

# Sophemis 4

# User Tutorial

[http://optimisation.l2ep.ec-lille.fr/download/Sophemis4\\_User\\_Tutorial.pdf](http://optimisation.l2ep.ec-lille.fr/download/Sophemis4_User_Tutorial.pdf)

## Table of content

I.	Required configuration .....	1
II.	Installation .....	1
1.	Matlab Runtime .....	1
2.	Sophemis .....	2
III.	Using Sophemis with a given model .....	3
1.	Pre-Sizing of a Building Electric System .....	3
2.	Single-objective optimization .....	5
2.	Bi-objective optimizations .....	12
IV.	Structure of folders and files.....	13
1.	Structure of user folder.....	13
2.	Structure of results file .....	14
3.	Exploit results structure to produce statistics and graphs .....	15

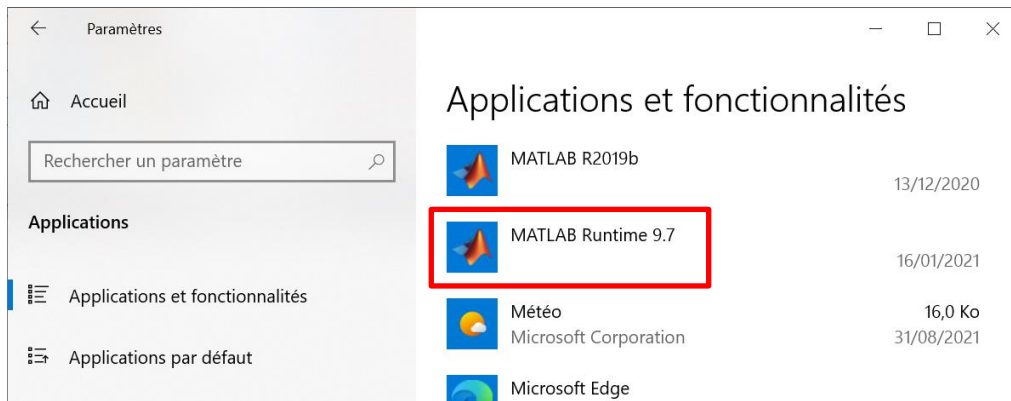
## I. Required configuration

- Windows 64 bits
- Internet connection with no restriction (eduroam doesn't work)
- Administrator permissions for installation of Matlab Runtime
- Screen with full HD resolution (1920x1080) or higher

## II. Installation

### 1. Matlab Runtime

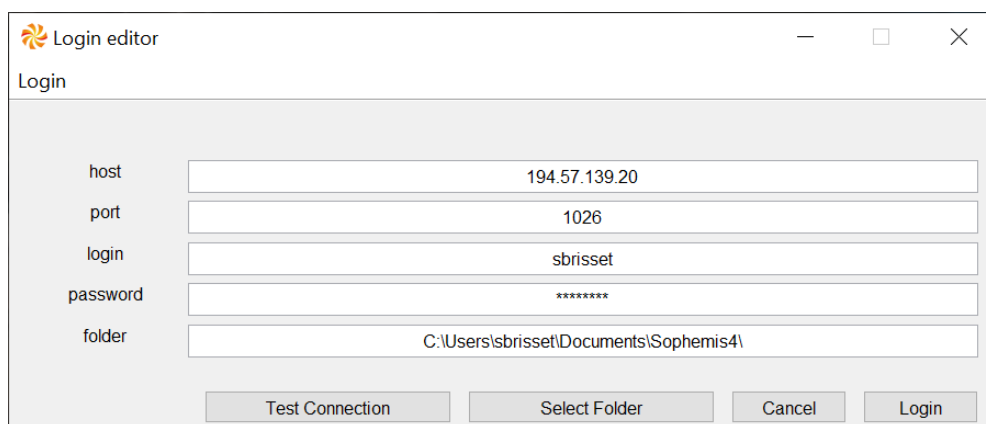
- Check is Matlab Runtime is installed
  - Open Microsoft Windows / Parameters / Apps & features Settings and search for Matlab Runtime 9.7



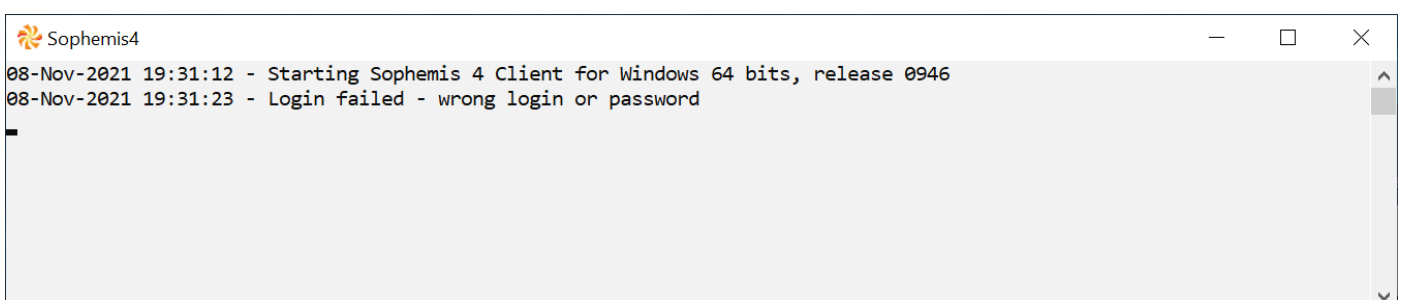
- If you don't find it:
  - Connect with a login having administrator permissions.
  - Download and install Matlab Runtime (MCR) release 2019b from [http://optimisation.l2ep.ec-lille.fr/download/MATLAB\\_Runtime\\_R2019b\\_Update\\_7\\_win64.zip](http://optimisation.l2ep.ec-lille.fr/download/MATLAB_Runtime_R2019b_Update_7_win64.zip)

## 2. Sophemis

- Download Sophemis 4 Client from <http://optimisation.l2ep.ec-lille.fr/download/Sophemis4.exe>, put it in your chosen folder and double-click on the icon to run it.
- The window below opens to select your Sophemis user folder. You can select a folder or create a new one by clicking on the « Select Folder » button. Enter the login and password that were sent to you and click on the « Login » button.

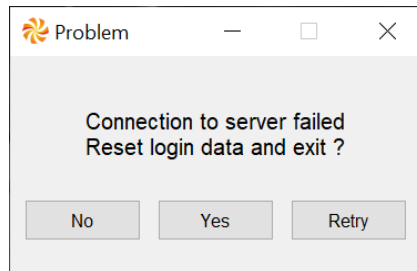


- Click on the « Test Connection » button and look at the message appearing in the console:
  - **Authentication successful:** You can click on the « Login » button that appears at the bottom-right of the « Login editor » window.
  - **Login failed – wrong login or password:** Check your user name and password.

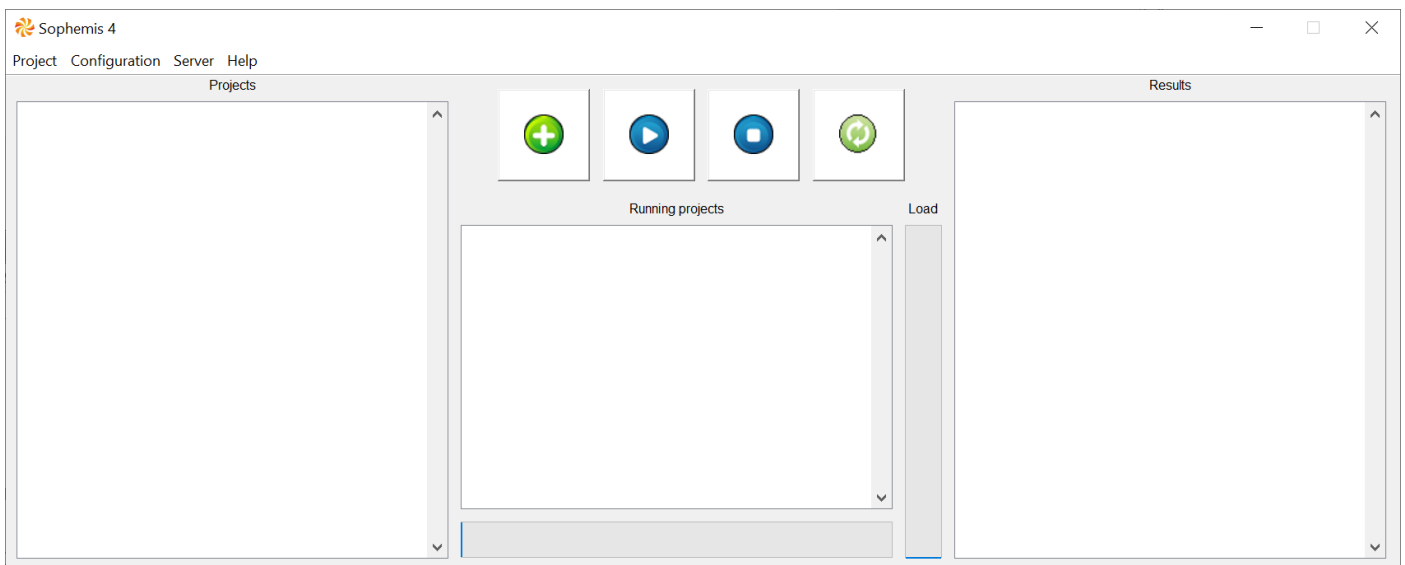


*Note: The color of the fonts and the background color of the console window may differ on your computer. To modify them, right-click on the title bar of the windows and select "Properties" in the context-menu.*


- **Login failed – port is locked by another user:** Try another port between 1026 and 1041.
  - **Login failed – user already logged elsewhere:** You are already connected to another port or you have projects running on another port: Use the same port as during your last connection and if you don't remember it, try one by one all the ports from 1026 to 1041.
  - **Connection to server failed – Unsuccessful open: ... :** Check your Internet connection and the « host » and « port ».
- At the next runs of Sophemis, the window below may appear:



- According to the message appearing in the console, you may choose « Retry » several times (recommended), « Yes » to re-enter the connection information or « No » if you think there is a network problem.
- After clicking on the « Login » button, a window with the title « Sophemis 4 » opens.



