Multiple Seasonal STL decomposition with discrete-interval moving seasonalities

Oscar Trull [a], J. Carlos García-Díaz [b], A. Peiró-Signes [c]

[a] Dep. Of Statistics, Universitat Politècnica de València, Valencia, Spain, otrull@eio.upv.es

[b] Dep. Of Statistics, Universitat Politècnica de València, Valencia, Spain, juagardi@eio.upv.es

[c] Management Department, Universitat Politècnica de València, Valencia, Spain, anpeisig@omp.upv.es

Abstract

The decomposition of a time series into components is an exceptionally useful tool for understanding the behaviour of the series. The decomposition makes it possible to distinguish the long-term and the short-term behaviour through the trend component and the seasonality component. Among the decomposition methods, the STL (Seasonal Trend decomposition based on Loess) method stands out for its versatility and robustness. This method, however, has one main drawback: it works with a single seasonality, and does not deal with the calendar effect. In this article we present a new decomposition method, based on the STL, which allows the use of different seasonalities while allowing the calendar effect and special events to be introduced into the model using discrete-interval moving seasonalities (MSTL-DIMS). To show the improvements obtained, the MSTL-DIMS technique is applied to short-term load forecasting in some electricity systems, and the results are discussed.

Keywords

MSTL; DIMS; decomposition; Loess; multiple seasonal

1. Introduction

This article presents a new decomposition algorithm based on seasonal trend decomposition based on loess (STL), in which an internal decomposition of multiple seasonalities is performed as part of the decomposition process itself. In addition, this algorithm allows the inclusion of discrete-interval moving seasonalities (DIMS), so that special events in the time series can also be decomposed. With this new algorithm, forecasters using methods based on this decomposition can obtain more accurate seasonal and trend components, as well as relevant seasonal components for special events. To demonstrate the improvement obtained, this new algorithm is applied to five series of hourly electricity demand in Europe, and the results are analysed. The article describes the new method, analyses the use of the parameters associated with the algorithm, and performs an analysis on the components obtained.

This article is divided into the following sections: Section 2 introduces the time series decomposition; Section 3 describes the methodology of the algorithm and the default parameters assigned; Section 4 describes the data used in this article; Section 5 shows the results of an application of the methodology to electricity demand series; and, finally, Section 6 discusses the results and lists the conclusions and further research.

2. Time Series Decomposition

The decomposition of a time series allows us to understand the behaviour of the series in a visual and simple way [1]. Understanding that the long-term behaviour of a series is predictable if the trend is known, or determining the shorter-term behaviour once the seasonal component is known, allows the forecaster to envisage their predictions. Within this area, the STL decomposition [2] stands out as the most commonly used method. The STL method is based on performing weighted local regressions (loess) on the seasonal indices and on the trend, and is an evolution of the SABL (Seasonal Adjustment at Bell Laboratories) method, which was published years earlier [3]. Other decomposition methods have been proposed, mainly based on ARIMA models, such as X11, X12 and X13 algorithms [4,5] and TRAMO/SEATS [6]. These methods do not have the versatility of the STL, which is completely configurable, but have had wide application in economic studies. A newer trend is to try to include the frequency domain, or even the time-frequency domain, in decomposition based on wavelets [7,8] or other alternatives including entropic segmentation algorithms [9]. The algorithm was originally implemented in FORTRAN and has not undergone major changes since then, because it is very reliable and robust. Some new applications have emerged that seek to select the parameters automatically and more efficiently [10]. This library has also been adapted to work with multiple seasonalities [11]. The mstl function with the R forecast library has recently been published [12]. This algorithm uses an iterative method to obtain the seasonal components one by one, and finally to obtain the trend.

The effect of special events can be noticeable in some series. In general, this effect is associated with the calendar, and occurs irregularly. The previous methods cannot deal with special events. The difficulty of dealing with special events means that this calculation is sometimes omitted, or that dummy variables are used to manage them [13]. Some proposed models handle special events and series irregularities outside of the decomposition itself, using algorithms that include ARIMA models [14,15].

Trull et al. [16,17] proposed the use of discrete-interval moving seasonalities (DIMS) to consider the calendar effect within a multiple seasonal Holt–Winters model, as part of the model itself. The results showed that this implementation outperformed previous methods, although the new model follows the guidelines of the original model. In this work, we implement an STL decomposition with the incorporation of DIMS as a fundamental part of the model to deal with special events (MSTL-DIMS). To check the effectiveness of the new decomposition method, we apply it to different time series of hourly electricity demand, proving that the method improves those previously described.

3. MSTL-DIMS decomposition

The STL decomposition model is traditionally expressed in its additive form as in (**1**).

$$Y_t = T_t + S_t + R_t \tag{1}$$

where Y_t are the observed values, and T_t stands for the trend component, S_t the seasonal component and R_t the remainder. This decomposition is based on separating local regressions using loess at the seasonal cycle level, and then smoothing and subtracting them from the series to obtain the trend, which is also done through loess regressions and smoothing. Loess uses the bicubic formulation shown in (**2**) as the weight function for the ith observation.

$$v_i(t) = \begin{cases} \left[1 - \left(\frac{|t_i - t|}{\delta_\gamma(t)}\right)^3\right]^3 & \text{for } 0 \leq \frac{|t_i - t|}{\delta_\gamma(t)} \leq 1 \\ 0 & \text{for } \frac{|t_i - t|}{\delta_\gamma(t)} \geq 1 \end{cases} \tag{2}$$

with $\delta_\gamma(t) = |t_i - t_\gamma|$ standing for the distance between the farthest t_i and the current t.

