CLIMAX

1. OD and ODS optimization

1.1. Approach overview

The codes **optimOD.py** and **optimODS.py** provide optimal sub-regional shares, within a target region, for solar and wind power (SP and WP, respectively). That is, an Optimal spatial Distribution (thereby the acronym OD). The sub-regions should be predefined in advance, and may not be the same for SP and WP. So we will have N_S sub-regions for the allocation of SP capacity, and N_W sub-regions for the allocation of WP capacity. The code **make-regions.R** can identify them based on homogeneity criteria (read more at the end of this document), but any other classification or clusters of grid-points (in case you are using gridded datasets) can be used. In any case, input data will be referred to each sub-region at play, whatever the criteria used for their definition.

The codes work on minimizing the following function:

$$\sum_{k=0}^{NTT}\left(\sum_{i=1}^{N_S} S_{Si}A_{Sik}+\sum_{j=1}^{N_W} S_{Wj}A_{Wjk}-B_k\right)^2 \tag{1}$$

where A_{Sik}/A_{Wjk} refers to the input SP/WP capacity factor (CF) data (absolute values, anomalies...) corresponding to the sub-region i/j at time k, and B_k constitutes a reference time series. NTT is the number of time steps in the series.

The CF time series should be provided in a file as the sample file file-cf1.txt, where each row is a time step and each column a sub-region, with the first N_S columns corresponding to each sub-region for the allocation of SP plants and the next N_W columns corresponding to each sub-region for the allocation of WP plants. The reference time series should be provided in a single column file as the sample file file-ref.txt, where each row is a time step (thus NTT rows expected).

The minimization of the above function, hereafter optimization process, will provide the optimal values of S_{Si}/S_{Wj}, these being the shares of SP/WP in the sub-region i/j.

For the optimization process, we impose the following general restrictions:

I. Positive share values:

$$S_{Si} \geq 0 \;\forall\; i=1,...,N_S$$

$$S_{Wj} \geq 0 \;\forall\; j=1,...,N_W$$

II. Total share = 1:

$$\sum_{i=1}^{N_S} S_{Si} + \sum_{j=1}^{N_W} S_{Wj} = 1$$

III. Guarantee of a minimum capacity of production:

$$\sum_{i=1}^{N_S} S_{Si}C_{Sik} + \sum_{j=1}^{N_W} S_{Wj}C_{Wjk} \ \geq \ M_k \ \forall \ k = 1, ..., NMP \tag{2}$$

where C_{Sik}/C_{Wjk} refers to absolute SP/WP capacity factor (CF) data corresponding to the region i/j at time k, provided in a file as the sample file file-cf2.txt, where each row is a time step and each column a sub-region, with the first N_S columns corresponding to each sub-region for the allocation of SP plants and the next N_W columns corresponding to each sub-region for the allocation of WP plants, and M_k is user-defined (e.g. a curve of electricity production per installed watt), provided in a file as the sample file file-min.txt, with a single column and each row being a time step. The implementation of this last condition allows guaranteeing a minimum production. Note that the number of time steps in the C_S, C_W and M series (NMP) should be the same, but not necessarily equal to NTT.

Additionally, and optionally, minimum and maximum thresholds for the sub-regional shares of each energy can be defined ($S_{SCminRi}$ and $S_{SCmaxRi}$, respectively, for the values of the solar power share in the sub-region i; $S_{WCminRj}$ and $S_{WCmaxRj}$, respectively, for the values of the wind power share in the sub-region j), that is:

$$\begin{aligned} S_{SCminRi} \ &\leq \ S_{Si} \ \leq \ S_{SCmaxRi} \ \forall i = 1, ..., N_S \\ S_{WCminRj} \ &\leq \ S_{Wj} \ \leq \ S_{WCmaxRj} \ \forall j = 1, ..., N_W \end{aligned} \tag{3}$$

If a value of a minimum/maximum threshold is below/above 0/1 here, such a bottom/top limit will have no effect. This is how the above restrictions can be disabled.

1.2. Options

1.2.1. OD (optimOD.py code)

The total shares of each technology in the whole region should be kept at pre-fixed values, S_{SC} for SP and S_{WC} for WP. This adds the following conditions:

$$\begin{gathered} \sum_{i=1}^{N_S} S_{Si} = S_{SC} \\ \sum_{j=1}^{N_W} S_{Wj} = S_{WC} \end{gathered} \tag{4}$$

Since $S_{SC} + S_{WC}$ must add up to 1, the former condition II is redundant here.