### III. Using Sophemis with a given model

- Download the model « BuildingElectricSystem.m » from the link <http://optimisation.l2ep.ec-lille.fr/download/> and move it to the folder « Documents\Sophemis4\models ».
- Press the synchronization button  to upload the model on the server.

#### 1. Pre-Sizing of a Building Electric System

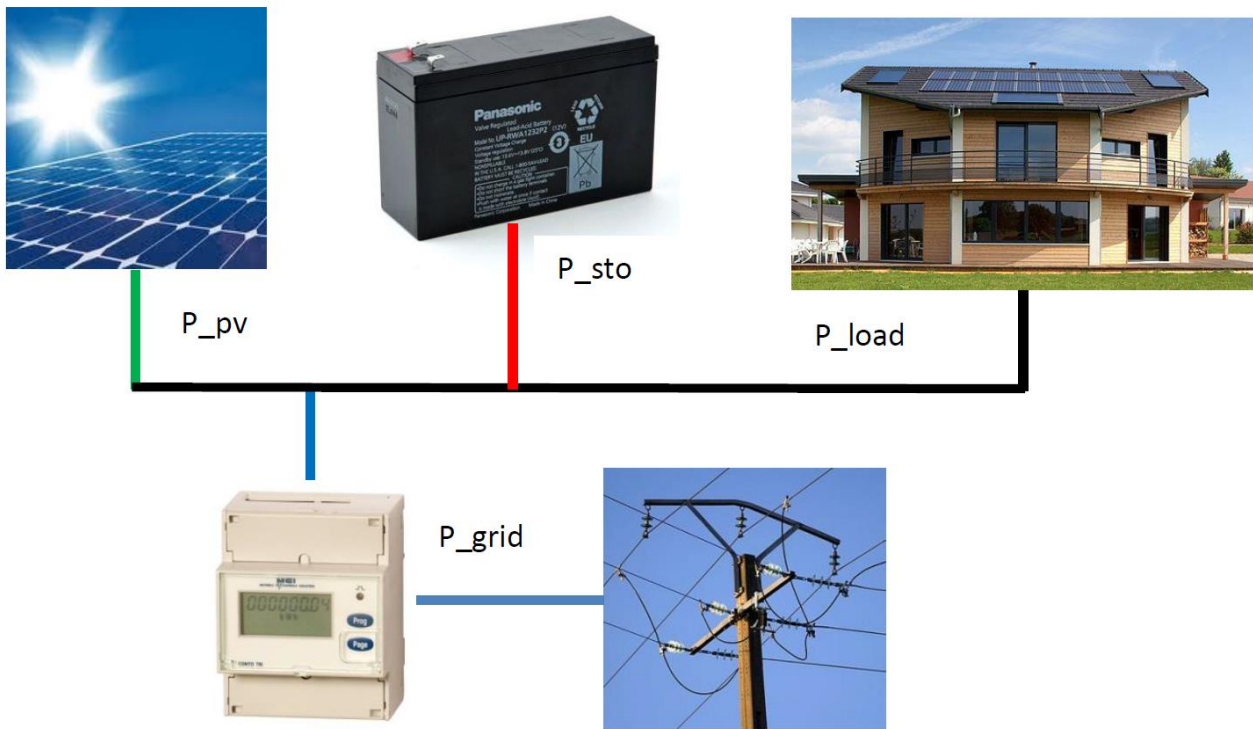
Optimization of electromagnetic devices and electric systems requires thousands of model evaluations. Therefore, an analytical model is a good tradeoff between the accuracy and the computing time.

The optimization problem proposed here is representative of the sizing of a building electric system but the size of the problem is reduced in order to solve it with an all-at-once approach and a black-box model. Therefore, only one day is simulated with a time-step of one hour. Real building electric system optimization requires considering at least a complete year with a time-step of some minutes or less.

The analytical model is based on power flow equations. The equations for the computation of the power coming from the grid as well as the assessment of investment cost for the equipments and energy consumption cost for 20 years are given.

This optimization problem exhibits interesting features such as:

- black-box model
- highly constrained
- badly scaled design variables and constraints
- multimodal, i.e. multiple minima



Among the inputs of the model are two-dimensional variables that are the photovoltaic panels surface  $S_{pv}$  and the maximum storable energy in batteries  $E_{sto}$  as well as the power supplied by the storage system during each hour of a typical day  $P_{sto}$  (24 temporal variables).

Among the outputs of the model are the costs and the state of charge of batteries during a typical day:  $SoC$  (23 temporal values) for the 23 first hours of the day and  $SoC_f$  for the last hour. It is assumed that the state of charge of batteries can vary between 0 and 1 without damaging batteries. The state of charge of batteries at the end of the typical day should return to its initial value that is 0.5.


The cost of the energy from the grid varies along the typical day and the excess energy sent back the grid gives no income.

The aim is to have an electric system with the lowest total cost while respecting some technical constraints:

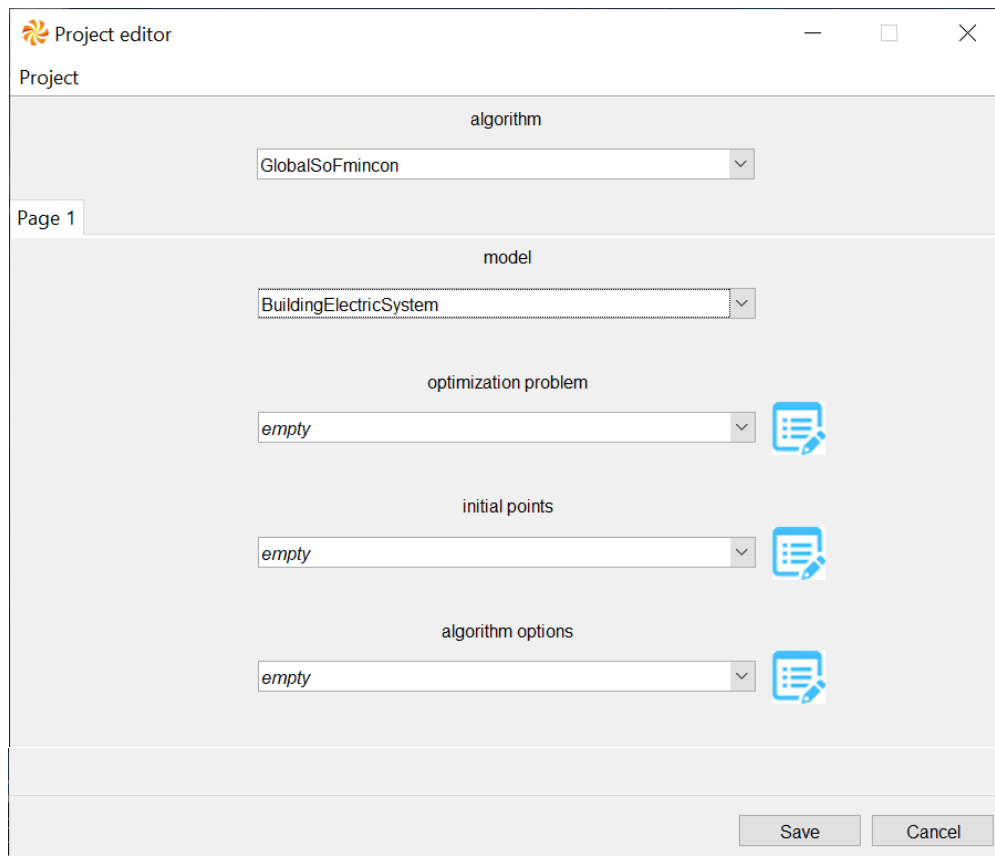
$$\min \text{ cost} \quad s. t. \quad \begin{matrix} 0 < SoC < 1 \\ SoC_f = 0.5 \end{matrix} \quad \text{with} \quad \begin{matrix} -2000W < P_{sto} < 2000W \\ 0 < S_{pv} < 50 m^2 \\ 1J < E_{sto} < 100MJ \end{matrix} \quad (1)$$


During this tutorial, you will learn how to use Sophemis to solve single-objective and bi-objective optimization problems. The example used is the pre-sizing of the building electric system. Sophemis includes some know-how on numerical techniques to tackle difficulties that often occur during engineering optimization. Analysis of results is made with Sophemis graphic interface and completed with Matlab programming.