As mentioned above, S_t can only address one seasonality in the original model. We propose a new model, in which the multiple seasonalities are obtained simultaneously and DIMS is also applied, resulting in the model shown in (**3**).

$$Y_t = T_t + \sum_{i=1,\dots,n_{seas}} S_t^{(i)} + \sum_{j=1,..,n_{DIMS}} D_{t^*}^{(j)} + R_t \tag{3}$$

Here, the former S_t has been split into n_{seas} seasonalities defined each one as $S_t^{(i)}$. The new term includes n_{DIMS} discrete seasonalities, defined as $D_{t^*}^{(j)}$. Notice that DIMS are defined only at time t^*. The number of seasonalities and DIMS considered depends on the dataset and the forecaster.

To support the following explanations, the hourly electricity demand series in Spain provided by the European association for the cooperation of transmission system operators (TSOs) for electricity (ENTSO-E) through its website (https://www.entsoe.eu/) is used. This series, shown in Figure 1, contains the hourly demand data for the Iberian Peninsula (Spanish pole) from January 2006 to the end of 2019. As can be seen from the figure, the series has daily seasonality and weekly seasonality. In addition, the effect of holidays is remarkable.

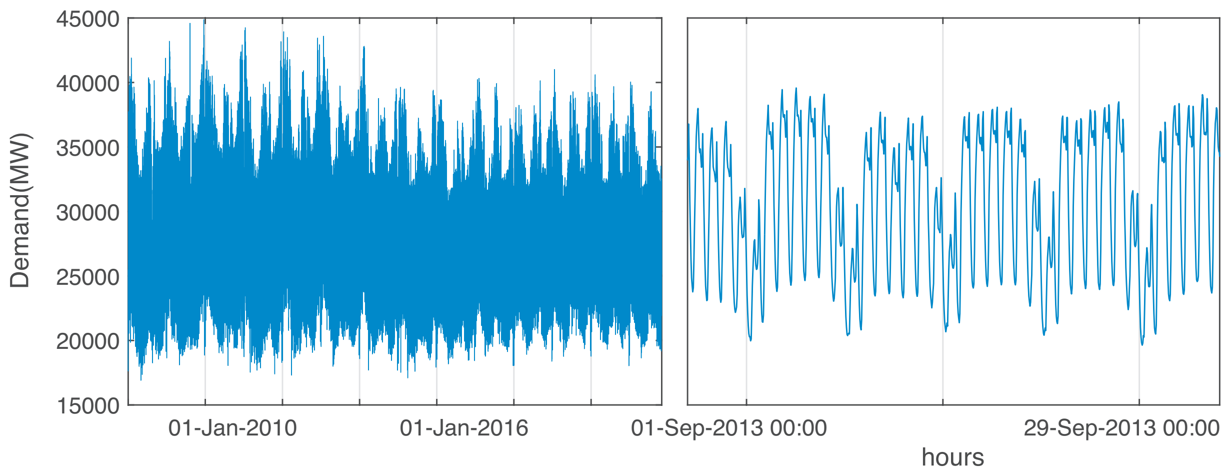


Figure 1. Hourly electricity demand in Spain.

The algorithm to obtain the components consists of two nested iterative processes, such that an inner loop obtains the seasonality and trend components, while the outer loop provides greater robustness to the algorithm, allowing the effects caused by outliers and other dissonant elements to be eliminated. Unlike the original algorithm, the algorithm works with different seasonalities simultaneously within the inner loop. Figure 2 reflects the working scheme of the algorithm. The process for obtaining the final components (seasonalities, trend and DIMS) from the time series is described below:

