

Project no: 608860

Project acronym: IDE4L

Project title: IDEAL GRID FOR ALL

Deliverable 6.1

Optimal scheduling tools for day-ahead operation and intraday adjustment

Part III: Commercial Aggregator tools and test results

Due date of deliverable: 31.08.2015

Actual submission date: 2.11.2015

Start date of project: 01.09.2013 Duration: 36 months

Lead beneficiary name: Catalonia Institute for Energy Research (IREC)

Authors:

Cristina Corchero, Miguel Cruz, Lucía Igualada, Grigorios Kotsis, Ioannis Moschos

Catalonia Institute for Energy Research (IREC)

Project co-funded by the European Commission within the Seventh Framework Programme (2013-2016)		
Dissemination level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

TRACK CHANGES

Version	Date	Description	Revised	Approved
1.0	21/07/2015	Draft tests results (presentation)	WP6 Partners	IREC
2.0	31/08/2015	Introduction Algorithm user manual Test results for day-ahead market	IREC	IREC
3.0	03/09/2015	Sensitivity assessment test included Review of all sections	IREC	IREC
4.0	19/09/2015	Final results	IREC	ALL PARTNERS

TABLE OF CONTENTS

1 Introduction..... 6

2 Distribution network tool 7

 2.1 Algorithm structure and description 7

 2.2 Installation and execution of the Algorithm..... 10

 2.2.1 Installation 10

 2.2.2 Execution 10

 2.2.3 Data Modification 11

3 Commercial Aggregator tool 15

 3.1 Structure..... 15

 3.2 Algorithm Description..... 17

 3.2.1 Commercial Aggregator (Commercial_Aggregator.m)..... 17

 3.2.2 Flexibility Forecast 17

 3.2.3 Commercial Optimal Planning 18

 3.2.4 Results Presentation..... 19

 3.3 Installation and execution of the Algorithm..... 19

 3.3.1 Installation 19

 3.3.2 Execution 20

 3.3.3 Trash Files 21

 3.3.4 Parameter Modification 21

 3.4 Genetic Algorithm for Optimal Incentive Policy 23

4 Tests..... 23

 4.1 Aggregator Flexibility Forecast: Price Incentives Sensitivity Assessment 23

 4.2 Example of CA results for 2015 and 2020 scenarios 32

LIST OF FIGURES

Figure 1-1: Illustration of the interconnection between Distribution Network and Commercial Aggregator tools..... 6

Figure 2-1: 'MainNetwork.m' file structure..... 7

Figure 2-2: Example of the necessary input data into the 'MainNetwork.m' file 8

Figure 2-3: Call of 'MainNetworks62.m' File 9

Figure 2-4: 'consumerData' matrix..... 10

Figure 2-5: User should provide the month and started hour of the simulation..... 10

Figure 2-6: t_1 and t_2 parameters defined in 'createNetwork.m' file 12

Figure 2-7: 'loadProfiles' matrix..... 13

Figure 2-8: 'TripTi' and 'TripTf' matrices..... 14

Figure 2-9: Solar power profiles data are loaded 15

Figure 3-1: Commercial Aggregator tool structure 16

Figure 3-2: CA algorithm execution screen 20

Figure 4-1: Cluster 1 - Base load profile 26

Figure 4-2: Cluster 2 – Base load profile (no EVs) 26

Figure 4-3: Cluster 1 – Low price incentive 27

Figure 4-4: Cluster 1 – high price incentive 28

Figure 4-5: Cluster 2 –high price incentive 29

Figure 4-6: Cluster 1 SOC – high price incentive..... 30

Figure 4-7: Sensitivity to starting time – low price. The color map correspond to the consumption (kW).... 31

Figure 4-8: Sensitivity to starting time – high price. The color map correspond to the consumption (kW)... 31

Figure 4-9: Scenario 1 (2015) distribution network topology and clusters of prosumers 35

Figure 4-10: Scenario 2 (2020) distribution network topology and clusters of prosumers 36

Figure 4-11: Load re-profiling for Cluster 2.1 38

Figure 4-12: Flexibility offered from each cluster for scenarios 1 and 2..... 39

Figure 4-13: CA daily profit and daliy savings from prosumers..... 40

LIST OF TABLES

Table 2-1: Description of input data for 'MainNetworks62' 9

Table 2-2: 'numberID' value definition 11

Table 2-3: Algorithm output files names 11

Table 2-4: Elements capacities in 'clustering.m' file 12

Table 2-5: Weibull distributions data manipulation 14

Table 3-1: Variables to be calculated 18

Table 4-1: Request matrix template 25

Table 4-2: Price incentives for the different energy purchased levels (€) 27

Table 4-3: Inputs for the Distribution network tool 32

Table 4-4: Inputs for node configuration 33

Table 4-5: Distribution network characteristics 34

Table 4-6: Results of the clustering process for Scenario 1 (2015) 34

Table 4-7: Results of the clustering process for Scenario 2 (2020) 35

Table 4-8: Incentive matrix for Scenarios 1 and 2 37

1 Introduction

This document covers Part 3 of 4 of IDE4L Deliverable D6.1 “Optimal scheduling tools for day ahead operation and intraday adjustment”. Its main objective is to provide a detailed description of the algorithms developed in Deliverable 6.1 – Parts I & II in such a way that any user can install and use them, just by following the instruction steps.

The document is organized as follows. Section 2 deals with the “Distribution Network Tool”, based on D6.1 Part I. This tool aims at the generation of test distribution networks using as an input a set of general parameters such as the area to be served, number of nodes, penetration of renewable energy sources, or number of users owning an electric vehicle. Network nodes obtained are clustered afterwards taking into consideration their similarities. Outputs from this tool are mapped in such a way that they can be used as an input by a Commercial Aggregator.

In Section 3, the Commercial Aggregator Tool (from D6.1 Part II) is described. The purpose of this tool is to simulate and present the operation of a Commercial Aggregator and its influence on the consumers. The tool is composed of a set of functions aiming at solving the different problems to be faced by a Commercial Aggregator when participating in electricity markets. By analyzing the results, interesting economic and technical conclusions can be reached.

These tools have been implemented using a combination of MatLab and GAMS software. The whole structure of the developed algorithm is illustrated in Figure 1-1. Main objective of these tools is to use them in IDE4L Task 6.4, where the Commercial Aggregator concept is simulated and emulated.

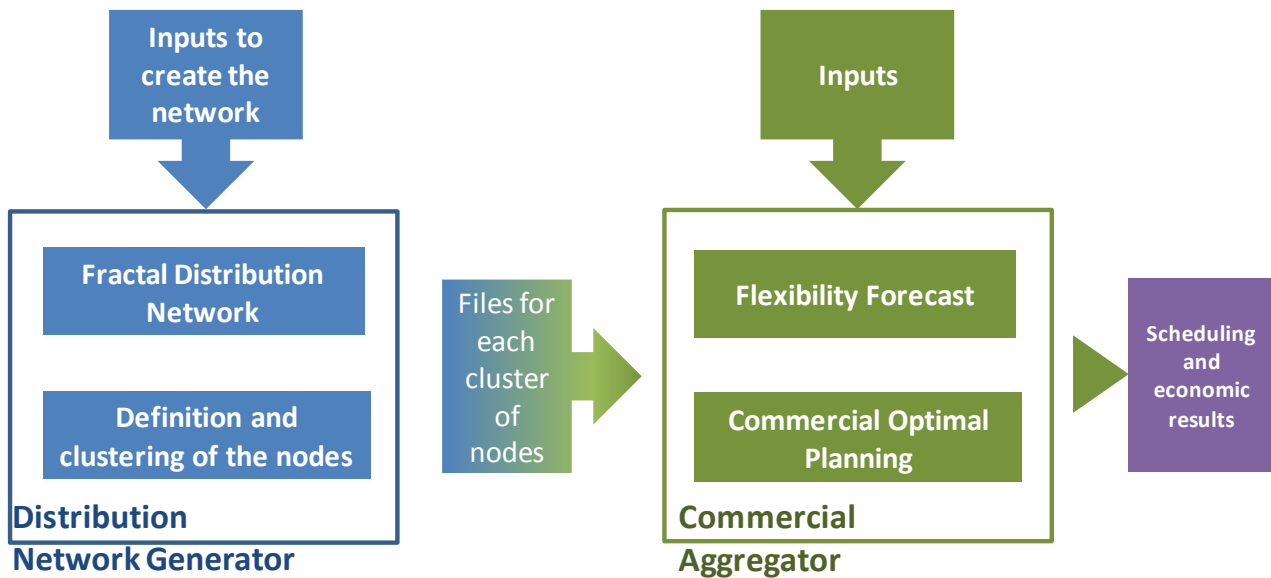


Figure 1-1: Illustration of the interconnection between Distribution Network and Commercial Aggregator tools.

Section 4 illustrates paradigm cases of how these tools can be used for different purposes. Preliminary results on the CA performance under different scenarios are provided. Further testing involving a wider set distribution networks covering different flexibility penetration scenarios will be carried out in IDE4L Task 6.4 – Deliverable D6.3 (due in May 2016).

2 Distribution network tool

2.1 Algorithm structure and description

The Distribution Network Tool is implemented using MatLab. It comprises of several files implemented in MatLab that generate a low voltage distribution network, the design of the configuration of every node in the network, and a clustering of all of them. Analytically, the main files are:

• MainNetworks.m	Main File
• createNetwork.m	Fractal and Clustering Algorithm Files
• fractalLayout.m	
• netConnect.m	
• clustering.m	
• substationLocationS.m	
• outputForAggregator.m	Interconnection Files with the Commercial Aggregator
• createOutputFileForFexibleTable.m	Commercial Aggregator

The interconnection files are used to pass data profiles from this tool to the Commercial Aggregator Tool.