## 2. Single-objective optimization

- Click on the button  to create a new optimization project.
- The figure below appears. It allows to select an algorithm, a model, an optimization problem, some initial points (optional), and options for the algorithm.
- Select the algorithm « GlobalSoFmincon » and then the model « BuildingElectricSystem ».

*The algorithm « GlobalSoFmincon » is based on the function « fmincon » in « Matlab Optimization Toolbox ». A better convergence is achieved with a scaling of variables, objective and constraints. This algorithm has a good probability to find the global optimum by using multiple starting points and gives the ranked list of local optima. The parallel processing of optimizations reduces the time with a speed-up close to the number of workers (cores).*





- Click on the icon  at the right of « optimization problem » to define the optimization problem. A window opens with all the inputs and outputs of the model.



- You have also to define the inequality constraints on SoC, the equality constraint on SoCf, and the objective to minimize with a typical value of 10000€:

Outputs			
Ppv	<input type="button" value="unused"/>		
Pgrid	<input type="button" value="unused"/>		
SoC	<input type="button" value="inequality"/>	min: <input type="text" value="0"/>	max: <input type="text" value="1"/>
SoCf	<input type="button" value="equality"/>	value: <input type="text" value="0.5"/>	
Egrid	<input type="button" value="unused"/>		
cinv	<input type="button" value="unused"/>		
cener	<input type="button" value="unused"/>		
cost	<input type="button" value="minimize"/>	typical: <input type="text" value="10000"/>	

- Save the optimization problem as « MinCost ». The Problem editor window closes.
- In the Project editor window, click on the icon  at the right of « algorithm options » to define the algorithm options. A window opens with all the options that are set to their default values.

 Options editor \_ □ ×

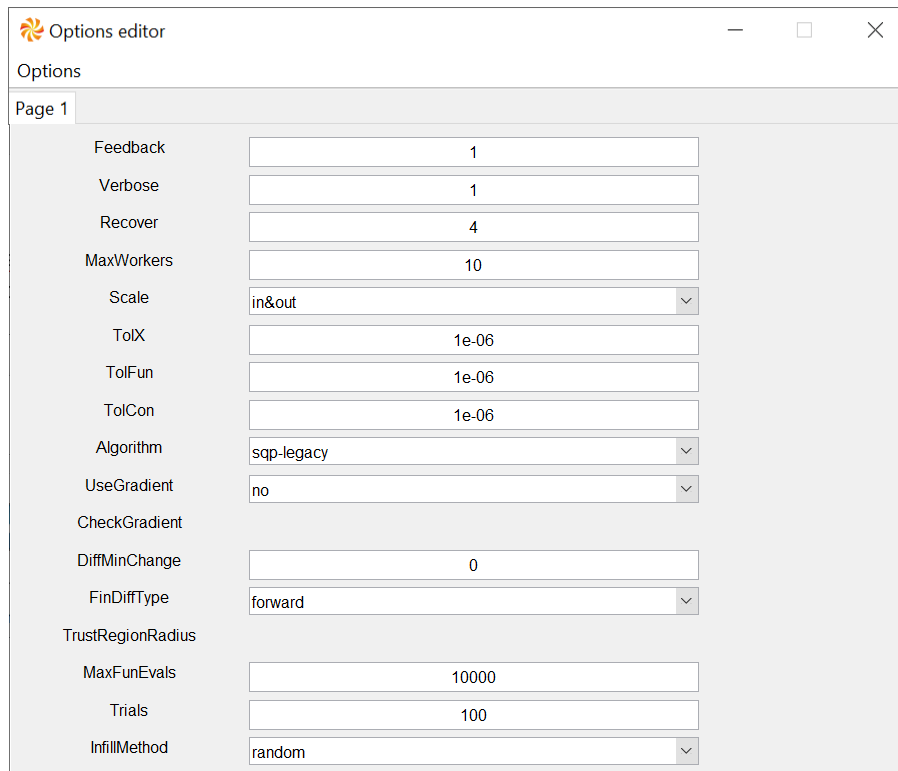
Options


Page 1

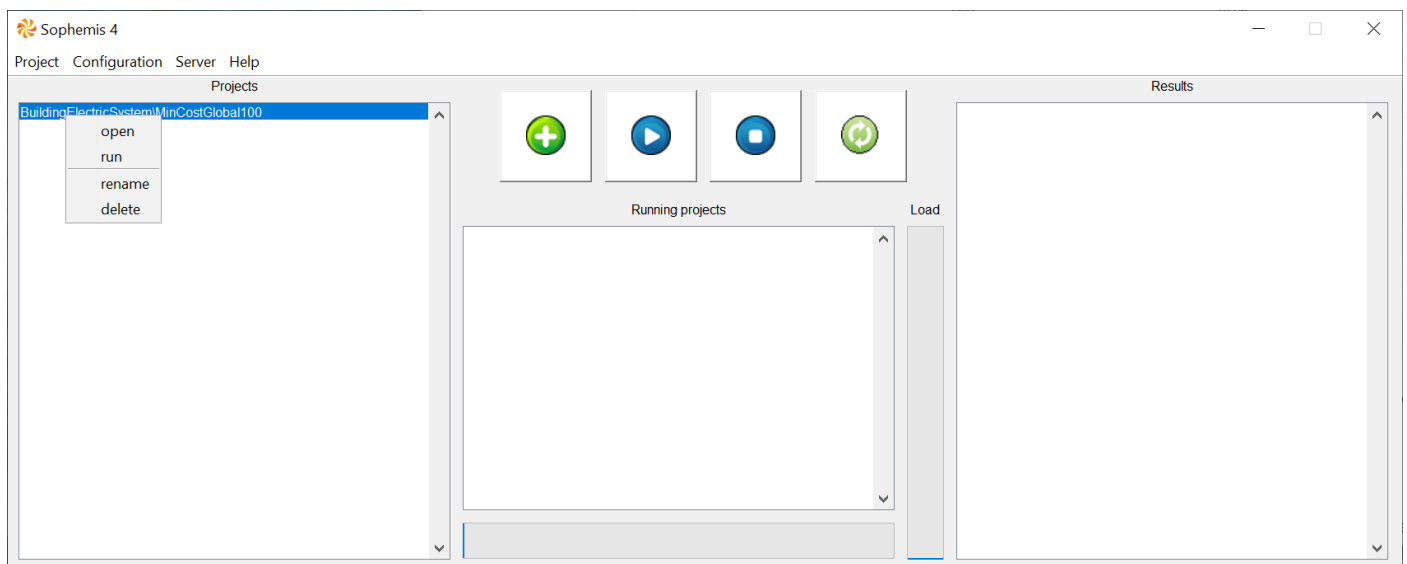
Feedback	<input type="text" value="1"/>
Verbose	<input type="text" value="1"/>
Recover	<input type="text" value="4"/>
MaxWorkers	<input type="text" value="0"/>
Scale	<input type="text" value="in&amp;out"/>
TolX	<input type="text" value="1e-06"/>
TolFun	<input type="text" value="1e-06"/>
TolCon	<input type="text" value="1e-06"/>
Algorithm	<input type="text" value="sqp-legacy"/>
UseGradient	<input type="text" value="no"/>
CheckGradient	
DiffMinChange	<input type="text" value="0"/>
FinDiffType	<input type="text" value="forward"/>
TrustRegionRadius	
MaxFunEvals	<input type="text" value="1000"/>
Trials	<input type="text" value="1"/>
InfillMethod	<input type="text" value="random"/>

Options « Feedback », « Verbose » and « Recover » are integer values. « Feedback » set the level of information that is returned during optimization for monitoring the convergence. « Recover » is the level of information that is recorded at the end of the optimization. The values are: 0 for no information, 1 for the progress indicator, 2 for the optimization summary, 3 for the synthesis graph, 4 (recommended) for the table of optimal solutions, 5 for the table of iterations, and 6 for the table of evaluations. A value of « Recover » higher than 4 leads to high memory requirement, big results file on the disk. A value of « Feedback » higher than 1 leads to high network traffic and possible jam. « Verbose » set the level of information in the optimization log. A value higher than 1 is useful to fix bugs in models and algorithms but leads to a big results file.

- Change the value of « Trials » to 100, « MaxFunEvals » to 10000, and « MaxWorkers » to 10 and save the options as « Trials100Workers10 ». The Options editor closes.