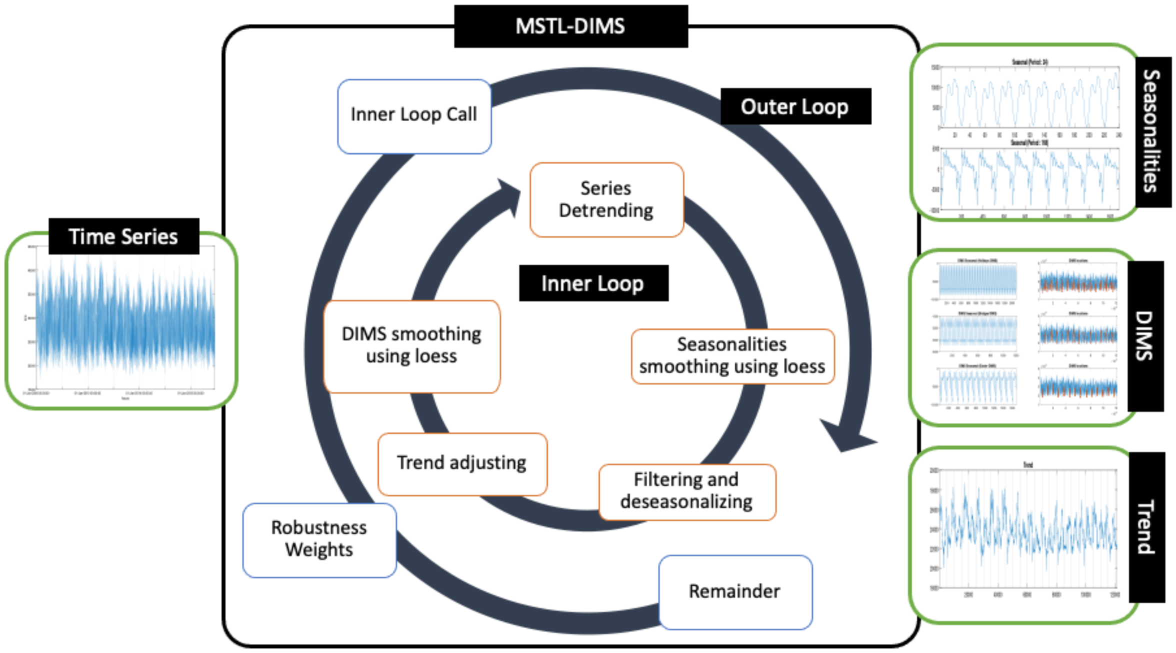


Figure 2. Description of the algorithm.

3.1. Inner loop

The inner loop is responsible for obtaining the components, including multiple seasonalities and DIMS. Each iteration k computes the components following the six steps set out below.

In step 1, a detrended series is obtained, resulting from the subtraction of the calculated trend from the working series, as indicated in (**4**). This sequence starts from a null trend ($T_t^0 = \mathbf{0}$), which is recalculated as the loop repeats.

$$DY_t^{(k)} \equiv Y_t - T_t^{(k-1)} = \sum_{i=1,\ldots,n_{seas}} S_t^{(i,k)} + \sum_{j=1,\ldots,n_{DIMS}} D_t^{(j,k)} + R_t^{(k)} \qquad \mathbf{(4)}$$

where $DY_t^{(k)}$ represents the detrended series at iteration k.

In step 2, the cycle components $C_t^{(i,k)}$ are obtained from smoothing the cycle subseries. At each instant within the seasonality and throughout the time series, loess smoothing is performed. In this way, a smoothed regression of the values for the same instant of seasonality in the entire series is carried out cross-sectionally.

Figure 3 shows how this process is carried out on a time series with a seasonality of 24 hours. The series is represented in a cycle subseries mode of 24 hours, and, for example, for hour 19 (represented with the black line) all the values are taken, and loess is applied. In this way, the seasonal value for hour 19 is obtained.

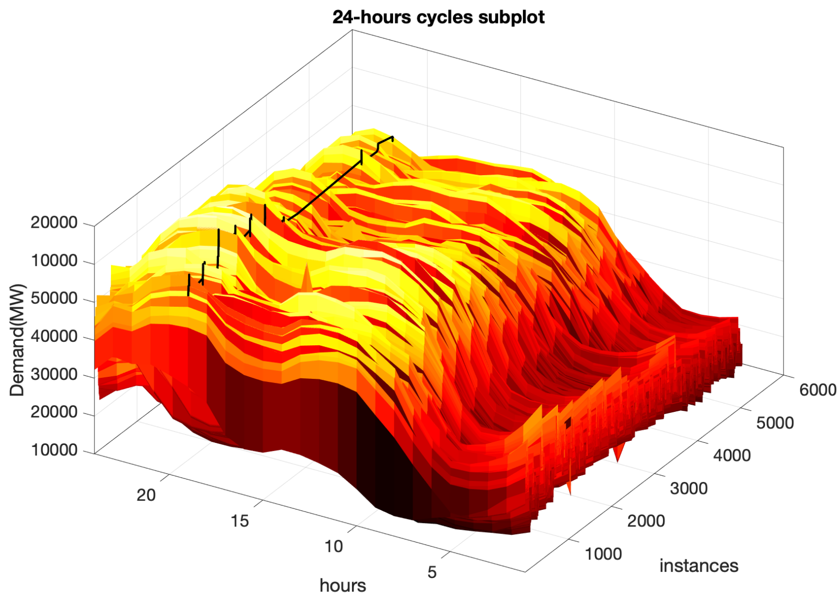


Figure 3. Seasonal smoothing for a 24-hour cycle in a time series. The surface graph represents the 24-hour cycle subseries. The black line at the top represents the seasonal smoothing for hour 19.

The weight assigned during smoothing is determined according to (**5**).

$$v_{i,R}(t) = \rho_v(t) v_i(t) \quad \textbf{(5)}$$

where $v_{i,R}(t)$ is the robust neighbourhood weight at time t, and $\rho_v(t)$ is the robustness weight. Two options are allowed at this point. (1) Once one $C_t^{(i,k)}$ is calculated, its values are removed from the detrended $DY_t^{(k)}$ and the process continues until it has been repeated n_s times. (2) Alternatively, each $C_t^{(i,k)}$ is obtained separately from $DY_t^{(k)}$, without taking into account any other seasonality.