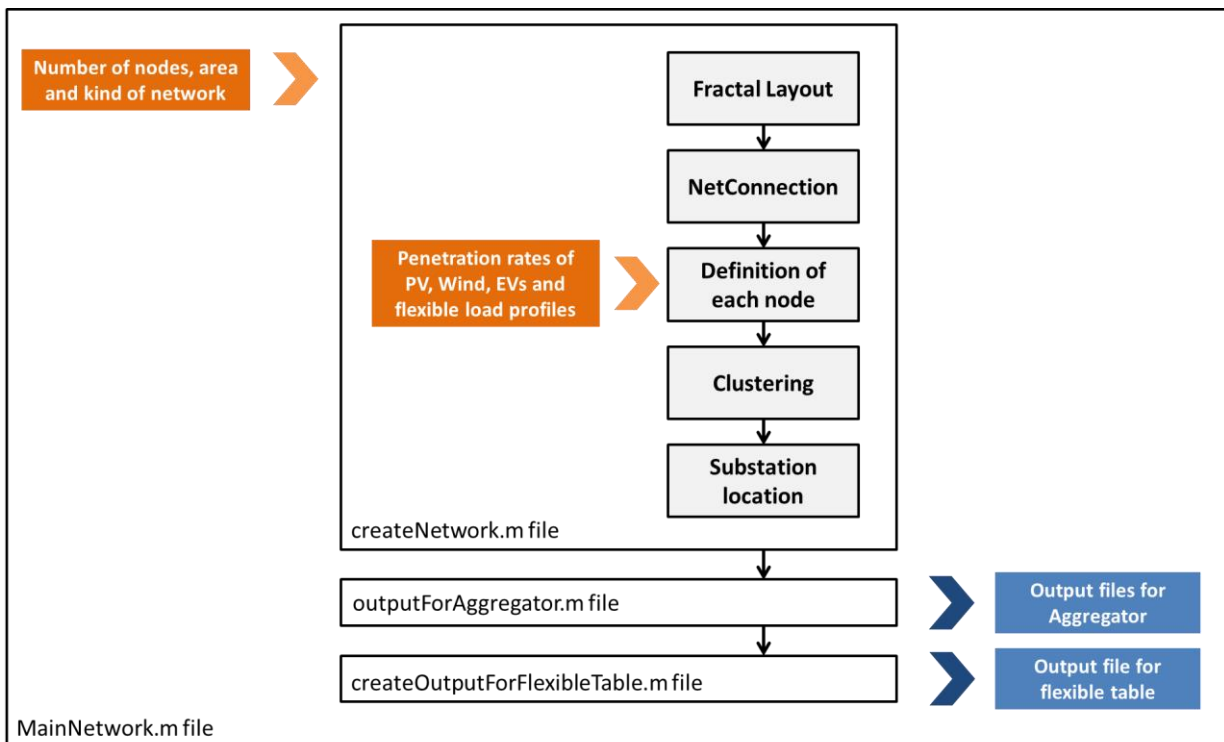


Figure 2-1: 'MainNetwork.m' file structure

First, the *MainNetworks.m* file (Figure 2-1) is executed in MatLab. In this file all the necessary inputs for generating the network are defined.

```

7      %%
8      %INPUTS for LV network%
9      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10     %km^2
11     - areaSize =0.043;
12     %For the same number of consumer, the recommended area is
13     %Urban : 0.4 km^2
14     %Semi: between 1.5 and 7 km^2
15     %Rural: 19.5 km^2
16     %number of consumers
17     - n = 40;
18     %percentage of RNN algorithm (BR) (must be between 0 and 1 and for LV
19     %network it is better values close to 1)
20     - p = 0.8;
21
22     %number of substations
23     - s = 1;
24     %type of area: Rural (R), Urban (U) or Intermediate (I)
25     - aKind = 'U';

```

Figure 2-2: Example of the necessary input data into the 'MainNetwork.m' file

There are some comments in the code in order to clarify the magnitude of the parameters. The values recommended for the *areaSize* variable are based on examples for 1000 consumers ($n = 1000$). For that reason, the value taken for this variable in the example shown in Figure 2-2 is lower than the recommended value.

- ! The parameter 'number of substations' can only take the value 1 in this version of the tool as agreed.
- ! RNN is the Random Neighbour Next algorithm where the nodes are randomly connected. The remaining nodes ($1 - p$) are connected using the Nearest Neighbour Next (NNN) algorithm. See D6.1 Part I for more details.

The *createNetworks.m* file is called to execute the fractal methodology and clustering of the nodes. The outputs from both the *fractalLayout.m* and *netConnect.m* files are used as inputs for the *clustering.m* file. In addition, this function requires other parameters in order to design the configuration of each node before doing the clustering. These parameters are entered by the user in the command windows as it is shown in Figure 2-3:


```

Command Window
>> MainNetworks62

frac_dim =

    1.1238

Enter the value of domestic rate: 1
Enter the value of solar rate: 0.2
Enter the value of wind rate: 0.05
Enter the value of EV penetration rate: 0.15
fx Enter the rate of prosumers: |
    
```

Figure 2-3: Call of 'MainNetworks62.m' File

A description for each new input is included in Table 2-1.

Table 2-1: Description of input data for 'MainNetworks62'

Domestic rate	Rate of the total nodes will be defined as domestic users. The remained nodes will be commercial users.	[0,1]
Solar rate	To calculate the total solar generation capacity, it is considered a ratio of the total load in the network.	[0,1]
Wind rate	To calculate the total wind generation capacity, it is considered a ratio of the total load in the network.	[0,1]
EV rate	Rate of the consumers with an electric vehicle	[0,1]
Prosumers rate	Rate of consumers with flexible load profiles	[0,1]

- ! The user must always enter the value 1 for the domestic rate in this version of the tool, to be improved in further versions.

The results are stored in a matrix in MatLab called *consumerData* (Figure 2-4) with dimensions (*number of consumers*)x11. In column 4, the type of consumer can be Domestic (D) or Commercial (C). From column 5 to column 8, the numbers in kW have to be positives. For the 9th column, the value 0 means that the consumer does not include an EV and the value 1 is otherwise. In column 10 it is similar, the value 0 means the consumer does not have a shiftable load and the value 1 is for a consumer with a flexible load profile. The consumers with ID cluster value equal to 0 means that these consumers are not users for the Commercial Aggregator because they do not have any flexibility to offer.

ID	coord. X	coord. Y	Type of Consumer	Peak Load	Contracted Power	Solar Power	Wind Power	EV	Shiftable Load	ID cluster
Integer	R+	R+	D/C	kW	kW	kW	kW	0/1	0/1	{0,1,2,3,4,5}

Figure 2-4: 'consumerData' matrix

Once the load for every node has been defined, the *substationLocationS.m* file is called to decide the location of the substation at the low voltage network.

In the second stage of the procedure, the *outputForAggregator.m* is called. This file transforms the information stored in the matrix *consumerData* (capacities and characteristics per consumers and per element) into files of text with profiles for 24 hours per cluster required by the CA Tool. For doing that, the algorithm asks by the Command Window the month and the started time of the simulation of the Commercial Aggregator (Figure 2-5).

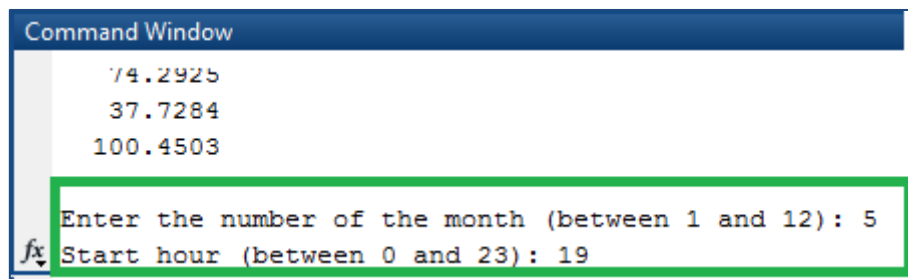


Figure 2-5: User should provide the month and started hour of the simulation

- ! Both wind and solar forecast functions are able to generate different profiles depending on the time of the year. Consequently, the final output files strongly depend on the selected month.

Finally, the *createOutputForFlexibleTable.m* file is called in order to generate and to save the information regarding the buses, branches and generator node. The output file has the extension *.raw*.

In another extended version of this tool, there is another extra step where a higher voltage level network is designed. For this version, the use of a test network is recommended (e.g. CIGRE or IEEE benchmark models). See D6.1 Part I for more details on this networks.

2.2 Installation and execution of the Algorithm

2.2.1 Installation

The algorithm only requires MatLab 2012b or newer. Only one step must be followed before the tool is ready to be executed; the path of the folder which contains all files must be set in Matlab.

2.2.2 Execution

The execution of the Algorithm starts by running the *MainNetworks.m* file. This file then automatically calls everything else and if no problem arises it saves the results in *.txt* files. Sometimes, the *k-means algorithm* fails because a cluster is empty. The solution is to repeat the execution.

By default, the picture of the distribution network shows the ID of each consumer. To remove this option, the value of the parameter *numberID* in the file *fractalLayout.m* has to be changed (Table 2-2).

Table 2-2: 'numberID' value definition

Value of <i>numberID</i>	Definition
1	The number of each node is printed
≠ 1	The number of each node is not printed

Before starting with another execution, the output files have to move to another folder because the tool overwrites them in every simulation. The name of the output files are described in Table 2-3.

Table 2-3: Algorithm output files names

Capacities*.txt	File with 4 rows; 1) Contracted power, 2) Solar capacity, 3) Wind capacity and 4) 1 if this cluster has EVs and 0 otherwise
Clusters.txt	This file contains the number of users per cluster
Critica*.txt	Load profile
Dev*.txt	Demand profile of an EV
Figure1.png	Picture of the distribution network
fractalNetwork.raw	File to calculate the flexible table
Ppv*.txt	Power solar profile
SOCmin*.txt	Constraints for the state of charge of an EV
Uev*.txt	Mobility profile of the EV
Windspeed.txt	Wind speed profile

- ! Output files Critica*.txt, Dev*.txt, Ppv*.txt, SOCmin*.txt, Uev*.txt and windSpeed.txt have 96 rows. Each row corresponds with a quarter of hour (24h x 4 1/4h = 96 time steps.)
- ! Only the data in capacities*.txt and windspeed.txt files are generated using statistical procedures. The remaining profiles are based on input data pre-defined by the user.

2.2.3 Data Modification

The baseline data can be modified in order to generate different scenarios.

2.2.3.1 Regarding the network

The main parameters of the fractal models, t_1 and t_2 , can be changed in *createNetwork.m* file. These parameters take different values depending on the kind of area selected in the first step ('U' urban, 'R' rural

or 'I' intermediated). The reference for the value of these parameters is based on examples for 1000 consumers. So, in our case, we can try other values whose results (form of the network) could be more adequate for us. An adequate value for t_1 in a small network can be 0.1.

```

1 function [consumerData, lines, Tjunctions, coordSubstation,h] = createNetwork(area
2 %INPUTS BR corresponds to the percentage of RNN. In LV is low.
3
4 switch areaKind
5     case 'U'
6         t1=0.1;
7         t2 = 1;
8     case 'R'
9         t1=0.02;
10        t2 = 270;
11     case 'I'
12        t1=0.015;
13        t2 = 30;
14 end

```

Figure 2-6: t_1 and t_2 parameters defined in 'createNetwork.m' file

- ! The values of t_1 and t_2 directly affect the distance among the nodes; the lower parameter values, the lower distances among consumers are. See D6.1 Part 1 for more details.

2.2.3.2 Regarding the configuration of the nodes

2.2.3.2.1 Capacities of the elements

The capacity of each element allocated to a node is defined by a random distribution: $r * rand() + min$. The parameters r and min can be changed in the file *clustering.m* per each element.

Table 2-4: Elements capacities in 'clustering.m' file

Solar capacity	PVmin	1
	PVr	2
Wind capacity	WindMin	1
	Windr	2
Domestic load peak	pDr	7
	pDmin	2
Commercial load peak	pCr	30
	pCmin	10

- ! The parameters PVmin and WindMin can be modified by the algorithm if there is any infactibility between these parameter values and the penetration rates defined previously.

2.2.3.2.2 Contracted Power

The regulated values of contracted power correspond with the Spanish values (taking into account both single-phase and three-phase installations). These values are defined in the file *contractedPowerSpain.m*

2.2.3.3 Regarding with the finals profiles of 24 hours

2.2.3.3.1 Load profiles

The load profiles are written in the matrix *loadProfiles* (Figure 2-7) within the *outputForAggregator.m* file. The matrix has 24 rows and as many columns as the load profiles taken into consideration. The following example contains three different profiles in kW.

```

1 function outputForAggregator(consumerData, numIntervals)
2
3 %% PARAMETERS
4 SOCmin = 0.2;
5
6 loadProfiles = [4.040103509    3.825899775    4.024818101 ;
7                4.265437914    3.325625977    3.847267195 ;
8                4.855349695    3.313885447    4.074156881 ;
9                2.520945906    3.857986778    2.50059331  ;
10               1.39097872    2.573515039    0.962318741 ;
11               0.593023245    1.332856624    1.169654463 ;
12               0.836790559    0.890127903    0.794823351 ;
13               0.87428417    0.658786307    0.495068791 ;
14               0.664817911    0.328166409    0.173418399 ;
15               0.678605072    0.32455309    0.450571736 ;
16               0.613227791    0.652885968    0.493641117 ;
17               0.447888272    0.127304293    0.472848169 ;
18               0.471007421    0.558780743    0.360716187 ;
19               0.33565895    0.38009937    0.533981528 ;
20               1.383111538    0.959955229    1.199974883 ;
21               3.078682332    3.037595154    3.051025408 ;
22               3.556323205    3.118777338    3.404587879 ;
23               3.535124902    3.286693052    3.499187153 ;
24               3.671541516    3.590081299    3.392889131 ;
25               3.626980262    3.619071521    3.44067146  ;
26               3.772110367    3.439922255    3.273682565 ;
27               3.771532079    3.481634051    3.634520513 ;
28               3.947939441    3.553054412    3.352775395 ;
29               3.981287725    3.647656405    3.325198924 ] ;
30

```

Figure 2-7: 'loadProfiles' matrix

2.2.3.3.2 Mobility profiles for EVs

The mobility profiles are defined by two matrixes: one for the initial time step of each trip (*tripTi*), and another one for the final time step of these trips (*tripTf*). They can be changed in *outputForAggregator.m* file.