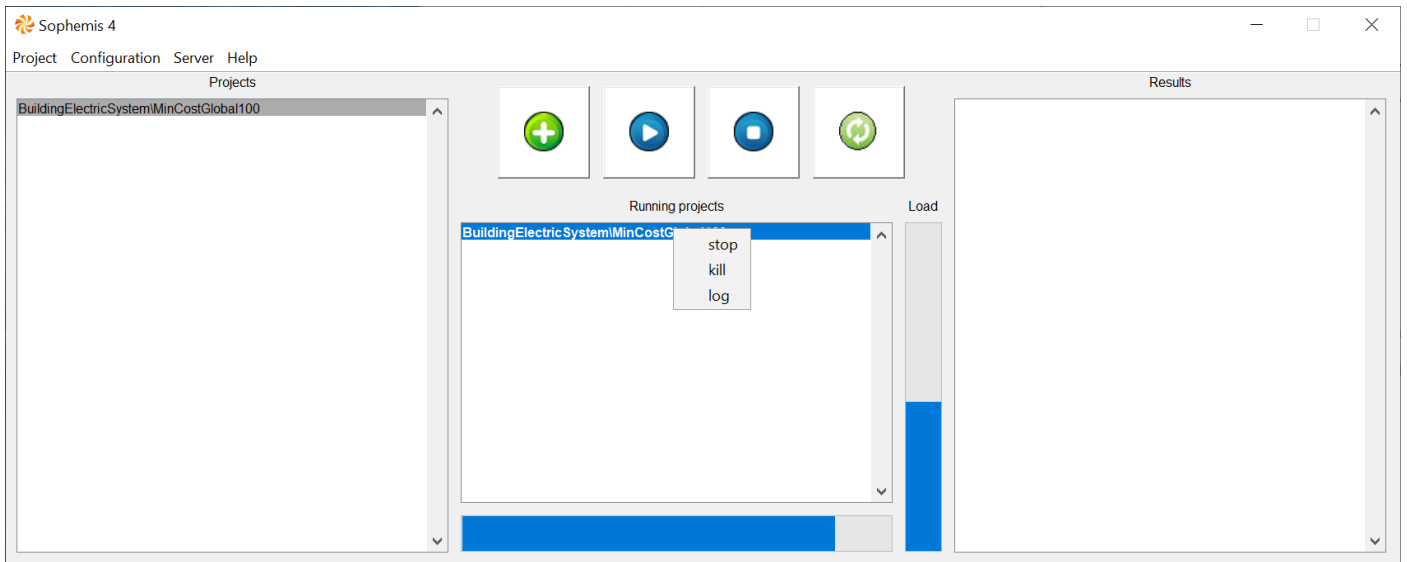


- Go back to « Project editor » windows and click on the button « Save ». A project name is required. Use « MinCostGlobal100 ». The project editor window closes.
- Go back to « Sophemis 4 » window. All the optimization projects are in the left frame of the window. To run one project, you may use the context menu and select run or click on the button .

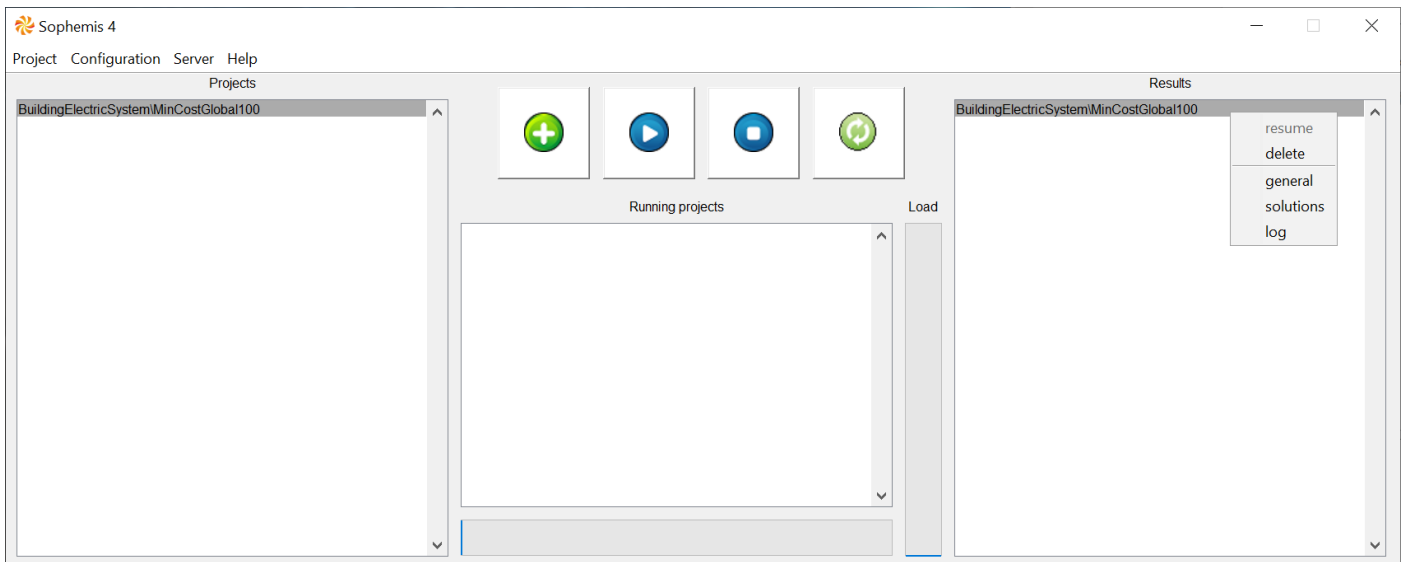


- The project now appears in the running project frame at the center of the window. The vertical bar gives the load of the server that is the number of workers used by all users. The horizontal bar gives the progress of the optimization. The context menu allows to kill a project, stop it or see the diary (log) of the optimization process. A stopped project can be resumed later.





- When the optimization is finished, the project appears in the results frame at the right side of the window. The context menu allows to see the optimization summary (general), the solutions and the diary (log) of the optimization process. A project stopped can be resumed here. The number of items in the context menu depends on the value of recover in the algorithm's options.



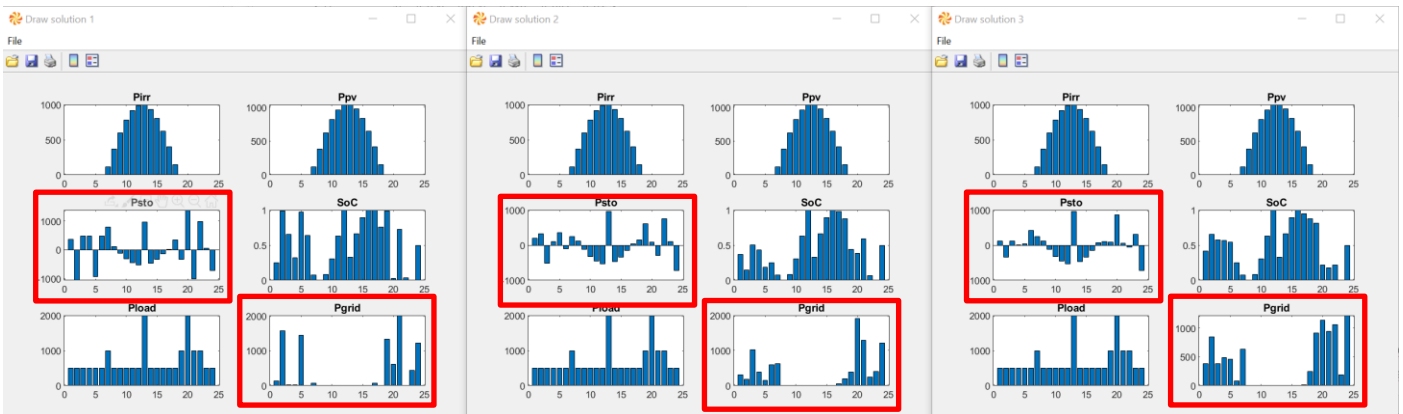
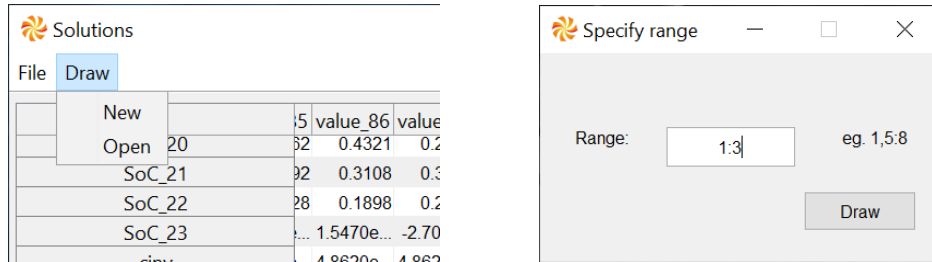
- Select general in the context menu. A window appears with the exit flag (a positive value means success), the total time in seconds, the total number of evaluations for the 100 trials and the progress that is 1 when the optimization is finished.