In step 3, a low-pass filter is applied to all the cycle components $C_t^{(i,k)}$ to eliminate the noise in the cycles. Each filter is the result of two applications of moving averages of length the cycle length, and a final moving average of length 3. The process finished with loess smoothing of all cycles and the variable $L_t^{(i,k)}$ is obtained.

In step 4, the possible trend present on the seasonalities is removed. Following (**6**), the smoothed filtered cycle component is removed from the original cycle.

$$S_t^{(i,k)} = C_t^{(i,k)} - L_t^{(i,k)} \tag{6}$$

At this point, depending on the extraction method chosen in step 2, additional actions must be carried out. If case (2) was selected, the seasonalities duplicate their common effect, so a sequential substraction starting from the second seasonality must be carried out, as shown in (**7**).

$$S_t^{(i,k)} = \begin{array}{ll} S_t^{(i,k)} & i = 1 \\ S_t^{(i,k)} - \sum_{j<i} S_t^{(j,k)} & otherwise \end{array} \tag{7}$$

Step 5 removes the seasonalities from the original time series. This step finishes with trend smoothing by loess, and the remainder is obtained. Equation (**8**) shows how to obtain both these components.

$$T_t^{(k)} + R_t^{(k)} = Y_t - \sum_{i=1}^{n_{seas}} S_t^{(i,k)} \tag{8}$$

Step 6: In order to obtain the DIMS, there is an additional step in which the series is seasonally adjusted without the trend. Figure 4 shows an example DIMS, where the holidays of the time series are represented. The smoothed value of the 24-hour seasonality is shown in red, and is clearly higher than that of the DIMS.

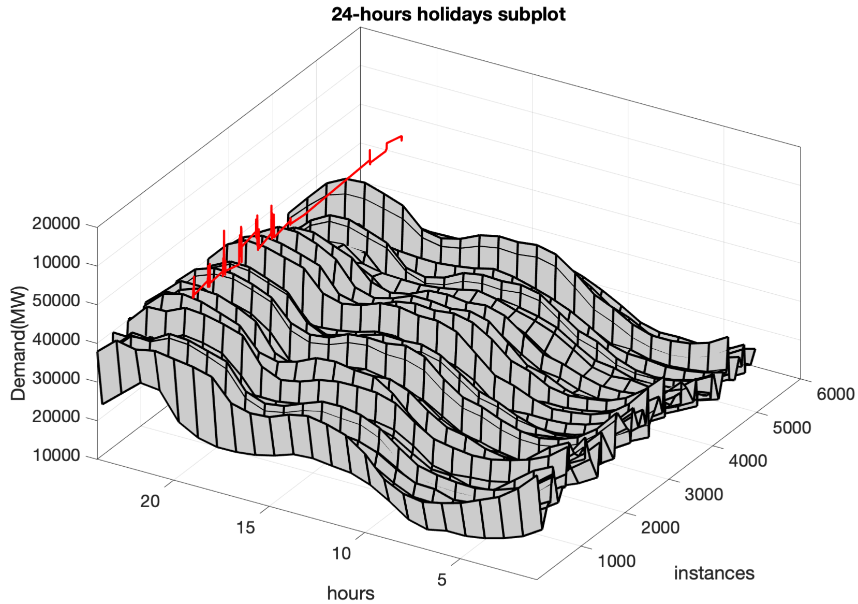


Figure 4. DIMS for the special event 'holiday'. The surface graph represents the cycle subseries, while the red line represents the value for hour 19 of the 24-hour seasonality.

Obtaining the components related to the DIMS can be done at the time the rest of the seasonalities are obtained, but it was finally decided to make the adjustment in this step. The fact that the appearance of the DIMS is sporadic and random and depends on the seasonality in which it is embedded forces its calculation to be carried out at this point and not during the calculation of the regular seasonalities. The variability thus achieved is greatly reduced. Thus, and after testing the two options with different series, the best results have been obtained by calculating the DIMS in step 6.

Once the DIMS of the series have been isolated, as can be seen in Figure 4, loess is performed in the same way as for the seasonalities to determine its value. One point to take into account is the simultaneity of two DIMS. It may be the case that, for example, two special events coincide on the same date, which would mean an overlap of the two. In such a case, the forecaster must select only one of them, taking it as a reference for that date. This can be extrapolated to other special events. Regarding recursion in these cases, the forecaster must select whether to decide for one or the other event, since they cannot be related to two previous ones.

Once this step is finished, the series is reconstructed to perform the loop again, although for step 1 the new trend obtained is used.