1.2.2. ODS (optimODS.py code)

The total shares of each technology in the whole region come also into the optimization game. This tool thus provides the Optimal Distribution and Shares (thereby the acronym ODS). Optionally, the following additional conditions can be imposed.

First, that the total share of SP in the whole target region should be greater than the total share of WP if the mean solar CF is greater than the mean wind CF in the region, and *vice versa*. This adds the following condition:

$$\begin{aligned} \sum_{i=1}^{N_S} S_{Si} \ &\geq\ \sum_{j=1}^{N_W} S_{Wj} \quad \text{if} \quad rs2w \ > \ 1 \\ \sum_{i=1}^{N_S} S_{Si} \ &\leq\ \sum_{j=1}^{N_W} S_{Wj} \quad \text{if} \quad rs2w \ < \ 1 \end{aligned} \tag{5}$$

where $rs2w$ is the ratio between the mean SP CF averaged over the whole region and the mean WP CF averaged over the whole region. This optional condition can be overseen just by assigning a negative value to the $rs2w$ parameter in the namelistOD.txt file (see below).

Alternatively, and also optionally, minimum and maximum thresholds for the total regional shares of each energy can be defined (S_{SCmin} and S_{SCmax}, respectively, for the values of the solar power share in the whole target region; S_{WCmin} and S_{WCmax}, respectively, for the values of the wind power share in the whole target region), that is:

$$\begin{aligned} S_{SCmin} \ &\leq\ \sum_{i=1}^{N_S} S_{Si} \ \leq\ S_{SCmax} \\ S_{WCmin} \ &\leq\ \sum_{j=1}^{N_W} S_{Wj} \ \leq\ S_{WCmax} \end{aligned} \tag{6}$$

Again, if a value of a minimum/maximum threshold is below/above 0/1 here, such a bottom/top limit will have no effect. This is how the above restrictions can be disabled.

1.3. Namelist

Finally, a fourth input file is needed. It should contain the values of the parameters N_S, N_W, $rs2w$, S_{SC}, S_{WC}, S_{SCmin}, S_{WCmin}, S_{SCmax}, S_{WCmax}, $S_{SCminRi}$ $\forall i$, $S_{WCminRj}$ $\forall j$, $S_{SCmaxRi}$ $\forall i$ and $S_{WCmaxRj}$ $\forall j$ for the **optimOD.py** and **optimODS.py** codes to run (even if eventually not needed or used by the code, and thus the assigned value is obviated), as in the sample file namelistOD.txt. Importantly, when these parameters are intended to be used by the codes, they should be within the solution space of the optimization problem. In particular, be careful that S_{SC} and S_{WC} are positive values and add up to 1 (they won't be used by the ODS code, so it does not actually matter in that case), and that S_{SCmin}, S_{WCmin}, S_{SCmax}, S_{WCmax}, $S_{SCminRi}$ $\forall i$, $S_{WCminRj}$ $\forall j$, $S_{SCmaxRi}$ $\forall i$ and $S_{WCmaxRj}$ $\forall j$ are also positive values and leave sufficient space for a solution to be found if these thresholds must be taken into account during the optimization process (read above how to disable these constraints otherwise).

1.4. Implementation

The codes are implemented in phyton. For the optimization process we used the *scipi.optimize.minimize* function. The default method BFGS (Nocedal & Wright, 2006) was used. Although this package allows for the implementation of boundaries and constrains, we introduced them in a different way. First, restrictions I and II were implemented using the variables X_{Si} and X_{Wi} such that:

$$S_{Si} = \frac{X_{Si}^2}{\sum_{i=1}^{N_S} X_{Si}^2} \qquad\qquad S_{Wi} = \frac{X_{Wj}^2}{\sum_{j=1}^{N_W} X_{Wi}^2}$$

Second, the condition III was implemented adding a new term to the minimizing function:

$$D \sum_{k=1}^{NMP} f_k$$

where

$$f_k = \begin{cases} \left(\sum_{i=1}^{N_S} S_{Si} + \sum_{j=1}^{N_W} S_{Wj} - M_k \right)^2 & \text{if } \sum_{i=1}^{N_S} S_{Si} + \sum_{j=1}^{N_W} S_{Wj} < M_k \\ 0 & \text{otherwise} \end{cases}$$

If the condition is verified then $f_k = 0$ and it doesn't affect the minimization of Eq. (1). If it is not verified this new term produces, during the minimization process, a gradient towards the permitted region. The constant D controls the strength of the gradient and it is adjusted by the code in order to guarantee that deviations from the permitted region are small enough. Note that in the limit $D \to \infty$ the condition would be rigorously verified.