Parameter	Value
exitflag	1
time	9.04924
evaluations	205180
progress	1

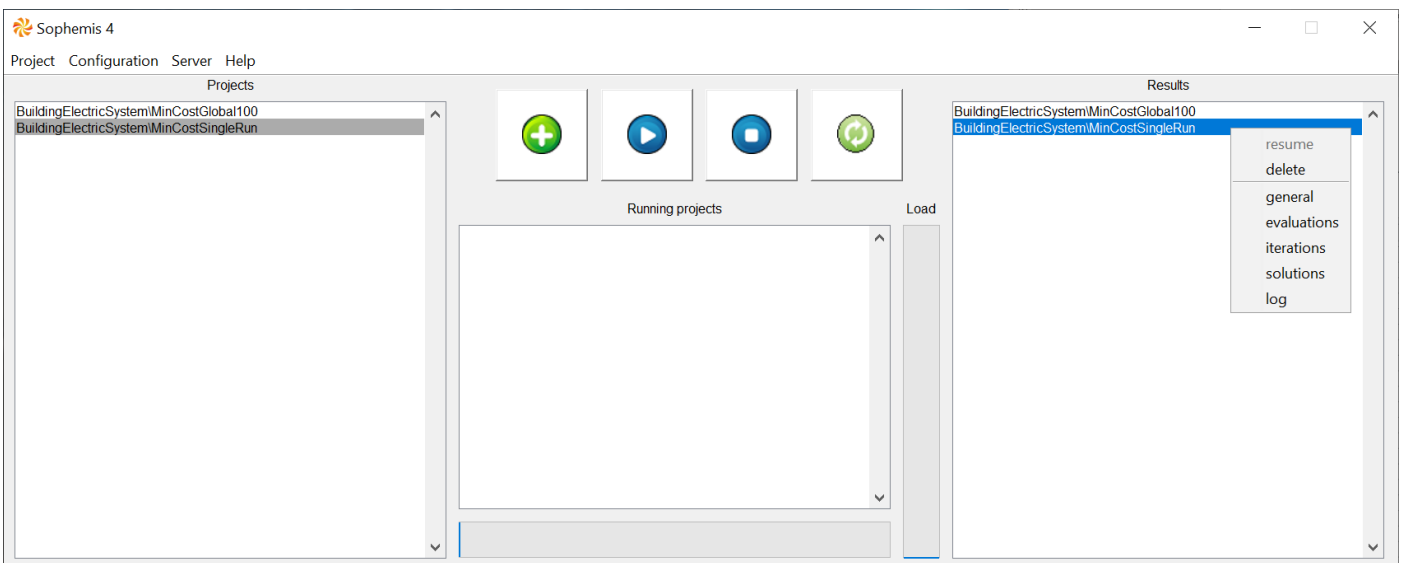
- Select solutions in the context menu. A window with all the solutions ranked is displayed. You can see that 86 of them have the same total cost. As the initial points are selected randomly, your number may slightly differ.

	5	value_86	value_87	value_88	value_89	value_90	value_91	value_92	value_93	value_94	value_95	value_96	value_97	value_98	value_99	value_100	type	min	max	comment	unit
SoC_20	32	0.4321	0.2412	0.1694	0.3846	0.3370	0.6551	0.3827	0.1637	0.1317	0.4517	0.1713	0.2243	0.8458	0.8563	0.1319					
SoC_21	32	0.3108	0.3565	0.4317	0.2114	0.2358	0.2815	0.1213	0.0673	0.5399	0.5368	0.2811	0.1791	0.7632	0.3231	0.1195					
SoC_22	38	0.1898	0.2180	0.0933	0.0982	0.1207	0.1045	0.0614	0.0412	0.0594	0.0848	0.1469	0.0414	0.0556	0.1724	0.1485					
SoC_23	...	1.5470e...	-2.7091...	2.0419e...	1.4299e...	4.7643e...	5.3244e...	-8.5178...	4.4869e...	-4.7140...	2.7079e...	5.4630e...	2.7926e...	4.4129e...	4.1208e...	9.2885e...					
cinv	...	4.8620e...	4.8625e...	4.8623e...	4.8626e...	4.8612e...	4.8610e...	4.8612e...	4.8665e...	4.8613e...	4.8578e...	4.8693e...	4.8588e...	4.8303e...	4.8901e...	4.8960e+...				investment cost	€
cenerv	...	1.0626e...	1.0626e...	1.0626e...	1.0626e...	1.0627e...	1.0628e...	1.0628e...	1.0622e...	1.0628e...	1.0632e...	1.0637e...	1.0666e...	1.0606e...	1.0625e+...					energy grid cost	€
Engrid	...	2.3614e...	2.3613e...	2.3614e...	2.3613e...	2.3616e...	2.3616e...	2.3607e...	2.3616e...	2.3624e...	2.3605e...	2.3632e...	2.3678e...	2.3578e...	2.3639e+...					total energy from grid	MWh
cost	...	1.5488e...	1.5489e...	1.5489e...	1.5489e...	1.5489e...	1.5489e...	1.5489e...	1.5489e...	1.5490e...	1.5492e...	1.5496e...	1.5496e...	1.5496e...	1.5521e+...	minimize				total cost	€
time	34	2.1048	0.3299	0.4512	2.1924	0.5649	0.5988	0.4434	2.1629	0.4134	0.3693	0.4895	2.0782	0.3428	0.2989	0.2310					

- Select Draw / New in the menu of the Solutions window and enter 1:3 in the edit box to draw the 3 first best solutions. For these solutions, the cost is the same but the curves of Psto and Pgrid are different.



- Create a new optimization project with the algorithm « SoFmincon », the same model, the same optimization problem, no initial point and the default options except « MaxFunEvals » and « Recover » that are respectively 10000 and 6. Save it as « MinCostSingleRun », run it and look at the context menu in the results frame. Iterations and evaluations are now available.



The algorithm « SoFmincon » is based on the function « fmincon » in « Matlab Optimization Toolbox ». A better convergence is achieved with a scaling of variables, objective and constraints. If the derivatives of the objective and constraints is made by finite difference (gradient is not provided), the parallel processing of evaluations reduces the time for models with computing time higher than one second.

- Select solutions in the context menu. A window with the solution is displayed. Comments, units, and the position in the optimization problem are reminded.

	value	type	min	max	comment	unit
SoC_20	0.3588					
SoC_21	0.6375					
SoC_22	0.2430					
SoC_23	-2.5654e-12					
cinv	4.8620e+03				investment cost	€
cener	1.0626e+04				energy grid cost	€
Egrid	2.3614e+11				total energy from grid	J
cost	1.5488e+04 minimize				total cost	€
time	2.4640					

- Select iterations in the context menu. A window appears with information at each iteration of the algorithm. funccount is the cumulative number of evaluations, f is the objective value, constraint violation is the maximum value of violations, first order optimality and step size should reduce at each iteration if convergence is in progress.

	iteration	funccount	f	constrviolation	firstorderopt	stepsize
1	0	27	5.7613	0.3238	4.0000	0
2	1	55	3.5766	0.4317	2.3232	1.2400
3	2	84	2.6253	0.0169	1.9585	0.6802
4	3	112	1.9469	0.0010	2.3993	0.6711
5	4	141	1.8665	0.0020	1.6669	0.4897
6	5	170	1.7456	0.0013	7.1294	0.3818
7	6	200	1.7430	0.0014	7.1273	0.1702
8	7	229	1.6879	2.4486e-04	15.7409	0.1241
9	8	257	1.6727	1.6286e-04	3.7137	0.1932
10	9	287	1.6558	1.1888e-04	1.3198	0.1326
11	10	317	1.6533	8.1624e-05	0.5892	0.1135
12	11	348	1.6406	5.8803e-05	0.5928	0.1119
13	12	380	1.6358	4.9185e-05	2.8616	0.0332
14	13	409	1.6204	2.6918e-05	1.4916	0.1381
15	14	440	1.6142	2.0991e-05	0.5964	0.0561
16	15	470	1.6099	1.4206e-05	0.7264	0.0919

- Select evaluations in the context menu. A window with all the evaluations is displayed.