3.2. Outer loop

One of the advantages of using STL for decomposition is the robustness of the method. This robustness is based on the use of a second external loop that tries to eliminate possible aberrations in the components obtained. To do this, a robust weights function is used as indicated in (**9**).

$$\rho_v(t) = \begin{cases} \left[1 - \left(\dfrac{|R_t|}{|6 \cdot \text{median}(|\{R_t\}_{t=1}^{N}|)|}\right)^2\right]^2 & \text{for } |R_t| < |6 \cdot \text{median}(|\{R_t\}_{t=1}^{N}|)| \\ 0 & \text{else} \end{cases} \quad \textbf{(9)}$$

The remainder R_t is calculated by subtracting, from the original series, the sum of all the extracted components.

3.3. Parameters

The algorithm allows the customization of all parameters. However, if no specification is indicated, the algorithm uses the default parameters, which are the result of the analyses carried out on the series that have been worked on.

The parameter $n_{(s)}$ for seasonal smoothing is possibly the most important. It is necessary to indicate the value for each seasonal component and DIMS; if not, the value provided is used for all of them. If no value is entered, the algorithm uses a default set, as explained in Section 5.

The calculation of the window for trend $n_{(t)}$ smoothing is performed according to (**10**).

$$n_{(t)} \geq \frac{1.5 \max(s_I)}{1 - 1.5\, n_{(s),I}}, \text{with } I = \max(s_i) \quad \textbf{(10)}$$

where s_i are the lengths of the periods of all the seasonalities, and $n_{(s),i}$ are the values of $n_{(s)}$ for the seasonality i selected.

4. Dataset

To check the effectiveness of the proposed method, an analysis was carried out on five electricity demand series from different countries (Spain, Italy, Germany, France and the United Kingdom). The dataset was obtained from [18], and comprises hourly aggregated electricity demand from certain European countries from 2006 to 2015. All the series share the same characteristics: strong seasonality at 24 and 168 hours; holidays that are not repeated on the same dates, including Easter; and little variability of the time series.

5. Analysis and results

The introduction of more than one seasonality requires a study on the parameter $n_{(s)}$, since it becomes the most important parameter. The original STL described in [2] uses as a default $n_{(s)} = \mathrm{nextodd}(10 \cdot N)$, where N is the number of observations in the time series. The function *mstl* in R uses a sequential number $n_{(s)} = [7,11,15,\dots]$.

In our case we prepared a set of possible $n_{(s)}$ based on previous works, on the one hand, determining the value that could be used in a general way in all seasons, and, on the other hand, using different values according to seasonality.

The method chosen to determine the effectiveness was to use the remainder to obtain the Root Mean Square Error (RMSE), described in (**11**). Here N stands for the number of observations of the time series, x_n are the individual observations and $\bar{x}$ is the mean of all the observations.

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})^2} \tag{11}$$

This indicator is usually used for the adjustment of smoothing models. The remainder measures how much of the series has not been interpreted by the decomposition. Table 1 shows the most notable results. Although a large number of options were tested, we want to highlight the ones shown in the table. There are two methods of obtaining the seasonalities (as discussed in Section 2). In addition, the option of using the robust method is included, to observe its behaviour.

Id	$\boldsymbol{n_{(s)}}$	**Sequential**	**Independent**	**Seq., Robust**	**Ind., Robust**
1	$\mathrm{nextodd}(10 \cdot N)$, for all	1724	1724	1724	1724
2	$\mathrm{nextodd}(5 \cdot N)$, for all	1724	1724	1724	1724
3	$\mathrm{nextodd}(N)$, for all	1724	1724	1724	1724
4	$\mathrm{nextodd}(0.5 \cdot N)$, for all	1724	1724	1724	1724
6	$\mathrm{nextodd}(0.1 \cdot N)$, for all	1705	1724	1704	1724
7	$\mathrm{nextodd}(10 \cdot s_{168})$for all	1462	1699	1460	1699

8	$\text{nextodd}(10 \cdot s_{24})$, for all	1181	1422	1177	1420
9	13, for all	**1032**	**967**	**1007**	**965**
10	15, for all	**936**	**1028**	**936**	**1026**
11	[13;15;17;21;25]	1086	1028	1058	1026
12	[13;17;21;25;29]	1081	1026	1054	1024
13	[11;17;23;29;35]	1081	1026	1054	1024
14	[11;81;55;11;11]	1126	1340	1104	1337
15	[11;755;755;755;755]	1155	1604	1136	1603
16	[11;15755;15575;15575;15575]	1178	1724	1160	1724

Table 1. RMSE of the residue using different $n_{(s)}$, according to the method used to obtain the seasonality and the robustness.

From the results it can be seen how the initial proposal of using $\text{nextodd}(10 \cdot N)$ is not the best. In fact, as we reduce its value, it remains unchanged (cases 1-6) up to a threshold, from which its efficiency begins to improve. We tried using the same formula but limiting it to seasonality (cases 7-8). Here an improvement is observed, which is linked to a reduction in the length of $n_{(s)}$. So we decided to use small values for all seasonalities (cases 9-10), which turned out to be the most effective. Tests with different lengths of $n_{(s)}$ were also performed: cases 11-13, in which a sequence of values was taken into account, following the indications of [19], and later approximations to the different cases studied (cases 14-16). From the results, we concluded that, for this application, the best results are obtained using the same $n_{(s)}$ for all seasonalities, of length 15.