Figure 2-8 shows an example of these two matrices. The number of rows corresponds with the number of trips per day while the number of columns must match the number of load profiles, it means, the number of columns of the matrix *loadProfiles*.

```

28 |                                     3.947939441    3.553054412    3.352775395 ;
29 |                                     3.981287725    3.647656405    3.325198924 ];
30 |
31 | -   tripTi = [ 1 49 1 ;
32 |              42 84 42 ;
33 |              89 97 89];
34 | -   tripTf = [ 4 78 4;
35 |              81 91 81;
36 |              96 97 96];

```

Figure 2-8: 'TripTi' and 'TripTf' matrices

- ! If there is some mobility profile with fewer trips per day (the second one in the example), a number greater than the number of time steps should be put, as it is shown within the red circle.

2.2.3.3.3 Wind speed data

The *windSpeedForecastinf.m* file contains the tables and parameters needed to generate the Weibull distributions depending on the month and hour.

Table 2-5 describes the parameters that must be changed in order to create new Weibull distributions.

Table 2-5: Weibull distributions data manipulation

Vectors S1, S2, S3, S4	Each of these vectors contains the month of the years associated with each season.
Vectors d1, d2, d3, d4 (1x24)	For every season, there is a vector which defines the night hours and the daytime hours.
Vector d (1x12)	Contains the number of wind directions taken into account
F*1 (1x12)	Frequency vector per each wind direction during the night hours.
F*0 (1x12)	Frequency vector per each wind direction during the daytime hours.
Matrix prmtrW (96x5)	This table contains per each season, wind direction and type of hour, the two parameters (shape and scale) needed to define a Weibull distribution.

2.2.3.3.4 Solar power profiles

In the *inputSolarData.m* file, the solar power profiles are loaded from the matrix *sol_madrid.mat* (Figure 2-9).

```

1  function [pSolar] = inputSolarData(month, hour)
2  %13/07/2015
3  %%Adaptación de los datos solares.
4  %Archivo original sol_madrid.mat
5  %Nos quedamos con 100 años y hacemos un día medio por mes.
6
7  %Inputs in W, Outputs in kW
8
9  load 'sol_madrid.mat';
10

```

Figure 2-9: Solar power profiles data are loaded

To use other profiles, one could load another file changing the line 9 (the new file must be in the same folder) or you can rewrite the matrix *sol_madrid.mat*.

3 Commercial Aggregator tool

3.1 Structure

The Commercial Aggregator Tool is implemented using both MatLab and GAMS software. It comprises of several files implemented in MatLab or GAMS that simulate the operation of a CA. Analytically, the files are:

• Commercial_Aggregator.m	Core Algorithm File (MatLab)	
• Flexibility_forecast.gms	Optimization File (GAMS)	Main Algorithm Files
• Commercial_Optimal_Planning.gms	Optimization File (GAMS)	
• Flexibility_forecast.m	Function	Interconnection Files
• Commercial_Optimal_Planning.m	Function	
• Plots_for_Economic_analysis.m		Plotting Files
• Plots_for_Load_Reprofiling.m		
• Load_data.m	Function	Ancillary file
• WP6_Data.mat	Saved Data	
• Incentives.mat	Saved Data	Ancillary file
○ Incentives_Scenario_1.mat		
○ Incentives_Scenario_2.mat		
• Incentive_fill.m	File (Matlab)	Ancillary file
• Disptable.m	File (Matlab)	Ancillary file

Besides the Main Algorithm files, there are some interconnection files, used to pass some parameters from MatLab to GAMS and vice versa, as well as some files that illustrate the results of the simulations by plotting several graphs.

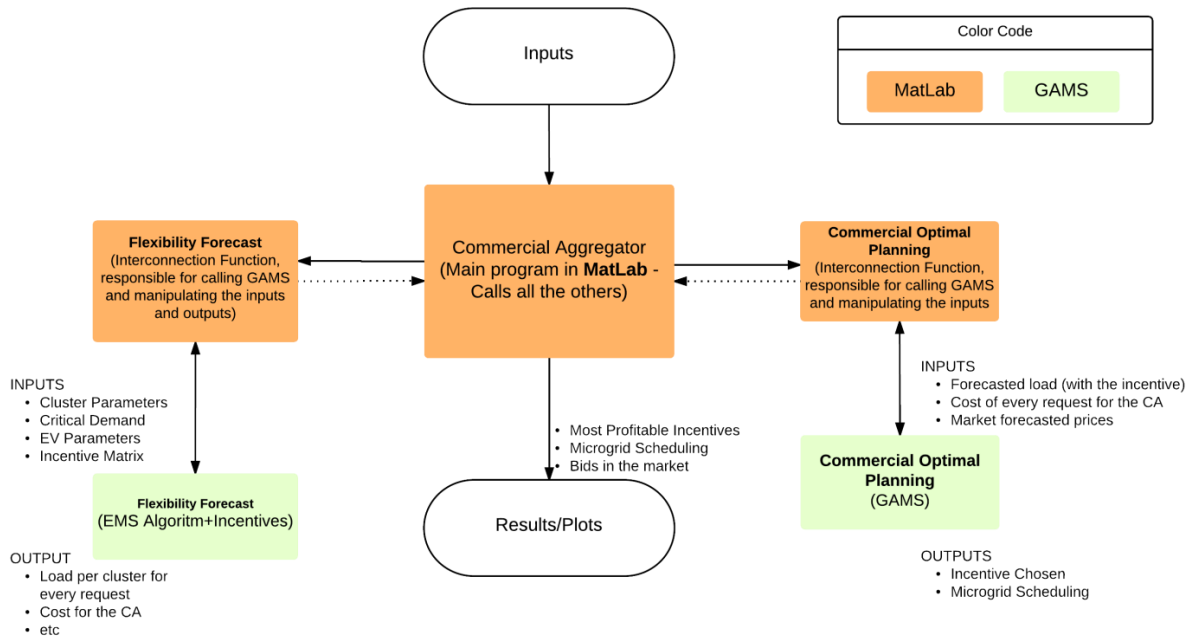


Figure 3-1: Commercial Aggregator tool structure

First, the Commercial_Aggregator.m file is executed in MatLab. This file loads all the necessary inputs of the case to be simulated. Then the flexibility_forecast.gms file is called to be executed in GAMS. The interconnection between the MatLab and GAMS software is achieved via the flexibility_forecast.m file, which contains the variables to be transferred between the 2 programs. This procedure is repeated for each one of the clusters separately and the results are stored in matrices in MatLab.

In the second stage of the simulation, the Commercial_optimal_planning.gms is called. This file contains the optimization algorithm that aims to maximize the profits of the CA, by choosing the optimal portfolio of incentives and flexibility bids to the market. Again the interconnection between MatLab and GAMS is achieved via the Commercial_Optimal_Planning.m file.

Finally, the results are stored in matrices, according to commands in the Commercial_Aggregator.m.

Various commands for illustrating the results can be found in the files: Plots_for_Economic_analysis.m, Plots_Load_Reprofiling.m

3.2 Algorithm Description

3.2.1 Commercial Aggregator (Commercial_Aggregator.m)

This is the core file of the CA Algorithm. It contains the commands for loading all the data and initializing the variables, as well as calling the optimization functions in GAMS and handling and storing the calculated results.

The inputs are loaded directly from the “Current Folder” of Matlab, by using the following commands:

- Load_data function: Loads the data files that have been automatically created by the random fractal network generator
- Load WP6_Data.mat: Loads some the flexibility table limits, the prices (spot and retail) and the incentive matrix for every cluster. The incentive matrix contains the starting-stopping time, the price and the corresponding consumption levels for every different incentive policy.
The incentive matrices are modified in order to investigate different behaviors of the consumers.

3.2.2 Flexibility Forecast

This function consists of an optimization algorithm that is run in GAMS and a MatLab file that acts as an interconnection between GAMS and MatLab.

The GAMS file (flexibility_forecast.gms) contains all the parameters of the microgrid of the prosumer¹. Parameters like EV battery size, maximum EV instantaneous power, or charges for grid interconnection, etc. can be found and changed here in the beginning of the file.

The flexibility_forecast.m file is only an interconnection file, thus it must only be changed if more or fewer variables must be transferred from MatLab to GAMS and vice versa. The variables that are exported from GAMS are also reported in the first line of the GAMS file, while the ones that are input, are shown in the line that starts with \$load.

Each execution of the flexibility_forecast.gms gives the values for 1 average user of each cluster. Thus these values must be multiplied by the number of consumers per cluster. This is done in the main file (Commercial_Aggregator.m)

¹ Prosumer and consumer denote the same physical entity in IDE4L project context

3.2.3 Commercial Optimal Planning

Like the Flexibility Forecast function, the Commercial Optimal Planning function consists of an optimization algorithm that is run in GAMS and a MatLab file that acts as an interconnection between GAMS and MatLab.

The GAMS file (Commercial_Optimal_Planning.gms) contains all the parameters that modify the bid formulation can be found and changed in the beginning of the file.

The Commercial_Optimal_Planning.m file is only an interconnection file, thus it must only be changed if more or fewer variables must be transferred from MatLab to GAMS and vice versa. The variables that are exported from GAMS are also reported in the first line of the GAMS file, while the ones that are input, are shown in the line that starts with \$load.

It is important to mention that there is the restriction in the GAMS file that an incentive policy (request) does not only have to be profitable for the CA, but also to trigger **a specific bill reduction to the consumer**, in order to be selected by the Commercial_Optimal_Planning.gms. This bill reduction is set by default to 10% but can be modified. More information about editing this can be found in chapter 4.

The most important variables that are calculated are shown in Table 3-1:

Table 3-1: Variables to be calculated

Variable	Sets	Description
Load_profile	(t, h, k)	It corresponds to the power that is purchased by the grid for each cluster and for each different request
Sold_power	(t, h, k)	Power sold to the grid
Incent_cost	(h, k)	Total cost for the CA from the incentives only
Wind_p	(t, h, k)	Wind production
Photov_p	(t, h, k)	PV production
	(t, 1:2, h, k)	Shiftable load schedule. There are 2 loads calculated separately. They are aggregated in the variable: total_shift
Char_p	(t, h, k)	EV charging schedule
Disch_p	(t, h, k)	EV discharging schedule
SOC	(t, h, k)	EV State of Charge
Und_crit	(t, h, k)	Undeliverable critical load (It must be 0)

flexi	(t, h, k)	Flexibility (difference of consumption compared to the base case)
Flexi_pos	(t, h, k)	Positive values of flexi corresponding to energy surplus. This can be sold in the wholesale market
CA_revenues, CA_revenues2	(h, k)	CA income
CA_costs	(h, k)	Total costs for the CA
CA_profit	(h, k)	Total Profit for the CA
CA_profit_diff	(h, k)	Difference between the initial profit (when no incentives are applied) for the CA and the current request's profit
Bill	(h, k)	Bill for the consumer per request (the incentive is not subtracted by the bill)
Bill_inc	(h, k)	Bill – the incentive is subtracted
Bill_diff_percentage	(h, k)	Bill reduction compared to the base case (%)
Selected_requests	(t, h, k)	Contains the request per cluster that has been chosen by the Commercial_Optimal_Planning.gms
Total_profit	-	Total profit of the CA by the selected requests

3.2.4 Results Presentation

The files **Plots_for_Economic_analysis** and **Plots_Load_Reprofiling** contain various commands for data illustration. The first file focuses in the economic analysis of the actions of the CA, while the second focuses in the re-profiling that is stimulated by applying different incentive policies.