	SoC_14	SoC_15	SoC_16	SoC_17	SoC_18	SoC_19	SoC_20	SoC_21	SoC_22	SoC_23	cinv	cener	Egrid	cost	time	x
1	0.3706	0.4658	0.5632	0.5745	0.5503	0.4549	0.4581	0.5499	0.6189	0.7143	3.3766e...	2.3848e...	4.7777e...	5.7613e...	0.0037	0
2	0.3706	0.4658	0.5632	0.5745	0.5503	0.4549	0.4581	0.5499	0.6189	0.7143	3.3766e...	2.3848e...	4.7777e...	5.7613e...	3.7340e...	0
3	0.3706	0.4658	0.5632	0.5745	0.5503	0.4549	0.4581	0.5499	0.6189	0.7143	3.3766e...	2.3848e...	4.7777e...	5.7613e...	3.3516e...	0
4	0.3706	0.4658	0.5632	0.5745	0.5503	0.4549	0.4581	0.5499	0.6189	0.7143	3.3766e...	2.3848e...	4.7777e...	5.7613e...	3.3713e...	0
5	0.3706	0.4658	0.5632	0.5745	0.5503	0.4549	0.4581	0.5499	0.6189	0.7143	3.3766e...	2.3848e...	4.7777e...	5.7613e...	0.0043	0
6	0.3706	0.4658	0.5632	0.5745	0.5503	0.4549	0.4581	0.5499	0.6189	0.7143	3.3766e...	2.3848e...	4.7777e...	5.7613e...	4.1796e...	0
7	0.3706	0.4658	0.5632	0.5745	0.5503	0.4549	0.4581	0.5499	0.6189	0.7143	3.3766e...	2.3848e...	4.7777e...	5.7613e...	3.1544e...	0

- Create a new optimization project with the algorithm « GlobalSoFmincon », the same model, the same optimization problem, no initial point and the same options as in « Trials100Workers10 » except Scale that is set to none. Save the options as « Trials100Workers10ScaleNone » and save the new project as

« MinCostGlobal100ScaleNone », run it and look at the solutions. You can see that very few solutions are similar to the best solution and this last has a higher cost compared to the best solution with scaling.

	value_1	value_2	value_3	value_4	value_5	value_6	value_7	value_8	value_9	value_10	value_11	value_12	value_13	value_14	value_15	value_16	value_17	value_18	valu	
SoC_23	-1.2546e-14	-3.1419...	1.1885e...	-7.2164...	-9.4702...	-1.2906...	-1.7542...	0.0036	0.0103	-1.3800...	-6.3083...	0.0316	0.0535	0.0729	0.0964	3.9956e...	-7.0099...	0.1425	5.86	
cinv	4.9836e+03	5.0866e...	6.0851e...	6.4061e...	7.2248e...	8.7560e...	9.3427e...	9.5967e...	9.6952e...	4.4167e...	4.3781e...	1.0030e...	1.0407e...	1.0773e...	1.1264e...	4.0333e...	3.9371e...	1.2413e...	3.69	
cener	1.0510e+04	1.0411e...	9.4551e...	9.1478e...	8.3640e...	6.9540e...	6.4259e...	6.2090e...	6.1419e...	1.1444e...	1.1580e...	5.9342e...	5.6573e...	5.4080e...	5.0732e...	1.2335e...	1.2546e...	4.2908e...	1.30	
Egrid	2.3413e+11	2.3243e...	2.1598e...	2.1069e...	1.9720e...	1.7296e...	1.6409e...	1.6025e...	1.5876e...	2.4852e...	2.5067e...	1.5404e...	1.4801e...	1.4248e...	1.3505e...	2.6326e...	2.6677e...	1.1768e...	2.75	
cost	1.5493e+04	1.5498e...	1.5540e...	1.5554e...	1.5589e...	1.5710e...	1.5769e...	1.5806e...	1.5837e...	1.5861e...	1.5958e...	1.5964e...	1.6064e...	1.6181e...	1.6338e...	1.6368e...	1.6483e...	1.6704e...	.67	
time		1.2881	1.5025	1.0663	1.1014	1.5312	2.2834	1.2309	1.4830	1.3678	0.4870	2.1399	1.0528	1.1330	1.4429	1.5333	2.1646	0.4176	1.4054	0.

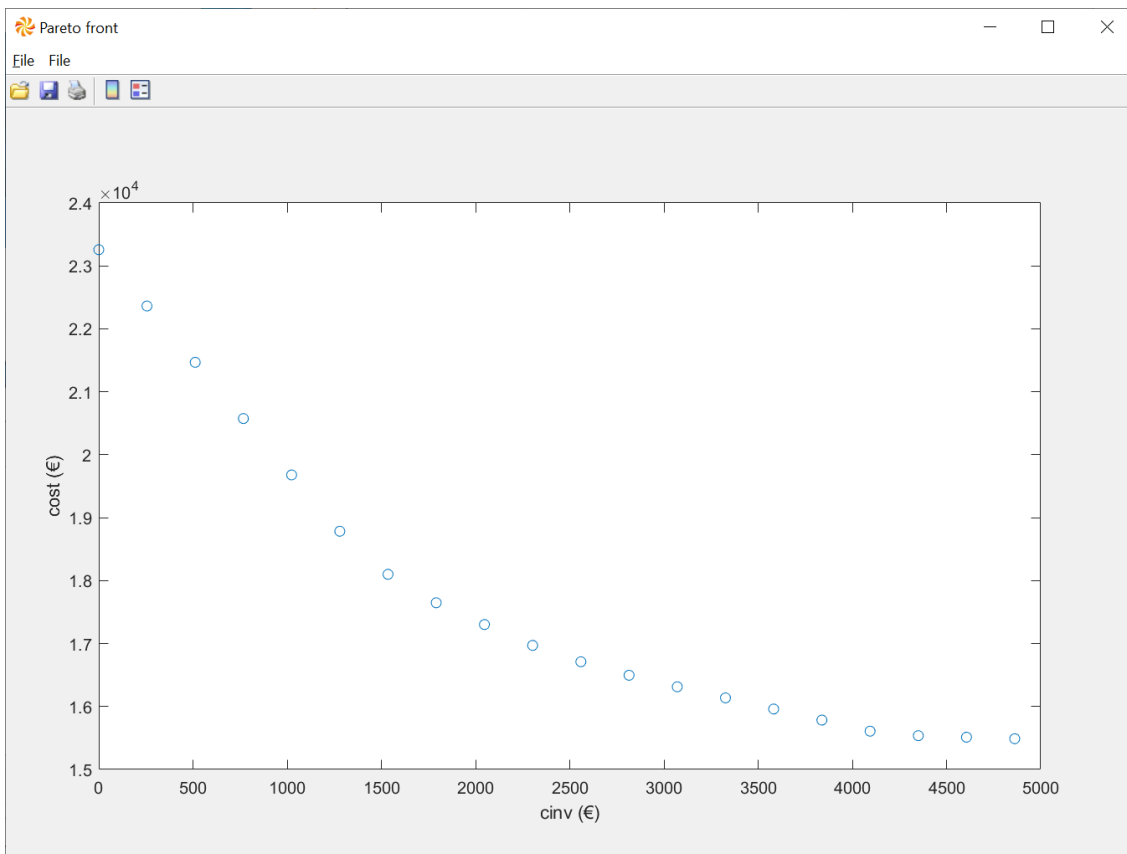
- Once again, the initial points are randomly selected so your best cost and the number of similar costs may differ.

## 2. Bi-objective optimizations

The aim is now to have an electric system with the lowest total and investment costs while respecting some technical constraints:

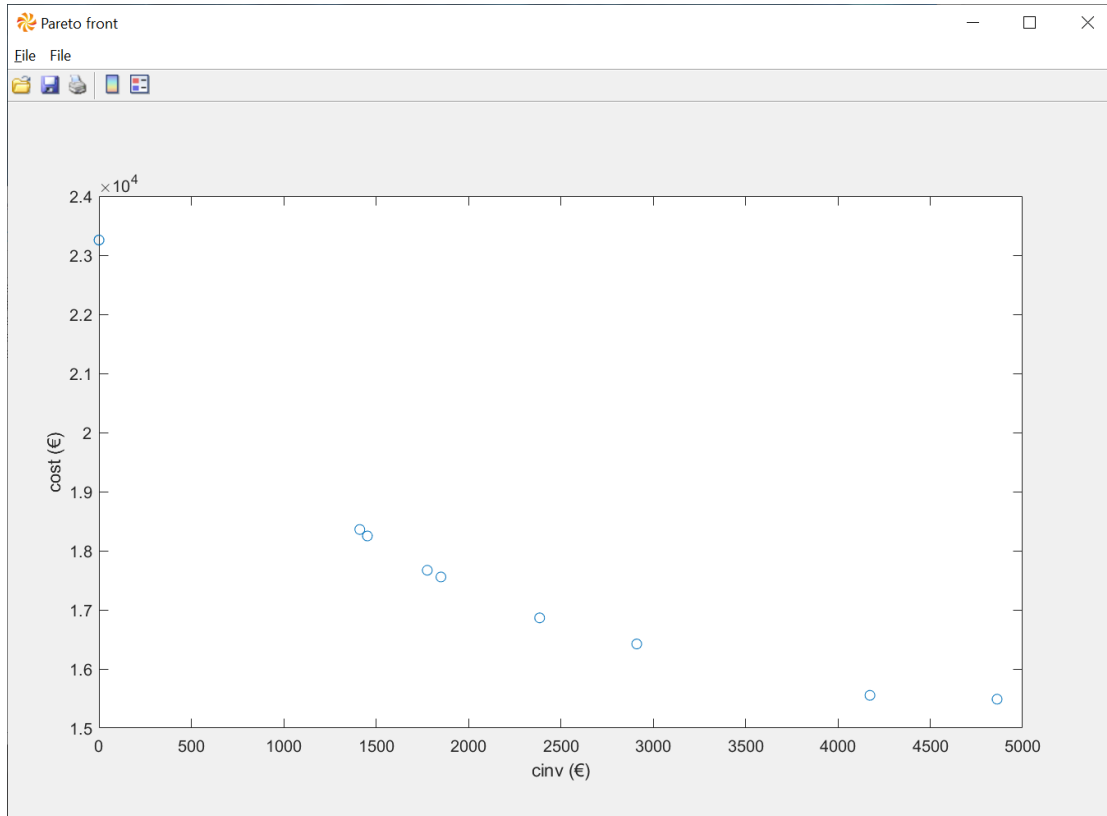
$$\min \begin{bmatrix} cost \\ cinv \end{bmatrix} \quad s.t. \quad \begin{matrix} 0 < SoC < 1 \\ SoCf = 0.5 \end{matrix} \quad \text{with} \quad \begin{matrix} -2000W < Psto < 2000W \\ 0 < Spv < 50 m^2 \\ 1J < Esto < 100MJ \end{matrix} \quad (2)$$