4.1. Benchmark

The initial proposed decomposition to benchmark the proposed method is shown in (**12**). Here we removed the special events, as the former methods cannot deal with them, so thus the comparison can be more realistic.

$$Y_t = T_t + S_t^{24h} + S_t^{168h} + R_t \quad \textbf{(12)}$$

As a reference to compare the behaviour of the algorithm, the result obtained is compared with those provided by the other methods previously described in section 2. Table 2 shows the list of alternatives analysed in the study.

Id	Description
(a)	STL decomposition using one Seasonal component of 24 hours length.
(b)	STL decomposition with one seasonal component of 168 hours length.
(c)	MSTL with two seasonal components (24 and 168 hours), using library 'forecast' in R.
(d)	STLplus decomposition using one Seasonal component of 24 hours length.
(e)	STLplus decomposition with two seasonal components (24 and 168 hours).
mstl	Proposed MSTL method, with two seasonal components (24 and 168 hours).

Table 2. List of methods used to compare the performance of the new proposal.

For each time series, the decomposition described in the method is applied, with the series filtered to avoid missing data, as some of the methods in Table 2 do not deal with missing values. The RMSE was again applied as the benchmark indicator. Table 3 shows the results.

	(a)	(b)	(c)	(d)	(e)	mstl
Spain	1701	1753	962	1613	1726	934
Italy	2626	2864	1644	2433	2804	1421
Germany	3057	3334	2030	2817	3266	1907
France	2589	2888	1731	2470	2799	1552
UK	3283	3725	1260	3034	3683	1203

Table 3. Comparison of different STL methods. Values are RMSE.

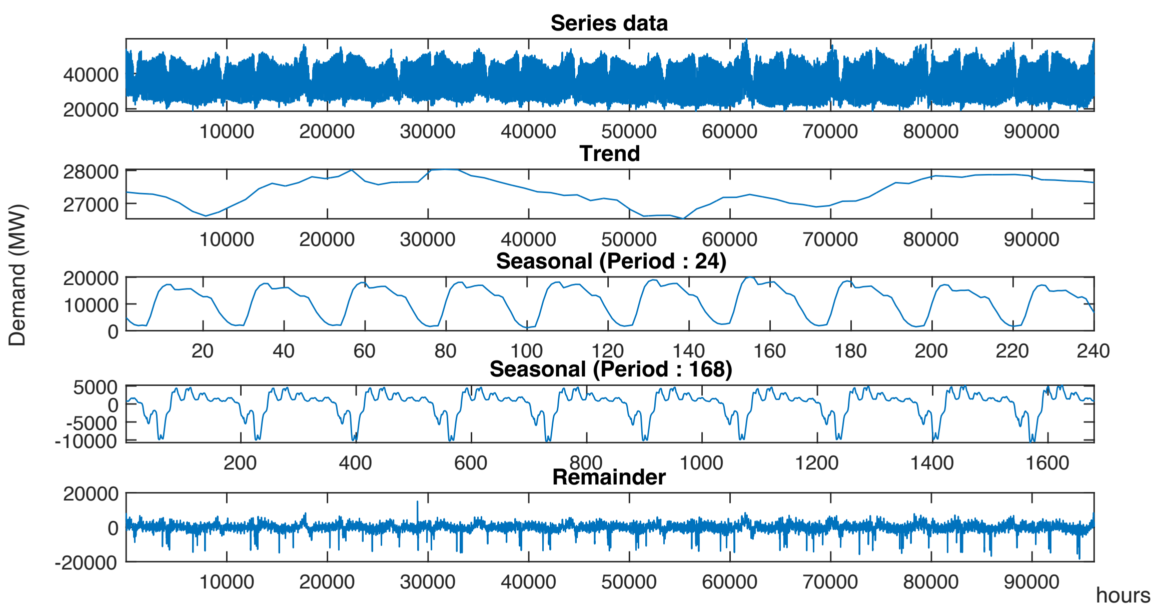
The results show how the multiple seasonal methods (cases c and mstl) substantially improve on the results obtained from the other decompositions. The difference is around a 50% improvement. Among these two selected cases, it is also the new proposal that provides the best results, thus confirming the suitability of the new algorithm.

4.2. Improvement by including DIMS

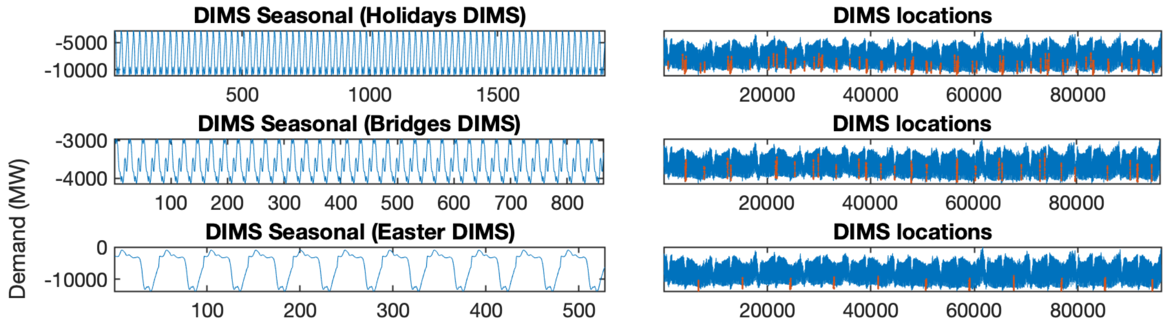
Once it had been observed that the new methodology was comparable with the best of the existing methods, the DIMS were introduced in the model. In this case, we decided to introduce holidays and bridge days associated with holidays. Easter was also included as a separate element. The decomposition is shown in (**13**).