3.3 Installation and execution of the Algorithm

3.3.1 Installation

The CA algorithm requires MatLab and GAMS 23.9.5 or newer (both programs must be either version 64 or 32 bits).

Some simple steps must be followed before the CA Algorithm is ready to be executed.

1. The necessary **paths must be set in Matlab**. This can be done by pressing the “set path” button in the main MatLab window. Then the following folders must be added:

- a. The **GAMS folder** (the one containing all the program files)
 - b. The `~\Documents\gamsdir\projdir` (it is usually in the User\Documents file)
 - c. The folder containing all the CA Algorithm files. This folder must be included with its subfolders
2. All the CA Algorithm files (GAMS, MatLab, data, and txt files) must be in the same folder which must be set as the current folder in MatLab.

3.3.2 Execution

The execution of the Algorithm starts by **running the Commercial_Aggregator.m** file. This file then automatically calls everything else and if no problem arises; it saves the results in matrices. Then using the plotting files, the results can be illustrated. It is recommended to run the plot files section by section and not using the run button in order to get the specific plot each time.

1. Put the *.txt files containing the parameters of the Fractal networks algorithm in a specific folder. This folder must be added in the **“set path”** of Matlab. This folder is usually contained in the folder where all the MatLab files for the CA algorithm are.
2. Set the Incentives.mat file that contains the incentive policies. This can be done either manually, or by using the incentive_fill file.
3. Open the Commercial_Aggregator.m file. The parameters for the number of clusters and the bill_percentage_diff are set here. The command `load Incentives` must have the correct filename used.
4. Run the Commercial_Aggregator.m file. If no error exists the screen would display: Processing cluster: 1, 2 etc.
5. If everything works well the final screen would be (Figure 3-2):

```

Command Window
Processing Cluster: 1
Elapsed time is 30.819407 seconds.
Processing Cluster: 2
Elapsed time is 50.981904 seconds.
Processing Cluster: 3
Elapsed time is 24.856627 seconds.
Processing Cluster: 4
Elapsed time is 28.964570 seconds.
Processing Cluster: 5
Elapsed time is 24.989872 seconds.
Commercial Optimal Planning Works
Elapsed time is 100.878348 seconds.

                Cluster 1  Cluster 2  Cluster 3  Cluster 4  Cluster 5
Selected Request                11      12      15      65      47
Incentive Start (Timeslot)      10      11      14      64      46
Incentive Price (Euros)         0.03    0.12    0.21    0.44    0.26
Positive Flexibility (kWh)      6.51    14.41   4.266   1.389   24.68
Bill Difference (%)             -11.34  -10.61  -5.115  -5.161  -5.99
Profit for the CA (Euros)       0.7401  2.126   3.845   1.428   16.73
Change in CAs profit (%)        43.33   10.42  -22.43  -24.75  -27.55

The total Profit of the CA is: 24.8655
    
```

Figure 3-2: CA algorithm execution screen

3.3.3 Trash Files

During each execution of the CA Algorithm several trash files may be created. These are usually automatically deleted, but if an error occurs they may appear in the folder of the CA Algorithm.

For keeping the number of the files of the algorithm low, the following files could be deleted if they appear in the CA Algorithm folder:

- Incdat.txt, incsol.txt, matdata.txt, opersol.txt, operdat.txt
- Flexibility_Forecast.lst, Commercial_Optimal_Planning.lst
- Matdata.gms
- GA_initials.mat, GA_param.mat GA_selected_results.mat

3.3.4 Parameter Modification

Most of the clusters specific parameters are set by the fractal network and are sent as an input via .txt files to the algorithm. So, these parameters like grid tie, critical load, EV schedule etc. can only be set using the fractal network algorithm or manually.

In case the data needs to be modified, then it can be done in the following files:

Incentives*.mat (for instance: Incentives_Scenario_1.mat)

- Incent*: These matrices (*=1,2,3,4,5 for cluster 1,2,3,4,5 respectively) include the incentive matrix for each cluster. The only way to change the incentive policies is here.
 - Column 1: incentive policy number
 - Column 2-3: Starting – stopping time of the incentive
 - Column 4-8: Price of the incentive, per consumption level
 - Column 9-13: Consumption levels

ATTENTION!!

- ! The matrices for the Incentives (Incent*) are saved in the Incentives*.mat. Thus they must be changed within this file each time, and then the file must be saved as a *.mat file. The filename must be changed in the load command in the Commercial_Aggregator.m file
- ! It is one of the parameters that must be modified very often, since every new cluster reacts differently to different incentives. Thus, many different incentive policies (combinations of different starting times and different prices per level of consumption) could be used in a big matrix.
- ! Another approach could be to run the Genetic Algorithm for every cluster –there is an automated procedure to run the GA for 5 clusters automatically-. The Genetic Algorithm gives the Optimal price and starting time per cluster.
This optimal values can then be used to fill the Incent* variables. For instance the optimal price can be filled for all the incentives polices, while only changing the starting time.
- ! In order to fill the whole matrix with the same price levels for all the requests of a cluster, the file Incentive_fill.m may be used.

GAMS file flexibility_forecast.gms

- All Parameters regarding the operation of the microgrid can be changed. These parameters include battery capacity, charging limitation, etc., which are considered to be the same for all the consumers.

WP6_Data.mat

- P_retail: The retail price of the electricity can be changed here
- P_spot: The spot price of the electricity
- Shift_prof: The profile of the Shiftable load (it is the same for all the consumers)
- Flexibility_table: The limitations of the flexibility table can be set here (no limitations by Default)
- CRP: The flexibility allocated to CRP is set here, this flexibility is considered Already sold so it is not used by the Commercial Optimal Planning tool. It is 0 by default.
- Cluster_risk: This is 100000 (negligible) by default
- Cluster_reliability: This is 1 (negligible) by default.

In order to change the file:

1. Load `WP6_Data.mat`
2. Change the parameters
3. Save `WP6_Data.mat`

Commercial_Aggregator.m

- Clusters: This is the number of clusters (5 by default). It can only be changed if the files produced by the fractal network algorithm are changed accordingly. So if we decide to add clusters, then we have to add more txt files.
- Bill_diff_per_lim: This is the minimum triggered reduction in the consumers bill in order for a policy to be accepted by the CA. This parameter is used by the Commercial Optimal Planning tool in order to select the optimal incentive policies.

3.3.4.1 Load_data.m

This file reads the *.txt outputs from the Fractal Network Algorithm and creates the matrices that contain the respecting data to be used in MatLab.

- At this time the wind power is not read, so by the time we obtain the data for the wind, a new command for reading the wind txt files must be added.
- Attention must be paid to the GAMS file in case of a change in the wind power input. The reason is that as it is, GAMS has the wind speed as an input from MatLab and then calculates the wind power. If the wind power is given directly by the .txt files, then some equations (they are already marked in GAMS file) must be skipped.

3.4 Genetic Algorithm for Optimal Incentive Policy

There is a group of files related with the Genetic Algorithm heuristic method. This method is used in order to find the best incentive policies (different combinations of starting time of the incentive and different incentive prices). This tool is automated and gives the final best combinations saved in the matrix 'selected_results'.

The procedure to be followed is:

1. **Edit the GA3_objective_function.m** (If Needed)

This file contains the objective function. The optimization problem is in fact defined in here. Every change in this file changes the optimization results.

 - The variable to be minimized (e.g. CA_profit) can be modified here along with the restrictions.

2. **Edit the GA2_Genetic_Algorithm.m** (If Needed)

All the parameters of the Genetic Algorithm parameters are found here. These parameters can influence the precision and speed of the algorithm. The Help file in MatLab describes analytically how this can be used.

There are 2 methods that we can choose from:

 - **Method 1:** divides the time horizon in several sectors and then finds the optimum value for the incentives in each sector. This method can be used to find more than one optimal prices. For each interval of starting times a different optimal price is calculated. Thus the CA can build a portfolio of incentive policies to trigger demand response thorough the time horizon, while maximizing its profit.
 - **Method 2:** finds one optimum incentive for the whole time horizon. Only one optimal price is calculated that corresponds to the maximum profit for the CA.

3. **Run the GA1_call_algorithm.m** (Core Program- All the functions are called by this one.)

This file contains some initialization commands and then calls the GA file. At the end of the simulation, the results are saved by the commands used here.

 - **The cluster to be analyzed** is chosen here by changing the respecting variable.
 - **The minimum desirable bill reduction** to the consumers' bill can be set here. It is measured in (%).
 - **The method** to be used in GA_2_Genetic_algorithm.m
4. The results are saved in the matrix selected_results (incentive starting time, incentive price, profit)

4 Tests

4.1 Aggregator Flexibility Forecast: Price Incentives Sensitivity Assessment

In this example the core Aggregator's function "Flexibility Forecasting" is tested. The objective of this function is to forecast the load profile of every group of consumers (cluster) before and after the application of a price incentive signal.

Step 1: Case study description

The microgrid consists of demand and generation components that essentially enable the DR flexibility of a household. More specifically, the consumption devices can be divided in three parts:

- **Critical load:** Refers to these devices which demand should always be satisfied and is uncontrollable.
- **Shiftable load:** These devices need to run consecutively for some time since they are operated, thus their usage can be transferred to a future time point. In this study case, the end-user flexibility in demand shifting is considered unlimited.
- **Adjustable load:** Usually, these refer to thermal-electric devices, like air-conditioning, heat pumps, central heating. These devices can be controlled in some extent, without a significant loss for the end-user comfort.

The Distributed Energy Resources (DER) from which the microgrid is comprised are:

- **Electric Vehicles (EVs):** EVs are a crucial source of flexibility in the system, since they are assumed to work bi-directionally (V2G concept). Important aspects for the simulation are the EV batteries maximum and minimum State of Charge (SOC), as well as the driving pattern considered for each of them.
- **Renewable Energy Resources:** A photovoltaic array and a small wind turbine are also considered.

Step 2: Request Matrix

The objective of Flexibility Forecast module, as stated above, is to predict prosumers' flexibility, under a specific price incentive signal that is sent by the CA to their Energy Management System (EMS). The idea is to model the EMS optimization algorithm as seen from the Aggregator's perspective, implement the price incentive signal and finally obtain the load pattern changes. Every cluster is represented by an average prosumer. The EMS is simulated independently for each cluster and the results are then multiplied by the number of prosumers in the cluster.

Therefore, the purpose of the Aggregator is to apply price incentive signals to the prosumers according to their energy consumption (purchased power from the grid); the price incentives are linked to pre-defined energy consumption levels. Evidently, there is a correlation between incentive I , consumption level Q and time step t . Combining these three factors many possibilities can arise, i.e. different policies would be created. That is the reason for creating a matrix of different combinations, called Request Matrix (Table 4-1). Each combination in it would be referred as a request (H). The 1st request in the Request Matrix will always refer to the baseline scenario, where no incentive is sent to the household. For each time step during a request time interval, the EMS algorithm can choose between any $I - Q$ combination.

