achieves high accuracy and low computational time for daily and weekly clustering, its performance on an hourly basis is not acceptable. It fails to reduce the computational burden of CEM and has a high error rate in this format of clustering. Not only the proposed clustering method but also the other investigated clustering methods suffer from the same limitation on hourly clustering, which defines the scope of the future research. In the future work, this issue will be addressed by designing a parallel clustering approach.

Additionally, the clustering method will be tested on a more realistic CEM in the future study. In this study, a simple deterministic CEM is used that ignores uncertainty in the input data, even though such uncertainty can significantly affect optimization results. To overcome this limitation, a stochastic CEM will be developed in the future study that incorporates different scenarios to account for uncertainty. Moreover, the current study does not consider the cost of storage degradation, which can impact storage operation and potentially change optimization outcomes. This particular limitation will be addressed in the future study by using the PyBaMM package in Python, which predicts battery's state of health based on operational conditions.

CRediT authorship contribution statement

Milad Riyahi: Writing – original draft, Software, Methodology, Investigation, Conceptualization. **Alvaro Gutiérrez Martín:** Writing – review & editing, Supervision, Methodology, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

Acknowledgement

This work has been funded by the Universidad Politécnica de Madrid Project "SDGine for Healthy People and Cities" which received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Marie Sklodowska-Curie grant agreement No. 945139 and from REPSOL, S.A.

Data availability

Data will be made available on request.

References

[1] Domínguez R, Vitali S. Multi-chronological hierarchical clustering to solve capacity expansion problems with renewable sources. Energy 2021;227:120491.
[2] Zhou Y, Zhai Q, Yuan W, Wu J. Capacity expansion planning for wind power and energy storage considering hourly robust transmission constrained unit commitment. Appl Energy 2023;302:117570.
[3] Pineda S, Morales JM. Chronological Time-Period clustering for optimal capacity expansion planning with storage. IEEE Trans Power Syst 2018;33(6):7162–70.
[4] Teichgraeber H, Brandt AR. Time-series aggregation for the optimization of energy systems: goals, challenges, approaches, and opportunities. Renew Sustain Energy Rev 2022;157:111984.
[5] Murphy FH, Smeers Y. Generation capacity expansion in imperfectly competitive restructured electricity markets. Oper Res 2005;53(4):646–61.
[6] Roh JH, Shahidehpour M, Wu L. Market-based generation and transmission planning with uncertainties. IEEE Trans Power Syst 2009;24(3):1587–98.
[7] García-Cerezo A, García-Bertrand R, Baringo L. Priority chronological Time-Period clustering for generation and transmission expansion planning problems with long-term dynamics. IEEE Trans Power Syst 2022;37(6):4325–39.
[8] Merrick JH. On representation of temporal variability in electricity capacity planning models. Energy Econ 2016;59:261–74.
[9] Li C, Conejo AJ, Siirola JD, Grossmann IE. On representative day selection for capacity expansion planning of power systems under extreme operating conditions. Int J Electr Power Energy Syst 2022;137:107697.
[10] Wen S, Zhang W, Sun Y, Li Z, Huang B, Bian S, Zhao L, Wang Y. An enhanced principal component analysis method with Savitzky–Golay filter and clustering algorithm for sensor fault detection and diagnosis. Appl Energy 2023;337:120862.
[11] Kaufman L, Rousseeuw PJ. Finding groups in data: an introduction to cluster analysis. Chichester, U.K.: Wiley; 1990.
[12] Norambuena-Guzmán V, Palma-Behnke R, Hernández-Moris C, Cerda MT, Flores-Quiroz A. Towards CSP technology modeling in power system expansion planning. Appl Energy 2024;364:123211.
[13] Mallapragada DS, Papageorgiou DJ, Venkatesh A, Lara CL, Grossmann IE. Impact of model resolution on scenario outcomes for electricity sector system expansion. Energy 2018;163:1231–44.
[14] Kotzur L, Markewitz P, Robinius M, Stolten D. Impact of different time series aggregation methods on optimal energy system design. Renew Energy 2018;117:474–87.
[15] Pfenninger S. 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. Appl Energy 2017;197:1–13.
[16] Akter H, Howlader HOR, Saber AY, Mandal P, Takahashi H, Senjyu T. Optimal sizing of hybrid microgrid in a remote island considering advanced direct load control for demand response and low carbon emission. Energies 2021;14(22):7599.
[17] Poncelet K, Hoschle H, Delarue E, Virag A, D'haeseleer W. Selecting representative days for capturing the implications of integrating intermittent renewables in generation expansion planning problems. IEEE Trans Power Syst 2017;32(3):1936–48.
[18] Yeganefar A, Amin-Naseri MR, Sheikh-El-Eslami MK. 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. Appl Energy 2020;272:115224.
[19] Du P, Li F, Shao J. Multi-agent reinforcement learning clustering algorithm based on silhouette coefficient. Neurocomputing 2024;596:127901.
[20] de Paula AN, de Oliveira EJ. m-ISODATA: unsupervised clustering algorithm to capture representative scenarios in power systems. International Transactions on Electrical Energy Systems 2021;31(9):1–23.
[21] Höppner F, Klawonn F, Kruse R, Thomas R. "Fuzzy cluster analysis: methods for classification, data analysis and image recognition". Chichester, U.K.: Wiley; 1990.
[22] Li C, Conejo AJ, Liu P, Omell BP, Siirola JD. Mixed-integer linear programming models and algorithms for generation and transmission expansion planning of power systems. Eur J Oper Res 2022;297:1071–82.
[23] García-Cerezo A, Baringo L, García-Bertrand R. Representative days for expansion decisions in power systems. Energies 2020;13(2):335.
[24] Munoz FD, Mills AD. Endogenous assessment of the capacity value of solar PV in generation investment planning studies. IEEE Trans Sustain Energy 2015;6(4):1574–85.
[25] Kaufman L, Rousseeuw PJ. Finding groups in data: an introduction to cluster analysis. Chichester, U.K.: Wiley; 1990.
[26] Scott IJ, Carvalho PMS, Botterud A, Silva CA. Clustering representative days for power systems generation expansion planning: capturing the effects of variable renewables and energy storage. Appl Energy 2019;253:113603.
[27] Bahl B, Söhler T, Hennen M, Bardow A. Typical periods for two-stage synthesis by time-series aggregation with bounded error in objective function. Front Energy Res 2018;5:35.
[28] Maiz S, Baringo L, García-Bertrand R. Dynamic expansion planning of a commercial virtual power plant through coalition with distributed energy resources considering rival competitors. Appl Energy 2025;377:124665.
[29] Arnold F, Lilienkamp A, Namockel N. Diffusion of electric vehicles and their flexibility potential for smoothing residual demand—a spatio-temporal analysis for Germany. Energy 2024;308:132619.
[30] Anderson O, Yu N, Oikonomou K, Maloney P, Wu D. Representative Period selection for robust capacity expansion planning in low-carbon grids. In: 2024 IEEE/PES transmission and distribution conference and exposition (T&D). IEEE; 2024 May 6. p. 1–5.
[31] Li Z, Xia * Y, Bo Y, Wei W. Optimal planning for electricity-hydrogen integrated energy system considering multiple timescale operations and representative time-period selection. Appl Energy 2024;362:122965.
[32] Theodorakos K, Agudelo OM, Becker T, Vanthournout K, D'hulst R, De Moor B. Explainable representative-days clustering on low-voltage grid meters and feeders, with noise-aware multi-objective Bayesian optimization, applied to grid-congestion events. Sustainable Energy, Grids and Networks 2025;41:101622.
[33] Ward Jr JH. Hierarchical grouping to optimize an objective function. J Am Stat Assoc Mar. 1963;58(301):236–44.
[34] García-Cerezo A, García-Bertrand R, Baringo L. Enhanced representative time periods for transmission expansion planning problems. IEEE Trans Power Syst