- Create a new optimization project with the algorithm « ParetoSoFmincon » and the same model. Select the optimization problem « MinCost », add a new output to minimize cinv with a typical value of 1000€, and save the optimization problem as « MinCostCinv ». In the options, set the values of « MaxWorkers » (10), « MaxFunEvals » (10000), Trials (10), and Pareto (20) and save as « Workers10Evals10000Trials10Pareto20 ». Save the project as « MinCostCinvEpsilon » and run it. Use the context menu in the results frame to display the graph.



« ParetoSoFmincon » algorithm transforms a multi-objective optimization problem into a series of single-objective problems that are solved by using the « GlobalSoFmincon » algorithm. Each solution of one single-objective problem is one point on the Pareto front that is the set of best tradeoffs between multiple conflicting objectives. The options « Pareto » and « Method » set the number of points on the Pareto front and the transformation method, respectively.

- Create a new optimization project with the same algorithm, the same model, and the same optimization problem. Select the options « Workers10Evals10000Trials10Pareto20 » and change the value of Method to weighted-sum, and save the options as « Workers10Evals10000Trials10Pareto20Weighted ». Save the project as « MinCostCinvWeighted » and run it. Use the context menu in the results frame to display the graph.



Obviously, the distribution of points along the Pareto front is better with the epsilon-constraint method. The bad distribution of points is a known issue of the weighted-sum method.

## IV. Structure of folders and files

This chapter explains the structure of Sophemis user folder and the structure of optimization project results in order to use them for producing any statistics and specific graphic.

### 1. Structure of user folder

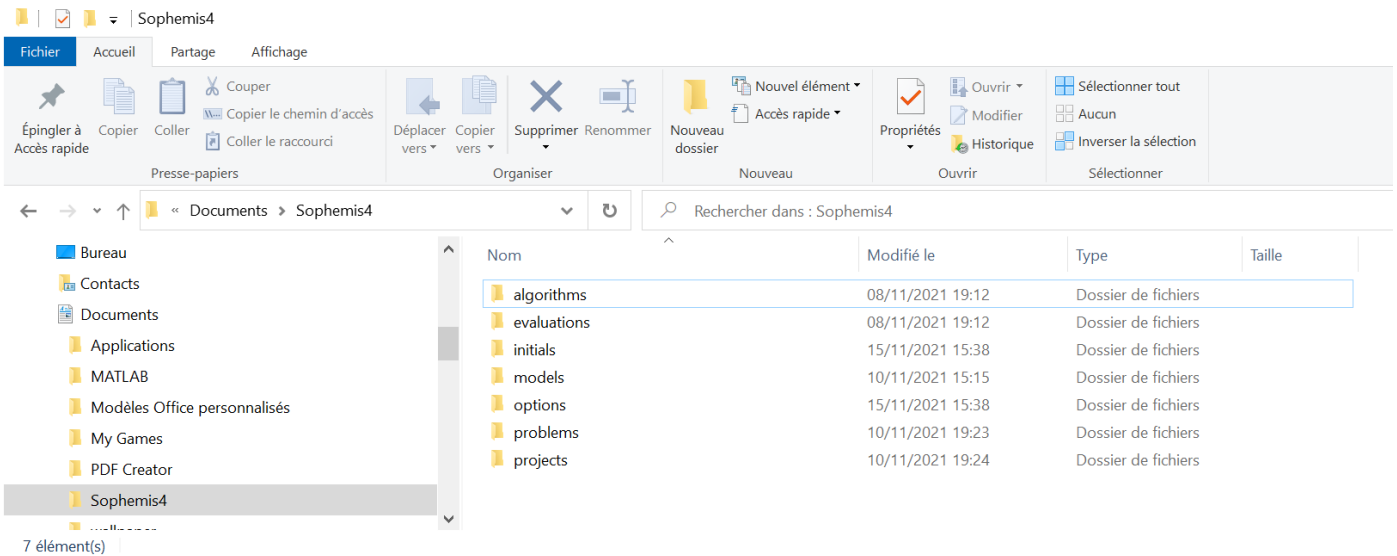
- Open the Sophemis user folder to discover its structure.

The folder « algorithms » contains all the algorithms created by the user. At this time, it is empty.

The folder « evaluations » contains all the evaluations of models made with the algorithm « EvaluateInputs ». At this time, it is empty.

The folder « models » contains all the models downloaded or created by the user. At this time, the only file is « BuildingElectricSystem.m ».

The folder « options » contains the options for all algorithms. There is one subfolder for each algorithm. In each subfolder, you find the options that have been saved with the file name you define.

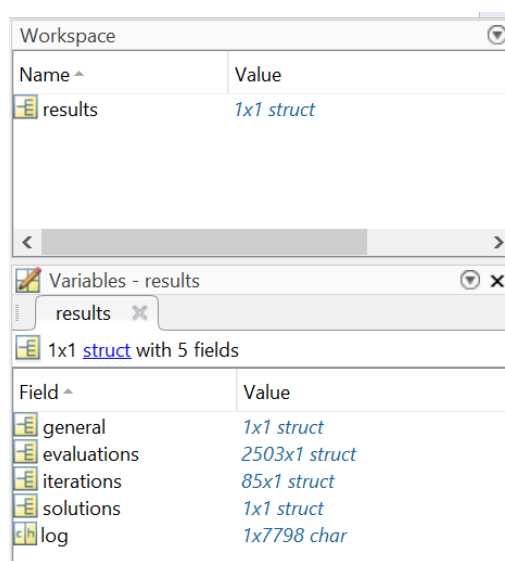


The folder « problems » contains the optimization problems. There is one subfolder for each model. At this time, the only subfolder is « BuildingElectricSystem ». In each subfolder, you find the problems that have been saved with the file name you define.

The folder « projects » contains the optimization projects. There is one subfolder for each project. The subfolder name is the name of the project that you have defined. In each subfolder, you find the Matlab data files (.mat) for the options, problem, and results. The « name.mat » file contains the names of algorithm and model. Matlab Figure files generated by using the « Draw » in the solutions and evaluations windows are also stored here.

## 2. Structure of results file

- Open the folder « projects\BuildingElectricSystem\MinCostSingleRun » and double-click on the results.mat file. One window « Import Wizard » pops up. Click on « Finish » button and open the variable « results » in the workspace. You see the fields inside the results structure that appeared also in the context menu in results frame.



- Open the « evaluations » field. You see that this is a structure with fields and two structures inside.

Fields	inModel	outModel	time	x	f	constrviolation	g	h
1	1x1 struct	1x1 struct	0.0037	26x1 double	5.7613	0.3238	46x1 double	0.3238
2	1x1 struct	1x1 struct	3.7340e-04	26x1 double	5.7613	0.3238	46x1 double	0.3238
3	1x1 struct	1x1 struct	3.3516e-04	26x1 double	5.7613	0.3238	46x1 double	0.3238
4	1x1 struct	1x1 struct	3.3713e-04	26x1 double	5.7613	0.3238	46x1 double	0.3238
5	1x1 struct	1x1 struct	0.0043	26x1 double	5.7613	0.3238	46x1 double	0.3238
6	1x1 struct	1x1 struct	4.1796e-05	26x1 double	5.7613	0.3238	46x1 double	0.3238

- Open the « inModel » and « outModel » structures. You see the values of all inputs and outputs, respectively.