Table 4-1: Request matrix template

Request (H)	Start time	Stop time	Incentive (I)	Consumption level (Q)
1	-	-	-	-
2	-	-	-	-
		...		
N	-	-	-	-

Step 3: Preparation of the EMS simulation per cluster

As described in D6.1 Part II section 4, each prosumer is assumed to have a local EMS which is supposed to minimize its energy costs. In order to predict prosumers’ response to the different requests, their EMS should be modeled. For doing so, a generic EMS algorithm is assumed for each cluster, which is modeled within CA Flexibility Forecast tool by means of an optimization algorithm. This optimization algorithm mimics the EMS installed in the prosumers’ premises. By running the algorithm for different incentive policies, the respective load pattern changes are simulated. Thus, the Aggregator forecasts the available flexibility and can later build their portfolio.

Step 4: Results

Several simulations are carried out, in order to assess the efficacy of the Flexibility Forecasting algorithm and calculate the load modification. All the simulations are performed for a 1-day period, divided in 96 timesteps of 15 minutes. For the sake of simplicity, the model is only tested on the following two clusters:

Cluster 1: A group of 100 residential prosumers. Each one employs the microgrid model previously described. For a more realistic approach, the EVs are not considered to be connected during the whole day, but each one follows a different schedule of connection-disconnection, which represents a typical daily usage. Each EV contributes with its stored energy to the microgrid only during the periods that it is connected. The load profile for the base load, i.e. when no incentive is applied by the Aggregator, is illustrated in Figure 4-1.

The loads that contribute to the flexibility of the system are the *shiftable load* (grey) and the *EVs* (blue). In this case these form a great part of the total consumption.

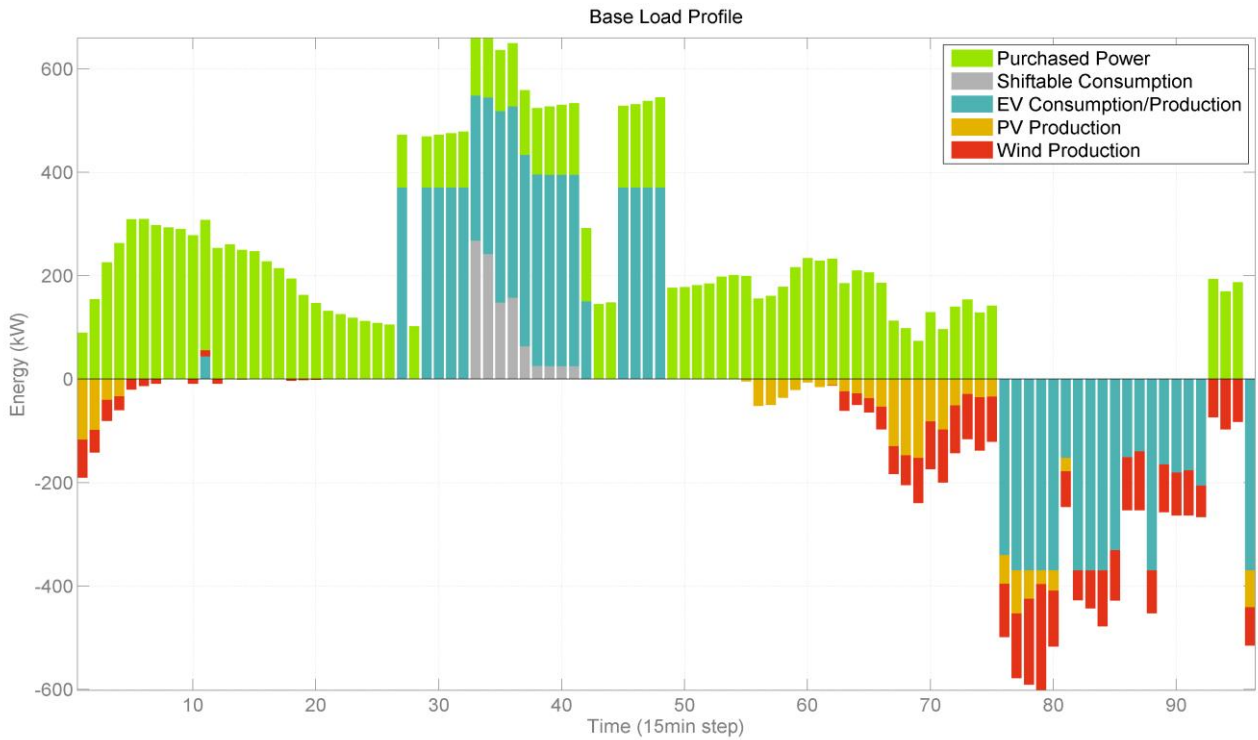


Figure 4-1: Cluster 1 - Base load profile

Cluster 2: This cluster is identical to the cluster 1, without considering the EVs. Hence, the flexibility in this case stems only from the shiftable loads. The base load profile of cluster 2 is shown in Figure 4-2.

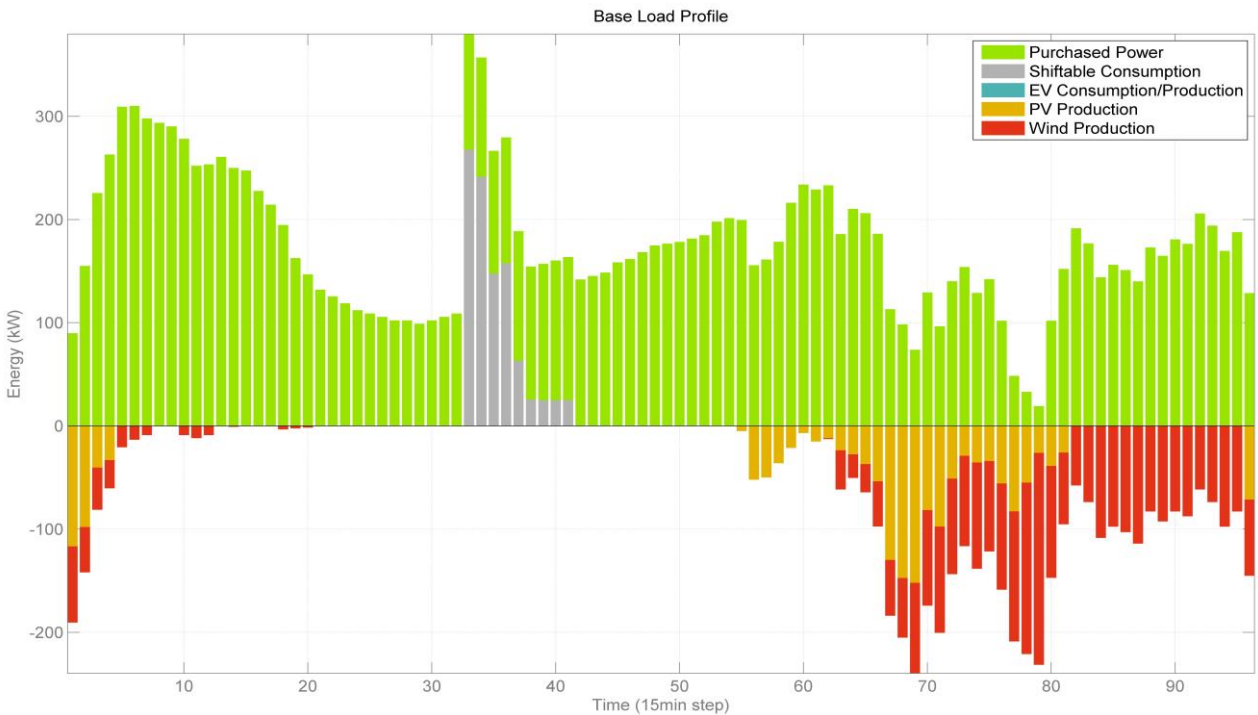


Figure 4-2: Cluster 2 – Base load profile (no EVs)

Load Modification

The application of a suitable incentive, sent by the Aggregator, triggers a change in the consumption pattern of each one of the prosumers. Depending on the incentive policy used, different load profiles occur to form a great variety of Demand Response products. The incentive policies are diverse by means of the price that is offered to the prosumer, during the period of interest, or the starting time step of the incentive.

Two different price incentives (linked to five consumption levels) are tested during the simulations and are presented in Table 4-2. The consumption levels 'Q' correspond to 15%, 35%, 40%, 50% of the Q_i (maximum grid tie capacity). The low price incentive is rather unrealistic, but is used as an indication of the algorithm's effectiveness.

Table 4-2: Price incentives for the different energy purchased levels (€)

	$Q < 0.15 * Q_i$	$Q < 0.25 * Q_i$	$Q < 0.35 * Q_i$	$Q < 0.40 * Q_i$	$Q < 0.50 * Q_i$
Low	0.002	0.001	0.0005	0.00025	0.000125
High	0.1	0.05	0.025	0.0125	0.00625

The following figures show the results of different incentive policies in terms of starting time, in comparison with the base load. Figure 4-3 shows the reaction of Cluster 1 to the low price incentive, for 2 different starting points of the incentive.

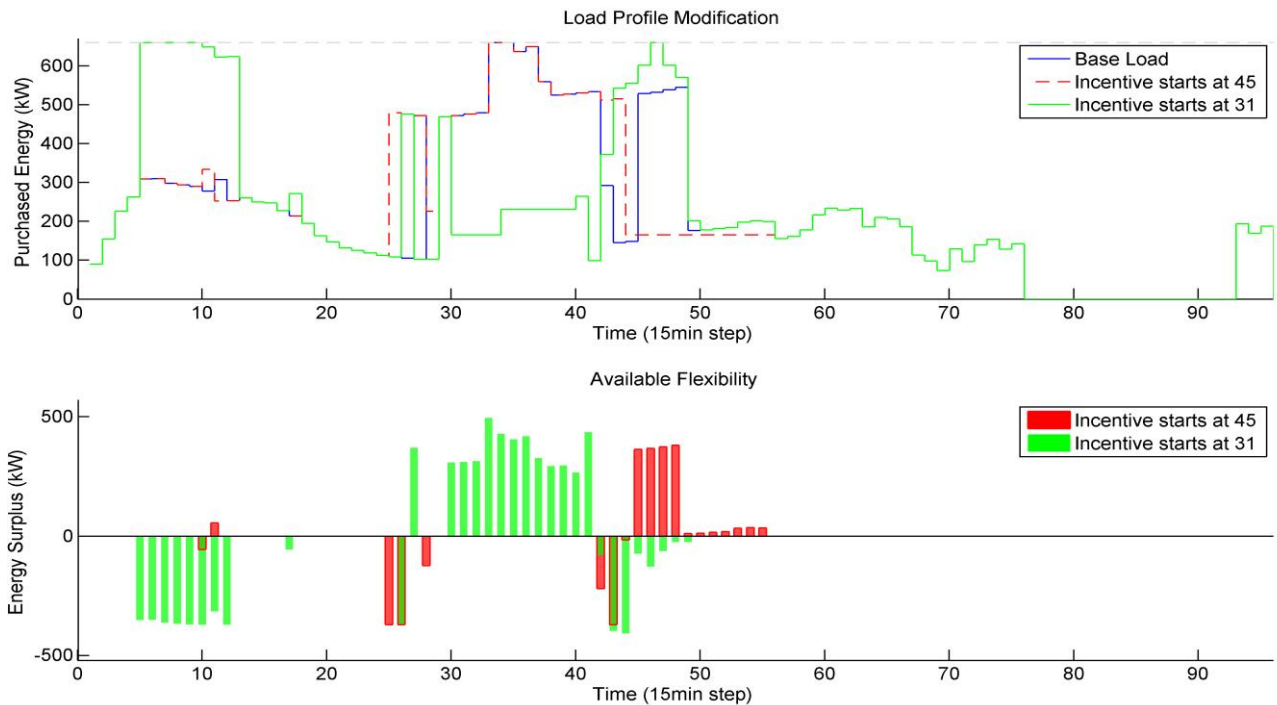


Figure 4-3: Cluster 1 – Low price incentive

Figure 4-4 shows how the load profile changes when applying a higher price incentive.