2021;36(4):3802–5.
[35] Marcy C, Goforth T, Nock D, Brown M. Comparison of temporal resolution selection approaches in energy systems models. Energy 2022;251:123969.
[36] Dvorkin Y, Fernandez-Blanco R, Kirschen DS, Pandžić H, Watson JP, Silva-Monroy CA. Ensuring profitability of energy storage. IEEE Trans Power Syst 2017; 32(1):611–23.
[37] Sisternes FJD, Jenkins JD, Botterud A. The value of energy storage in decarbonizing the electricity sector. Appl Energy 2016;175:368–79.
[38] Almaimouni A, Ademola-Idowu A, Kutz JN, Negash A, Kirschen D. Selecting and evaluating representative days for generation expansion planning. In: Proceedings of the 2018 power systems computation conference (PSCC), Dublin, Ireland; June 2018. p. 11–5.
[39] Liu F, Deng Y. Determine the number of unknown targets in open world based on elbow method. IEEE Trans Fuzzy Syst 2021;29(5):986–95.
[40] Teichgraeber H, Brodrick PG, Brandt AR. Optimal design and operations of a flexible oxyfuel natural gas plant. Energy 2017;141:506–18.
[41] Schütz T, Schraven MH, Fuchs M, Remmen P, Müller D. Comparison of clustering algorithms for the selection of typical demand days for energy system synthesis. Renew Energy 2018;129:570–82.