Field ^	Value
Pload	24x1 double
Pirr	24x1 double
cgrid	24x1 double
cpv	400
epv	0.1500
csto	4.0000e-04
SoCi	0.5000
Psto	24x1 double
Spv	11.9455
Esto	7.2469e+07

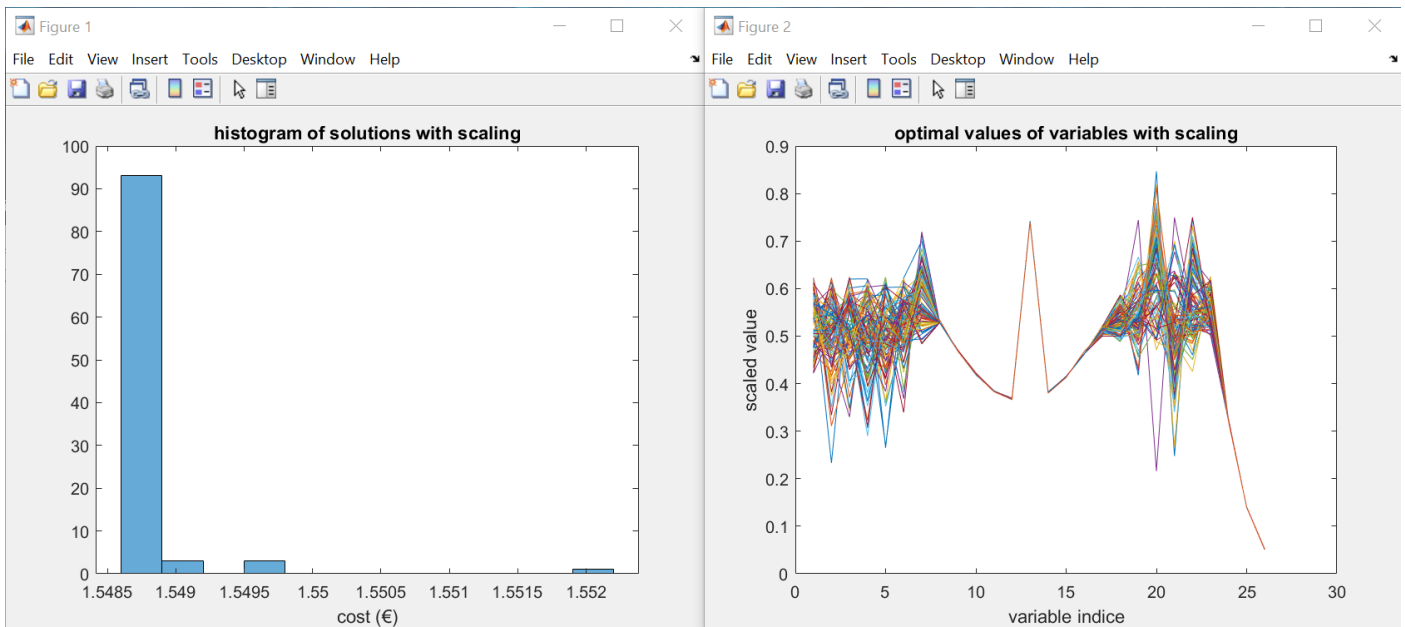
Field ^	Value
Ppv	24x1 double
Pgrid	24x1 double
SoCf	0.6619
SoC	23x1 double
cinv	3.3766e+04
cener	2.3848e+04
Egrid	4.7777e+11
cost	5.7613e+04

### 3. Exploit results structure to produce statistics and graphs

- Import the results of the project « MinCostGlobal100 » and enter the instructions below in the Matlab command window.

```
>> cost = [results.solutions.f];
>> scaledVar = [results.solutions.x];
>> figure
>> histogram(cost)
>> xlabel('cost (€)')
>> title('histogram of solutions with scaling')
>> figure
>> plot(scaledVar)
>> xlabel('variable indice')
>> ylabel('scaled value')
>> title('optimal values of variables with scaling')
```

Figure 1 is a histogram of optimal values of the objective. For the 100 trials, the optimal cost of about 90 of them are very close to the lowest value. This means that the probability to find an optimal solution from any initial point is about 90% what is relatively high. On figure 2, the values of the 26 optimization variables are one curve. The order of variables is the same than in the Problem window: the 24 first values are Psto, then Spv and Esto. The scaled values are between 0 and 1. 0 stands for lower bound and 1 for upper bound. About 100 curves are drawn in this graph. It can be seen that the optimal values differ for many variables depending on the initial points. However, the last two values for Spv and Esto have always the same values such as Psto for sunny hours.



- Now, import the results of the project « MinCostGlobal100ScaleNone » and enter the instructions below in the Matlab command window.

```
>> cost = [results.solutions.f];
>> figure
>> histogram(cost, 'BinWidth', 1000)
>> xlabel('cost (€)')
>> title('histogram of solutions without scaling')
```

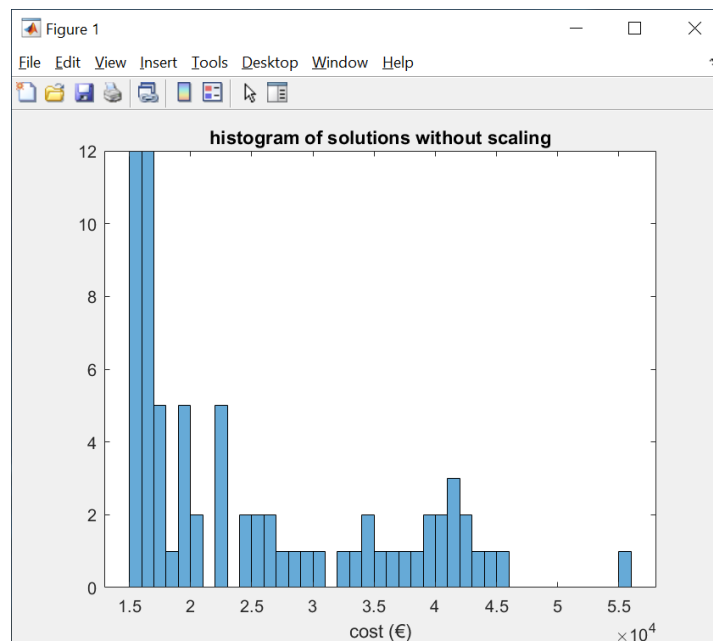


Figure 1 is now a histogram of optimal values of the objective where the bin width is set to 1000€. For the 100 trials, the lowest value of cost or a value less than this value plus 1000€ is found 12 times. This means that the probability to find an optimal solution from any initial point is low without scaling of variables.

- Import the results of the project « MinCostCinvEpsilon » and enter the instructions below in the Matlab command window.

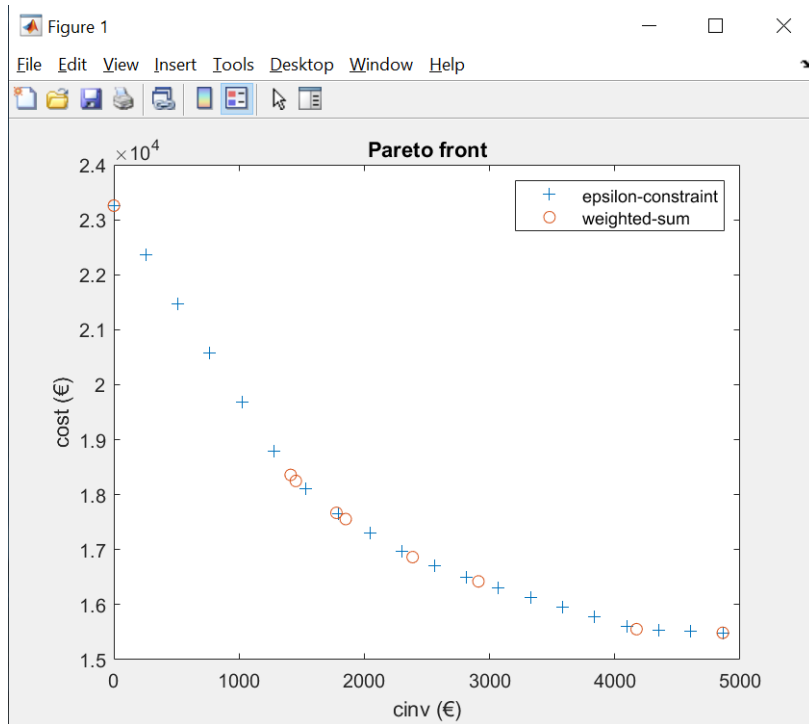
```
>> figure
>> plot(results.graph.values(1,:), results.graph.values(2,:), '+')
>> xlabel(results.graph.labels{1})
>> ylabel(results.graph.labels{2})
>> title('Pareto front')
```



```
>> hold on
```

- Import the results of the project « MinCostCinvWeighted » and enter the instructions below in the Matlab command window.

```
>> plot(results.graph.values(1,:), results.graph.values(2,:), 'o')  
>> legend('epsilon-constraint', 'weighted-sum')  
>> hold off
```



Now, the Pareto fronts built with the epsilon-constraint and weighted-sum methods are on the same graph.

*This Sophemis 4 User Tutorial is finished.*

*You can continue with the Sophemis 4 Designer Tutorial to learn how to build a new model.*

[http://optimisation.l2ep.ec-lille.fr/download/Sophemis4\\_Designer\\_Tutorial.pdf](http://optimisation.l2ep.ec-lille.fr/download/Sophemis4_Designer_Tutorial.pdf)