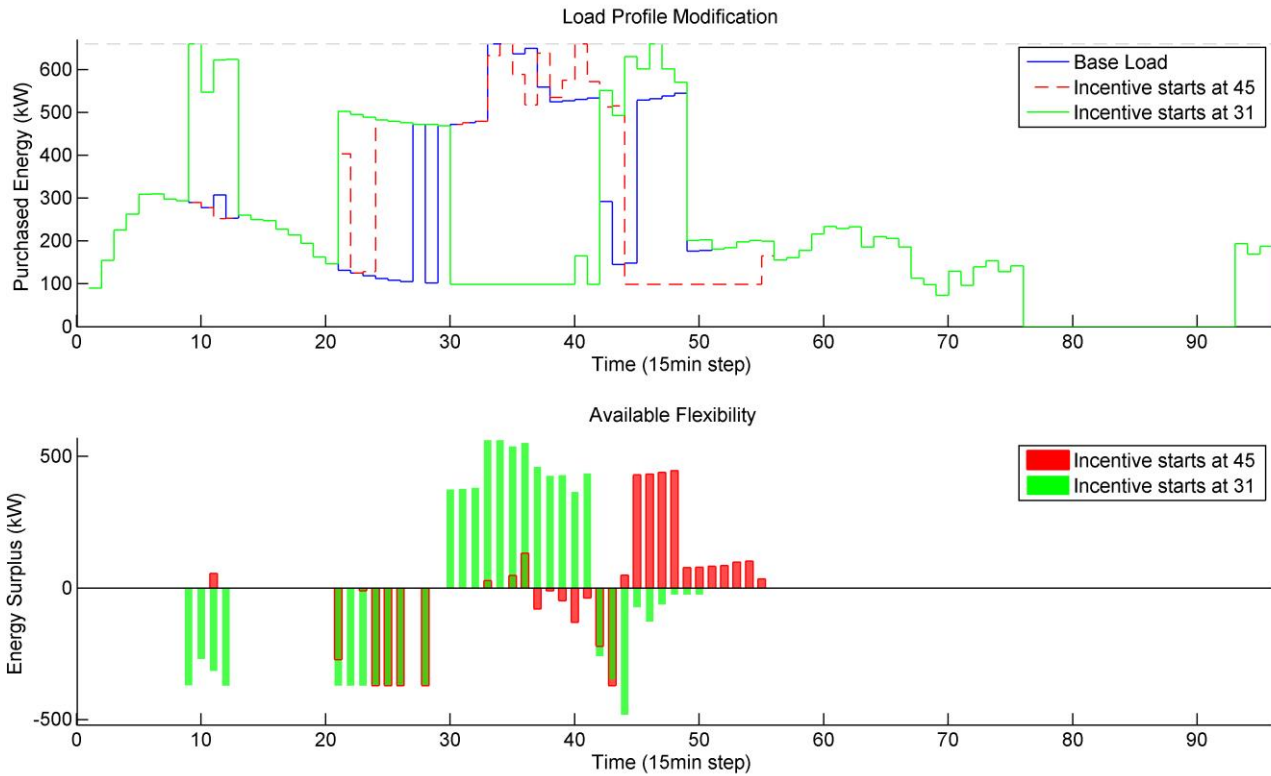


Figure 4-4: Cluster 1 – high price incentive

In both figures it is obvious that the incentive triggers a time shifting of the peak, causing a reduction of the consumption during the period of interest. It is important to observe that applying the incentive in different periods, leads to completely different load profiles, thus different patterns of flexibility.

Figure 4-5 illustrates the response of Cluster 2 for two selected incentives and for high price incentive policy. The less available flexibility is due to the less shiftable load (there are no EVs). The load profile corresponding to request 25 is shifted through time. However, a pay-back effect can be observed for the second incentive (red line); this is an unwanted effect in reality, as it may cause violations to grid constraints.

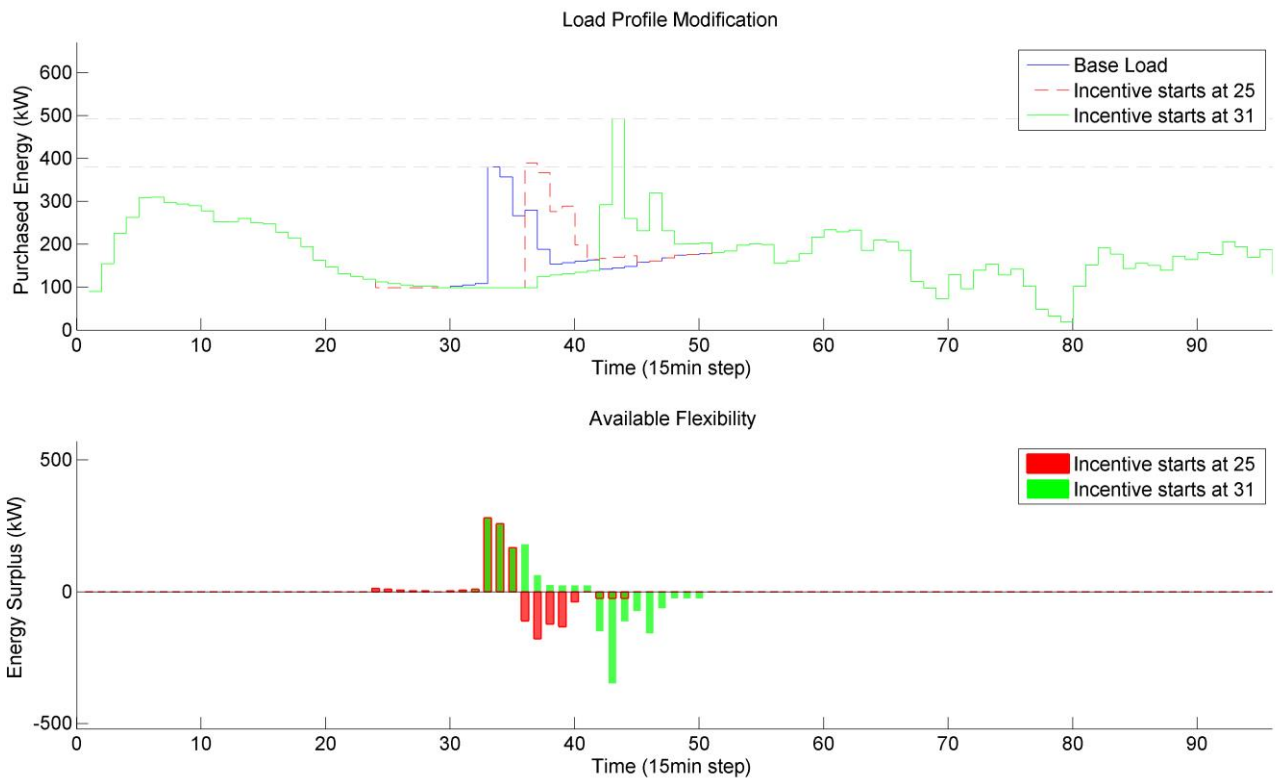


Figure 4-5: Cluster 2 –high price incentive

State of Charge of the EVs

Since the major part of the available flexibility comes from the EVs, it is of a great importance to check if the re-profiling interferes with the requirements of the end user. During the simulation it is assumed that the user sets some values for the minimum acceptable SOC of the EVs’ battery, in order to reserve the available energy for the trip. This requirement, along with the SOC of the EVs for 3 different requests (base load, or 2 different incentives) is shown at Figure 4-6 for the high price incentive. It is interesting to see that for Request 45, during the application of the incentive, part of the energy is served by the EV2 battery (Vehicle - to - Grid effect).

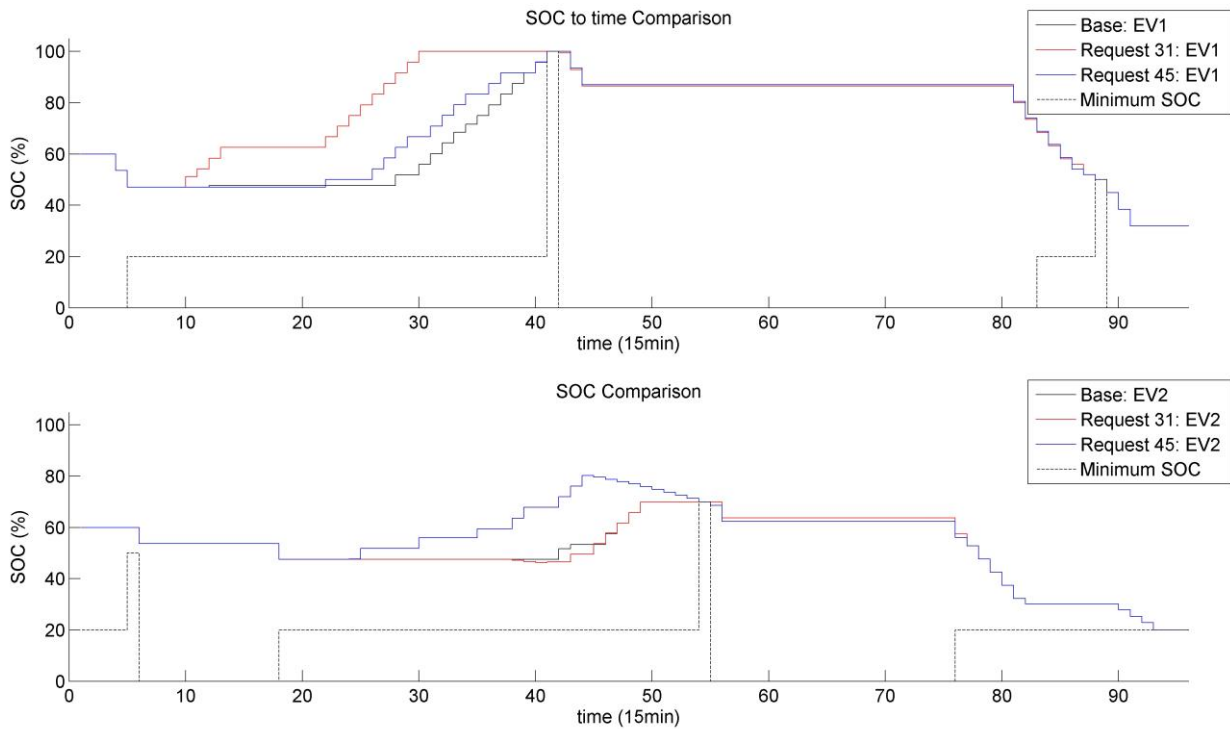


Figure 4-6: Cluster 1 SOC – high price incentive

Step 5: Sensitivity Analysis

A sensitivity analysis is performed in order to investigate the different responses of the EMS, to the different incentives. Getting a clearer picture of the way that the incentives affect the algorithm helps the Aggregator to optimize their portfolio of bids. This analysis is focused on the *time variation* of the application of the incentive. The reason is that the variation of price has a rather proportional correlation with the microgrid's load profile.