A similar procedure was used for the rest of the specific conditions of both, OD and ODS, codes; we added a different function for each of the conditions. In fact, one can add as many conditions as desired in this way, by modifying the codes accordingly.

After reading all the data, the code first tries to verify if all the conditions are consistent: it tries to find a set of parameter X_{Si} and X_{Wj} that verifies all the conditions. This is done by performing the minimization process including only the artificial functions constructed for each of the constrains, but not including the function of Eq. (1). In this way the minimized function (sum of all the constructed functions for each condition) should take zero value, if this is not the case we assume that the conditions should be revised and information about the conditions not verified is shown.

1.5. Application example

This is how we used the OD and ODS codes for the illustrative applications shown in the webpage and in the reference CLIMAX paper.

A_{Sik} and A_{Wjk} are the monthly anomalies of the spatially-averaged SP and WP CF series in the *i-th* sub-region for SP and the *j-th* sub-region for WP at time k. NTT is 12 times the number of years in the series.

$B_k = 0 \;\; \forall k$.

The M series comprises the mean annual cycle (at the monthly time-scale, thus $NMP = 12$) of the regional series of total CF computed by (1) spatially averaging the SP and WP CF series over the whole target region, and (2) weighthening the resulting regional SP and WP CF series by the current regional shares of SP and WP in the region before adding them up.

We disabled the restrictions related to the minimum and maximum thresholds for the sub-regional shares of each energy ($S_{SCminRi}$ and $S_{WCminRj} < 0 \;\; \forall i/j$, and $S_{SCmaxRi}$ and $S_{WCmaxRj} > 1 \;\; \forall i/j$).

Finally, for the ODS executions, we used the conditions given by the $rs2w$ parameter and thus disabled the restrictions related to the minimum and maximum thresholds for the regional shares of each energy (S_{SCmin} and $S_{WCmin} < 0$, and S_{SCmax} and $S_{WCmax} > 1$).

2. OL and OLS optimization

2.1. Approach overview

This couple of codes, **optimOL.py** and **optimOLS.py** (OL accounts for Optimal Location), are similar to the OD and ODS ones, but consider amounts of installed capacity instead of shares of each energy and so they work on minimizing the following function:

$$\sum_{k=0}^{NTT} \left(\sum_{i=1}^{N_S} I_{Si} A_{Sik} + \sum_{j=1}^{N_W} I_{Wj} A_{Wjk} - B_k \right)^2 \tag{7}$$

where A_{Sik}/A_{Wjk} refers to the input SP/WP capacity factor (CF) data (absolute values, anomalies...) corresponding to the sub-region i/j at time k, and B_k constitutes a reference time series. NTT is the number of time steps in the series.

The CF time series should be provided in a file as the sample file file-cf1.txt, where each row is a time step and each column a sub-region, with the first N_S columns corresponding to each sub-region for the allocation of SP plants and the next N_W columns corresponding to each sub-region for the allocation of WP plants. The reference time series should be provided in a single column file as the sample file file-ref.txt, where each row is a time step (thus NTT rows expected).

The minimization of the above function will provide the optimal values of I_{Si}/I_{Wj}, these being the installed capacity of SP/WP in the region i/j. Note that here I refers to installed capacity, in absolute terms, where S refers to shares in the OD and ODS codes.

For the optimization process, we impose the following general restrictions:

I. Positive values of the installed capacity in each sub-region:

$$I_{Si} \geq 0 \; \forall \; i = 1, ..., N_S$$

$$I_{Wj} \geq 0 \; \forall \; j = 1, ..., N_W$$

II. Guarantee of a minimum production:

$$\sum_{i=1}^{N_S} I_{Si} C_{Sik} + \sum_{j=1}^{N_W} I_{Wj} C_{Wjk} \;\geq\; M_k \; \forall \; k = 1, ..., NMP \tag{8}$$

where C_{Sik}/C_{Wjk} refers to absolute SP/WP capacity factor (CF) data corresponding to the region i/j at time k, provided in a file as the sample file file-cf2.txt, where each row is a time step and each column a sub-region, with the first N_S columns corresponding to each sub-region for the allocation of SP plants and the next N_W columns corresponding to each sub-region for the allocation of WP plants, and M_k is user-defined (e.g. a certain percentage of the electricity demand), provided in a file as the sample file file-min.txt, with a single column and each row being a time step. The implementation of this last condition allows guaranteeing a minimum production. Note that the number of time steps in the C_S, C_W and M series (NMP) should be the same, but not necessarily equal to NTT.