$$Y_t = T_t + S_t^{24h} + S_t^{168h} + D_t^{Easter} + D_t^{holidays} + D_t^{bridges} + R_t \quad \textbf{(13)}$$

The graphical result of the decomposition is shown in Figure 5, which is the result of the output of the application developed in MATLAB(R) to carry out this decomposition. Figure 5(a) shows how the series breaks down into its regular seasonalities, trend, and the remainder. Figure 5(b) shows the discrete seasonalities of the DIMS, as well as their position in the time series.



(a)



(b)

Figure 5. MSTL decomposition with DIMS. (a) Original series, regular seasonalities, trend and remainder. (b) DIMS seasonal indices (left) and locations in the original series (right).

The introduction of DIMS makes it possible to model the irregularities caused by special events, so that the results improve. The results for each of the series studied are shown in Table 4. The demand series for each country are ordered in the columns. The rows start with the original series, and in the next row the holidays and long weekends are added. Row 3 includes, in addition to holidays and long weekends, Easter. In row 5, the original model returns, but distinguishing the third seasonality, discretely (years and leap years). In the last row all the DIMS described are included.

	Spain	**Italy**	**Germany**	**France**	**UK**
Original	934	1421	1907	1552	1203
+ Holidays	764	1085	1555	1345	1128
+ Holidays+Easter	725	1021	1418	1309	1118

Table 4. RMSE for the selected model per country, including DIMS.

In all cases, it is observed how the introduction of DIMS produces an improvement in the RMSE of the remainder. This is because the algorithm correctly captures the discrete seasonalities contained in the DIMS. The result is a more efficient decomposition that is better able to interpret the characteristics of the series.

Finally, we proceeded to analyse the behaviour of the decomposition measures for the series. In this way, the previous results can be compared. For this, strength measures were used. The measurement of the trend is shown in (**14**), while that of the seasonalities is described in (**15**).

$$1 - \frac{\mathrm{Var}(R_t)}{\mathrm{Var}(T_t + R_t)} \tag{14}$$

$$1 - \frac{R_t}{Var\left(S_t^{(i)} + R_t\right)} \tag{15}$$

If the introduction of DIMS improves the decomposition, the strength of the trend and seasonality components should be improved, since we are removing from the remainder a component that we can now model. The results are shown in Figure 6.

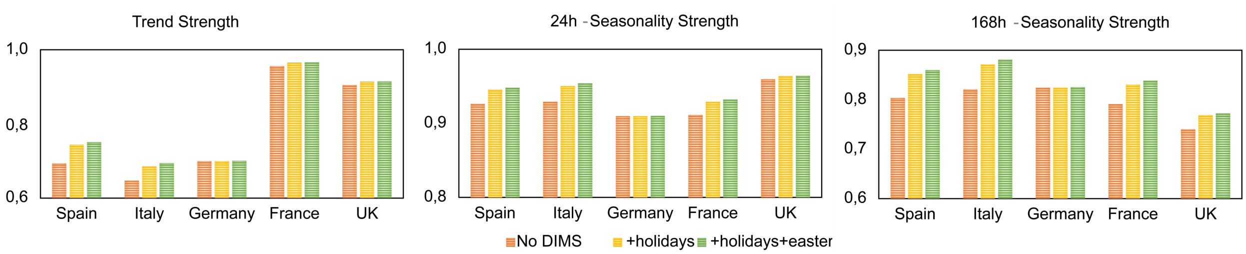


Figure 6. Strength to trend, 24h- and 168h-seasonality.

In the figure, the model presented as No DIMS is the decomposition model with seasonalities of 24 and 168 hours, which is used as a base. The +holidays model includes DIMS with information on holidays and long weekends. Finally, the +Easter model also includes the Easter DIMS.

It is observed that the strength increases in all cases, corroborating the situation seen in Table 4.

6. Discussion and Conclusion

In this article, a new calculation algorithm for the decomposition of time series based on the STL decomposition is presented. This new method focuses on multiple seasonal decomposition and allows the inclusion of special events such as the calendar effect, through the use of discrete seasonalities (DIMS). In order to verify the proper functioning of the algorithm, it is applied to five series of hourly electricity demand from five European countries.

To check the effectiveness of the new decomposition method, the results are compared with other methods already available today, comparing the remainder that the algorithm cannot model. The results obtained show that this new algorithm improves the previously obtained results.