The variable of this sensitivity analysis is the period when the incentive is applied. The duration of the incentives is 3 hours. 86 different requests are tested. Apart from the first request, which corresponds to the base load (no incentive is applied), every following requests shifts the starting point of the incentive to the next timestep. For instance, in the 2nd request, the incentive starts at timestep 1, while in request 25, the incentive starts at 24. Using an iterative algorithm, the results of simulating these 86 requests are saved and illustrated in the following figures. Figure 4-7 corresponds to the low price incentive, while Figure 4-8 to the high price incentive. The x-axis of the figures shows the timeline (24 hours divided in 96 timesteps), while the y-axis shows the applied requests, which almost coincide with the starting point.

Every horizontal line of these contour graphs, corresponds to the load profile for the respective request. The diagonal grey lines indicate the periods when the incentives are applied. It is obvious that the application of the incentive forces the algorithm to reduce the consumption, during the periods of interest by shifting the peaks. The control is not direct, since no direct signals are used but only price signals, but it is very effective.

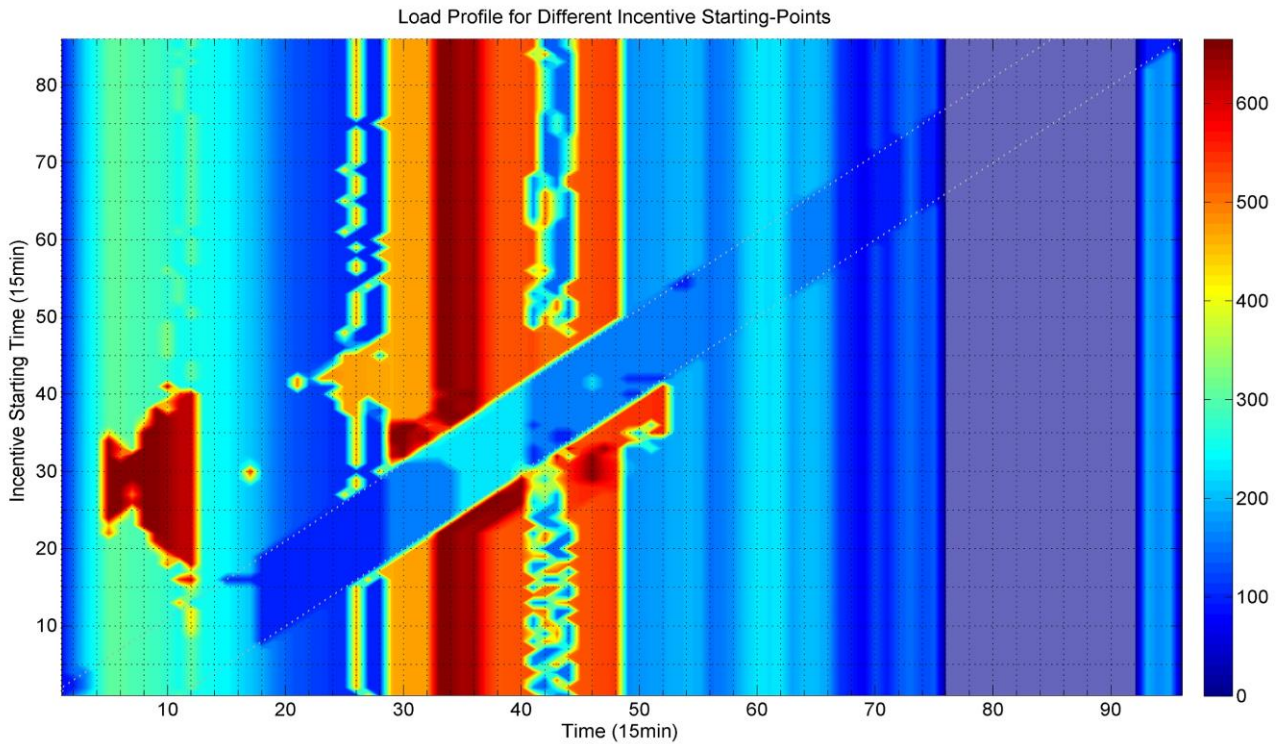


Figure 4-7: Sensitivity to starting time – low price. The color map correspond to the consumption (kW)

It is interesting to show the response of the cluster to a higher price, so the same sensitivity analysis will be performed, by only considering price changing.

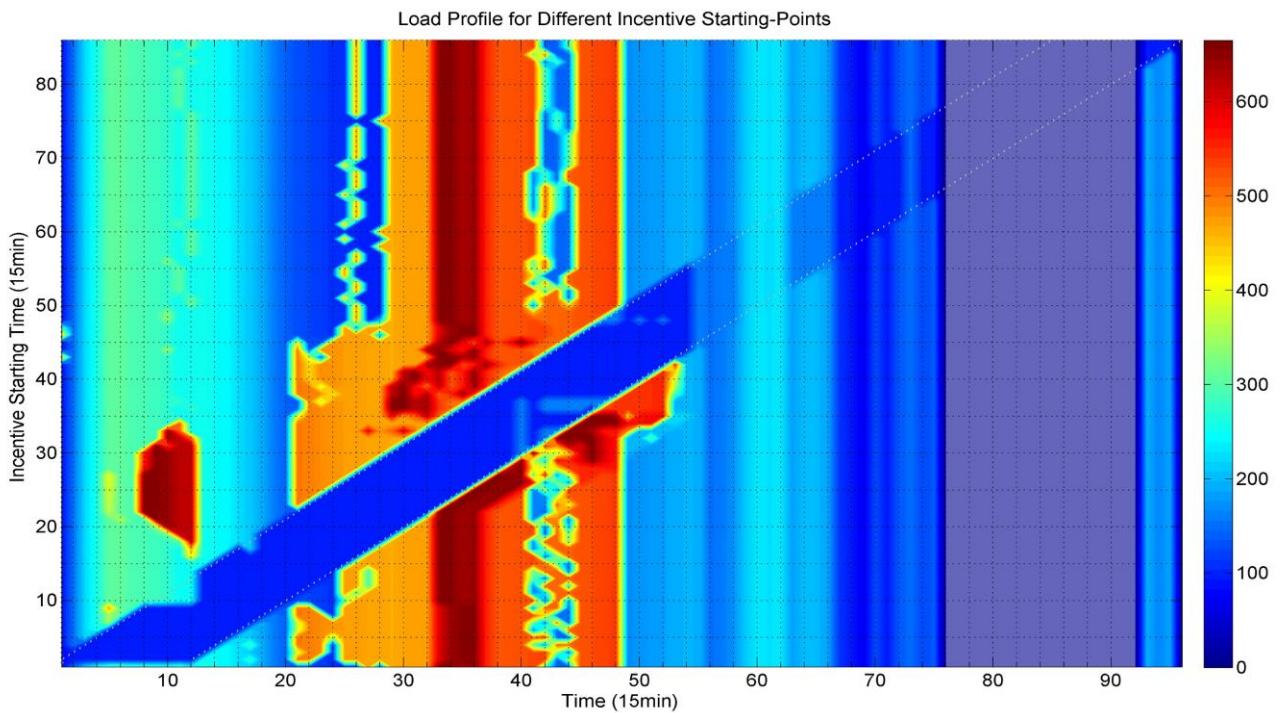


Figure 4-8: Sensitivity to starting time – high price. The color map correspond to the consumption (kW)

Figure 4-8 shows that the re-profiling is “stronger” now. This higher incentive price may not be profitable for the Aggregator when the objective is to sell the energy surplus to the market. However, this price makes sense when a contingency requires the direct and immediate re-profiling.

The 2nd cluster has a similar behavior, but the consumption is quite lower (no presence of EVs) and the flexibility is only based on the shiftable load.

Step 6: Conclusions

The simulations in this example show that the incentive is a very effective way of re-profiling the consumption of a cluster. Given the fact that the main requirements, set by the user, are satisfied, the load re-profiling comes without sacrificing the convenience of the user.

By modifying the time, when the incentive is applied, different load profiles can be obtained. Thus by observing the behavior of the EMS algorithm, subject to the incentives, the Aggregator can optimize their portfolio to meet the buyers’ needs.

The value of the Flexibility Forecasting tool is obvious; the Aggregator can utilize the available flexibility in the various electricity markets, optimizing profits and delivering real benefits in the system. TSOs and DSOs could use DR products to resolve contingencies in the network or simply avoid costly grid expansions. Other energy stakeholders could optimize their portfolio by participating in the DSM scheme.

4.2 Example of CA results for 2015 and 2020 scenarios

The aim of this example is to test two different scenarios, one based on current data and another one considering a possible futuristic scenario. These scenarios are built by the Distribution Network Generator, whose results are sent to the Commercial Aggregator in order to show its functionalities, and at the same time, the interconnection between the two tools.

Step 1: Design inputs for the Distribution Network Tool

Same general inputs for both 2015 and 2020 scenarios regarding geographical characteristics of the distribution network as well as the number of nodes and number of substations have been considered (Table 4-3). These values have been calculated taking into account European LV network for domestic users recommended by CIGRE (see D6.1 Part I).

Table 4-3: Inputs for the Distribution network tool

Kind of area:	Urban
Area size:	0,043 km ²

Number of nodes: 40

Number of substation: 1

Step 2: Inputs for node configuration

The following characteristics have been considered for 2015 and 2020 scenarios:

- Scenario 1 (2015): Scenario 1 takes conservatives values based on a distribution network of Cuellar and mobility values of Barcelona
- Scenario 2 (2020): In scenario 2 more optimistic values expected for 2020 have been considered.

Table 4-4: Inputs for node configuration

Scenario 1 (2015)		Scenario 2 (2020)	
Penetration rate of	Value	Penetration rate of	Value
Domestic Users	100%	Domestic Users	100%
Solar Generation	1.38%	Solar Generation	20%
Wind Generation	0.1%	Wind Generation	5%
Electric Vehicles	4%	Electric Vehicles	15%
Flexible load profiles	40%	Flexible load profiles	70%

Step 3: Distribution Network tool execution

Results regarding distribution network characteristics for scenarios 1 and 2 are shown in

Table 4-5. For both cases, a LV distribution network composed of 3 feeders with a total length of about 600m and 225 peak load is obtained. This brings two distribution networks that despite being different, are totally comparable.

Table 4-5: Distribution network characteristics

Scenario 1 (2015)			Scenario 2 (2020)		
	Length [km]	Load [kW]		Length [km]	Load [kW]
Feeder 1	0.133	57.911	Feeder 1	0.2000	45.134
Feeder 2	0.374	124.67	Feeder 2	0.0237	12.494
Feeder 3	0.147	44.754	Feeder 3	0.371	161.85
Total	0.654	227.335	Total	0.5947	219.478

Results of the clustering process for both scenarios are described in Table 4-6 and

Table 4-7. For both cases, prosumers connected to the distribution network are clustered as a function of their characteristics. Results shown in the tables are average values for solar, wind and contracted power. Electric vehicle availability is characterized by yes/no.

It can be seen how the amount of prosumers increases when Scenario 2 is compared to Scenario 1. While for Scenario 1 only 19 prosumers including controllable loads are obtained, for Scenario 2 this figure is increased to 34. Average values for PV and wind generation capacity is also increased, while contracted power remains practically unchanged. As a result, higher flexibility is significantly increased when Scenario 2 is compared to Scenario 1.