Additionally, and optionally, minimum and maximum thresholds for the values of the installed capacity of each energy in each sub-region can be defined ($I_{SCminRi}$ and $I_{SCmaxRi}$, respectively, for the values of solar power installed capacity in the sub-region i; $I_{WCminRj}$ and $I_{WCmaxRj}$, respectively, for the values of wind power installed capacity in the sub-region j), that is:

$$\begin{aligned} I_{SCminRi} \ \leq \ I_{Si} \ \leq \ I_{SCmaxRi} \ \forall i = 1, ..., N_S \\ I_{WCminRj} \ \leq \ I_{Wj} \ \leq \ I_{WCmaxRj} \ \forall j = 1, ..., N_W \end{aligned} \tag{9}$$

If a value of a minimum threshold is 0 or below 0 here, such a bottom limit will have no effect. On the other hand, huge values of the maximum thresholds will act to leave the top limits with no effect. This is how the above restrictions can be disabled.

2.2. Options

2.2.1. OL (optimOL.py code)

The total amount of installed capacity of each technology in the whole region, I_{SC} for SP and I_{WC} for WP, is fixed. This adds the following conditions:

$$\begin{aligned} \sum_{i=1}^{N_S} I_{Si} = I_{SC} \\ \sum_{j=1}^{N_W} I_{Wj} = I_{WC} \end{aligned} \tag{10}$$

2.2.2. OLS (optimOLS.py code)

The total amount of installed capacity of each technology in the whole region come also into the optimization game. Optionally, the following additional conditions can be imposed.

First, that the total amount of SP installed capacity in the whole target region should be greater than the total amount of WP installed capacity if the mean solar CF is greater than the mean wind CF in the region, and *vice versa*. This adds the following condition:

$$\begin{aligned} \sum_{i=1}^{N_S} I_{Si} \ \geq \ \sum_{j=1}^{N_W} I_{Wj} \quad \text{if} \quad rs2w \ > \ 1 \\ \sum_{i=1}^{N_S} I_{Si} \ \leq \ \sum_{j=1}^{N_W} I_{Wj} \quad \text{if} \quad rs2w \ < \ 1 \end{aligned} \tag{11}$$

where $rs2w$ is the ratio between the mean SP CF averaged over the whole region and the mean WP CF averaged over the whole region. This optional condition can be overseen just by assigning a negative value to the $rs2w$ parameter in the namelistOL.txt file (see below).

Alternatively, and also optionally, minimum and maximum thresholds for the total amounts of installed capacity of each energy in the whole region can be defined (I_{SCmin} and I_{SCmax}, respectively, for the solar power; I_{WCmin} and I_{WCmax}, respectively, for the wind power), that is:

$$\begin{aligned} I_{SCmin} \;&\leq\; \sum_{i=1}^{N_S} I_{Si} \;\leq\; I_{SCmax} \\ I_{WCmin} \;&\leq\; \sum_{j=1}^{N_W} I_{Wj} \;\leq\; I_{WCmax} \end{aligned} \tag{12}$$

Again, these bottom and top limits can be disabled by assigning a value of 0 or below to the minimum thresholds and a huge value to the maximum thresholds.

2.3. Namelist

Finally, a fourth input file is needed. It should contain the values of the parameters N_S, N_W, $rs2w$, I_{SC}, I_{WC}, I_{SCmin}, I_{WCmin}, I_{SCmax}, I_{WCmax}, $I_{SCminRi}$ $\forall i$, $I_{WCminRj}$ $\forall j$, $I_{SCmaxRi}$ $\forall i$ and $I_{WCmaxRj}$ $\forall j$ for the **optimOL.py** and **optimOLS.py** codes to run (even if eventually not needed or used by the code, and thus the assigned value is obviated), as in the sample file namelistOL.txt. Importantly, when these parameters are intended to be used by the codes, they should be within the solution space of the optimization problem. In particular, be careful that I_{SCmin}, I_{WCmin}, I_{SCmax}, I_{WCmax}, $I_{SCminRi}$ $\forall i$, $I_{WCminRj}$ $\forall j$, $I_{SCmaxRi}$ $\forall i$ and $I_{WCmaxRj}$ $\forall j$ are positive values and leave sufficient space for a solution to be found if these thresholds must be taken into account during the optimization process (read above how to disable these constraints otherwise).