Comparisons have been made using the RMSE as an indicator. The results obtained show that the new algorithm substantially improves the methods used with a single seasonality, reducing the RMSE by almost 50%. In addition, it improves the existing models with various seasonalities, sometimes by up to 10%. Moreover, the introduction of the DIMS makes it possible to improve these results further (an improvement of up to 25% depending on the series used) and to obtain a seasonal decomposition for special events, such as the calendar effect.

This contribution allows the simple use of a time series decomposition methodology based on STL that includes special events and the calendar effect. Although in the article the proofs have been based on electricity demand series, these conclusions can be extrapolated to many other types of time series. Future lines of research will focus on determining the types of DIMS to use, as well as automatic detection.

7. REFERENCES

[1] R.P. Hafen, D.E. Anderson, W.S. Cleveland, R. MacIejewski, D.S. Ebert, A. Abusalah, M. Yakout, M. Ouzzani, S.J. Grannis, Syndromic surveillance: STL for modeling, visualizing, and monitoring disease counts, BMC Med. Inform. Decis. Mak. 9 (2009) 1–11. https://doi.org/10.1186/1472-6947-9-21/FIGURES/11.

[2] R.B. Cleveland, W.S. Cleveland, J.E. McRae, I. Terpenning, STL: A seasonal-trend decomposition procedure based on loess, J. Off. Stat. 6 (1990) 3–73.

[3] W.S. Cleveland, D.M. Dunn, I.J. Terpenning, SABL: A resistant seasonal adjustment procedure with graphical methods for interpretation and diagnosis, in: Seas. Anal. Econ. Time Ser., NBER, 1978: pp. 201–241.

[4] E.B. Dagum, G. Huot, M. Morry, Seasonal adjustment in the eighties: Some problems and solutions, Can. J. Stat. 16 (1988) 109–126. https://doi.org/https://doi.org/10.2307/3315220.

[5] E.B. Dagum, S. Bianconcini, Seasonal adjustment methods and real time trend-cycle estimation, Springer, 2016.

[6] V. Gómez, A. Maravall, Programs TRAMO(Time Series Regression with ARIMA Noise, Missing Observations and Outliers) and SEATS (Signal Extractions ARIMA Time Series). Instructions for the user

(beta version: junio 1997), 1997. http://www.bde.es (accessed September 10, 2018).

[7] J. Wang, Z. Wang, J. Li, J. Wu, Multilevel wavelet decomposition network for interpretable time series analysis, Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (2018) 2437–2446. https://doi.org/10.1145/3219819.3220060.

[8] A. Ben Mabrouk, N. Ben Abdallah, Z. Dhifaoui, Wavelet decomposition and autoregressive model for time series prediction, Appl. Math. Comput. 199 (2008) 334–340. https://doi.org/10.1016/j.amc.2007.09.067.

[9] Y. Yin, P. Shang, J. Xia, Compositional segmentation of time series in the financial markets, Appl. Math. Comput. 268 (2015) 399–412. https://doi.org/10.1016/j.amc.2015.06.061.

[10] R. Hafen, stlplus: Enhanced Seasonal Decomposition of Time Series by Loess, Compr. R Arch. Netw. (2016).

[11] J. Arnerić, Multiple STL decomposition in discovering a multi-seasonality of intraday trading volume, Croat. Oper. Res. Rev. 12 (2021) 61–74. https://doi.org/10.17535/crorr.2021.0006.

[12] R. Hyndman, G. Athanasopoulos, C. Bergmeir, G. Caceres, L. Chhay, M. O'Hara-Wild, F. Petropoulos, S. Razbash, E. Wang, F. Yasmeen, forecast: Forecasting functions for time series and linear models, R Packag. Version 8.16. (2022). https://pkg.robjhyndman.com/forecast/ (accessed April 3, 2022).

[13] R. Weron, Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach, John Wiley & Sons, Ltd, Chichester, England, 2006.

[14] D. Ollech, Seasonal Adjustment of Daily Time Series, J. Time Ser. Econom. 13 (2021) 235–264. https://doi.org/doi:10.1515/jtse-2020-0028.

[15] E. Caro, J. Juan, J. Cara, Periodically correlated models for short-term electricity load forecasting, Appl. Math. Comput. 364 (2020) 124642. https://doi.org/10.1016/j.amc.2019.124642.

[16] O. Trull, J.C. García-Díaz, A. Troncoso, Application of Discrete-Interval Moving Seasonalities to Spanish Electricity Demand Forecasting during Easter, Energies. 12 (2019) 1083. https://doi.org/10.3390/en12061083.

[17] O. Trull, J.C. García-Díaz, A. Troncoso, One-day-ahead electricity demand forecasting in holidays using discrete-interval moving seasonalities, Energy. 231 (2021) 120966. https://doi.org/10.1016/j.energy.2021.120966.

[18] Power Statistics, (n.d.). https://www.entsoe.eu/data/power-stats/ (accessed April 3, 2022).

[19] R. Hyndman, G. Athanasopoulos, C. Bergmeir, G. Caceres, L. Chhay, M. O’Hara-Wild, F. Petropoulos, S. Razbash, E. Wang, F. Yasmeen, forecast : Forecasting functions for time series and linear models, R Packag. Version. (2022). https://pkg.robjhyndman.com/forecast/.