2.4. Implementation

For the optimization process we implemented identical procedures as in the OD and ODS problems. In this case, the variables used were X_{Si} and X_{Wi} such that:

$$I_{Si} = X_{Si}^2 \qquad\qquad I_{Wj} = X_{Wj}^2$$

For each of the different conditions a new term was added to the minimizing function in the same way as described in section 1.4.

2.5. Application example

Optimization goal: to minimize the daily fluctuations of the wind-plus-solar production around the residual electricity demand curve. Therefore, A_{Sik} are the daily anomalies of the solar CF (daily means subtracted) in the *i-th* region at time k, A_{Wjk} are the daily anomalies of the wind CF in the *j-th* region at time k, and B_k is the series with the electricity generated with the traditional fuels. NTT is the number of days in the series.

Optimization constrains: to guarantee that the daily wind-plus-solar production is above 80 % of the residual electricity demand at nightime, in monthly averaged terms. Therefore, C_{Sik}/C_{Wjk} are monthly annual cycles with the mean nightime values of the solar/wind CF in the *i-th*/*j-th* region at time k, and M_k is the monthly means of the residual electricity demand at nightime multiplied by 0.8. NMP is 12.

3. Clustering code

The code **make-regions.R** contains the R function that clusters input time series from a gridded dataset of size S in groups of similar temporal variability following Lorente-Plazas et al. (2015). For that, the code should be executed in 3 consecutive times.

The **first execution** performs a **Principal Component Analysis** (PCA) by diagonalizing the correlation matrix, finding an orthonormal basis of eigenvectors (Empirical Orthonormal Functions, EOFs, unit modulus) and the corresponding diagonal entries in this coordinate system, the eigenvalues (Principal Components, PCs, units: same as the original input data).

Based on the PCA results (particularly, in the fraction of the total variance of the original data that is explained by each PC), for the **second execution** of the code, it needs to know the **number of principal components to be retained**. The decision can be subjective or be based on objective rules. An option is to look at a scree plot and check for “elbow” or the onset of an asymptotic behavior, or simply for a specific threshold of the cumulative explained variance to be satisfied. With that, the code performs a Ward's hierarchical clustering and provides the Ward's distance (D), defined in the EOFs space, as a function of the the number of clusters (N_C), from which the **number of clusters to be obtained** should be decided and provided as input for the third execution of the code (the decision may be based in a sufficiently small value of D). The Ward's distance for N_C clusters is defined as the *joining minimum penalty* when going from N_C to $N_C - 1$ clusters, that is:

$$D(N_C) = W(N_C - 1) - W(N_C)$$

where W is defined as:

$$W(N_C) = \sum_{i=1}^{N_C} \sum_{j=1}^{N_G} \parallel \boldsymbol{x}_j - \bar{\boldsymbol{x}}_i \parallel^2 = \sum_{i=1}^{N_C} \sum_{j=1}^{N_G} \sum_{k=1}^{N_{EOF}} (x_{j,k} - \bar{x}_{i,k})^2$$

where N_G is the number of points clustered in the corresponding *i-th* cluster, N_{EOF} is the number of retained PCs/EOFs, $x_{j,k}$ is the *k-th* coordinate of $\boldsymbol{x}_j$ in the EOFs space and $\bar{x}_{i,k}$ is the *k-th* coordinate of the centroid of the *i-th* cluster in the EOFs space. Actually, Ward's method begins with S single-member groups and merges two groups at each step, until all the data are in a single group at the $S - 1$ step. The criterion for choosing which pair of groups to merge at each step is that the pair to be merged minimizes W. The above definition for W is, in fact, such a minimum value. Note that W is 0 for $N_C = S$, and that D is defined for N_C ranging from 2 (when D reaches its maximum value) to S (when D reaches its minimum).

Finally, in the **third execution**, the final groups are determined using non-hierarchical clustering (K-means).

The function inputs are, in this order:

- Number of the execution (1, 2 or 3).
- PCA mode S? (TRUE or FALSE).

- Main input data matrix in ascii or txt format. Each grid point in a column, each time step in a row. May use the CDO command *cdo outputf, %g filein.nc > fileout*.

- Secondary input data matrix, same format described above. The clustering is performed over the main input data matrix, but the code will also provide the regional series (spatially averaged over each cluster of grid points) from the series in this secondary input data matrix.

- Number of principal components to be retained.

- Value of $S - N + 1$.

- Number of clusters to be obtained (N).

- Input file with the geographical coordinates of the grid points in ascii or txt format. Each grid point in a row. Longitude in the first column, latitude in the second column.

- Size of the grid (S), i.e. number of grid points.