Table 4-6: Results of the clustering process for Scenario 1 (2015)

Average consumer (number of prosumers in cluster)	Cl. 1 (1)	Cl. 2 (1)	Cl. 3 (2)	Cl. 4 (2)	Cl. 5 (13)
Solar [kW]	0	0	1,32	0,5	0
Wind [kW]	0	0	0	1,01	0
Contracted Power [kW]	3,45	5,75	8,05	4,6	5,75

EV	Yes	Yes	No	No	No
----	-----	-----	----	----	----

Table 4-7: Results of the clustering process for Scenario 2 (2020)

Average consumer (number of prosumers in cluster)	Cl. 1 (4)	Cl. 2 (2)	Cl. 3 (8)	Cl. 4 (11)	Cl. 5 (9)
Solar [kW]	1,728	2,636	1,94	1,547	0
Wind [kW]	0,5	0	0,44	0,553	0
Contracted Power [kW]	8,05	3,45	4,6	8,05	5,75
EV	Yes	Yes	No	No	No

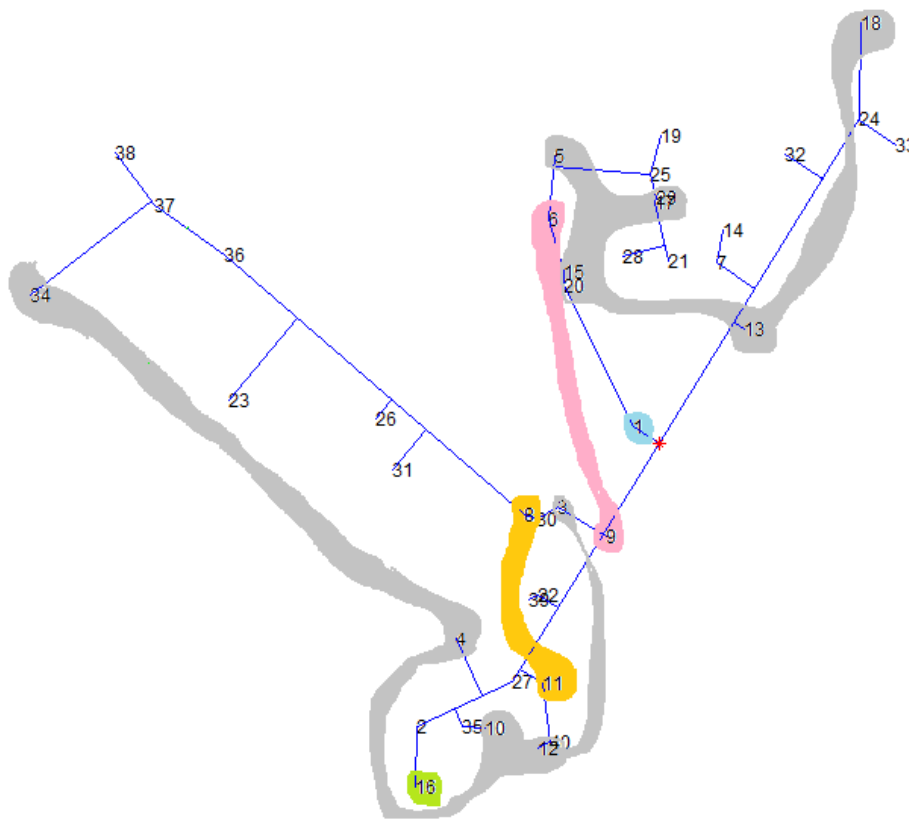


Figure 4-9: Scenario 1 (2015) distribution network topology and clusters of prosumers

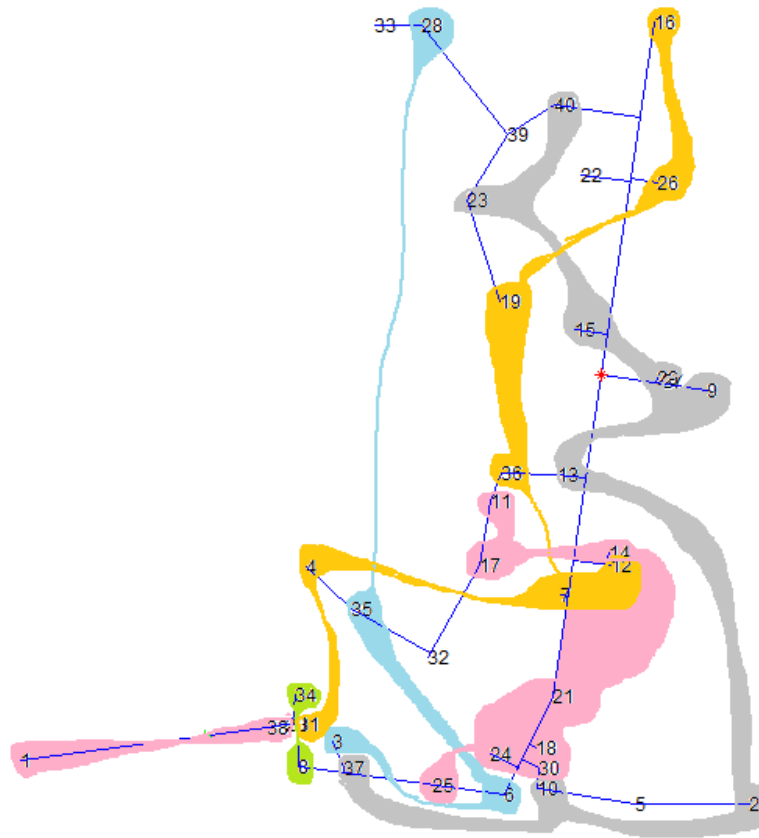


Figure 4-10: Scenario 2 (2020) distribution network topology and clusters of prosumers

In Figure 4-9 and Figure 4-10 both distribution networks are depicted. The following colors have been used for showing which cluster each prosumer belongs to. Non colored nodes do not include flexibility resources:

- Blue color: Cluster 1 (solar, wind, EV, controllable loads)
- Green color: Cluster 2 (Solar, EV, controllable loads)
- Yellow color: Cluster 3 (solar, wind, controllable loads)
- Pink color: Cluster 4 (solar, wind, controllable loads)
- Grey color: Cluster 5 (controllable loads only)

Step 4: Applied incentives per cluster

Once distribution networks are obtained and prosumers are grouped into the five aforementioned clusters, the Commercial Aggregator computes the optimal incentives to be applied to each cluster aiming at maximizing their profits in the day-ahead market. Table 4-8 shows the incentive to be sent to each cluster for Scenario 1 and Scenario 2 respectively.

It can be seen how despite the base load profile considered for both scenarios is the same, starting and stopping time for incentives is slightly moved due to the different amount of available flexibility. Also, price

levels are modified seeking for the most cost-effective combination of Commercial Aggregator costs (price incentives) versus prosumers costs (level of consumption not to be surpassed).

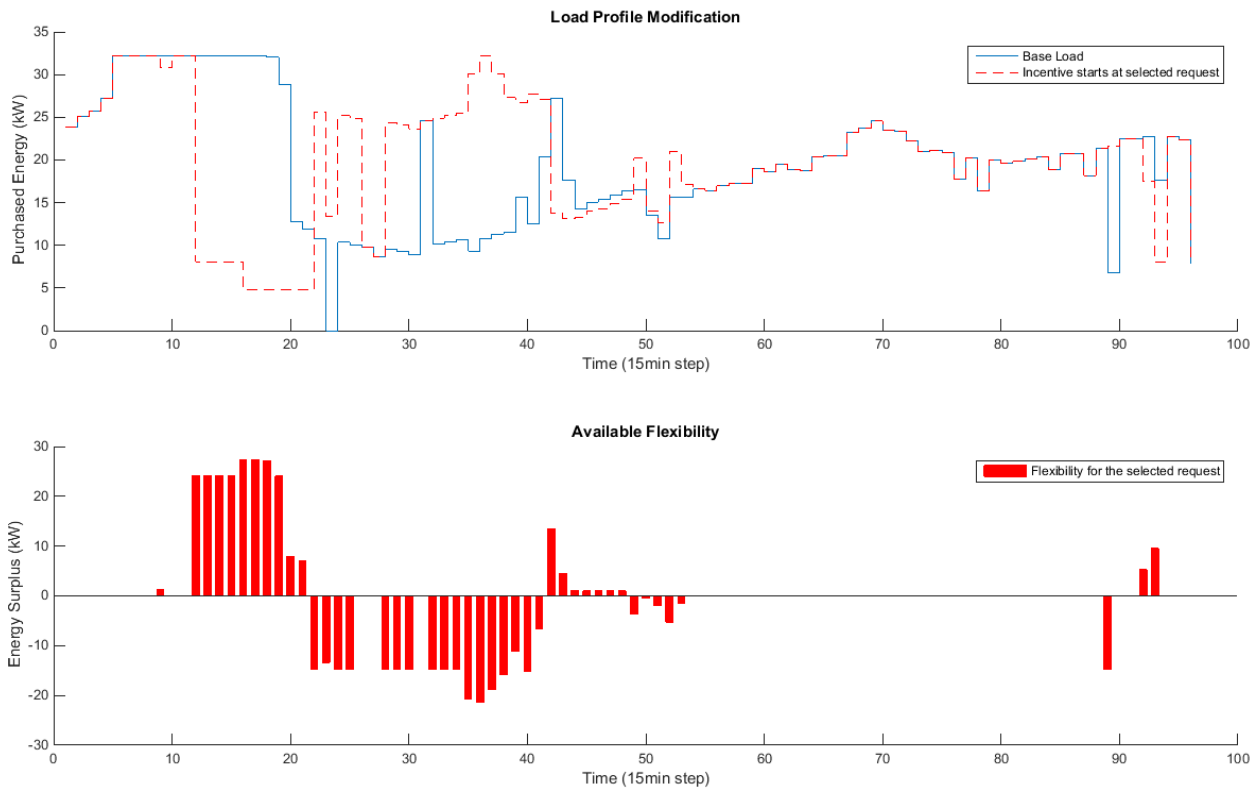
Table 4-8: Incentive matrix for Scenarios 1 and 2

Scenario 1 (2015)												
Cluster	Starting time	Stopping Time	Price Incentives [€]				Levels Of Consumption [kw]					
1.1	10	21	0,03	0,024	0,018	0,012	0,006	0,5175	0,8625	1,2075	1,38	1,725
1.2	13	24	0,11	0,088	0,066	0,044	0,022	0,8625	1,4375	2,0125	2,3	2,875
1.3	14	25	0,42	0,336	0,2552	0,168	0,084	1,2075	2,0125	2,8175	3,22	4,025
1.4	51	62	0,07	0,056	0,042	0,028	0,014	0,69	1,15	1,61	1,84	2,3
1.5	13	24	0,24	0,192	0,144	0,096	0,048	0,8625	1,4375	2,0125	2,3	2,875
Scenario 2 (2020)												
Cluster	Starting time	Stopping Time	Price Incentives [€]				Levels Of Consumption [kw]					
2.1	10	21	0,17	0,136	0,102	0,068	0,034	1,2075	2,0125	2,8175	3,22	4,025
2.2	5	16	0,05	0,04	0,03	0,02	0,01	0,5175	0,8625	1,2075	1,38	1,725
2.3	17	28	0,04	0,036	0,028	0,016	0,008	0,69	1,15	1,61	1,84	2,3
2.4	13	24	0,49	0,392	0,294	0,196	0,098	1,2075	2,0125	2,8175	3,22	4,025
2.5	15	26	0,17	0,136	0,102	0,068	0,034	0,8625	1,4375	2,0125	2,3	2,875

Step 5: Results comparison

Once incentives to be conveyed to available prosumers are selected by the CA, results obtained for both scenarios are compared.

First, an example of load re-profiling is shown for cluster 2.1 (Scenario 2 – Cluster 1, Figure 4-1).



Error! Reference source not found. The blue line represents the base load (without incentives) and red-dashed line represents the re-profiled load once the incentive is activated. It can be seen how the amount of purchased energy from the grid dramatically decreases while the price incentive (0,17€) is activated from time step #10 to time step #21. A rebound effect therefore appears once the price incentive is removed, especially from time step 22 to time step 42. Small load changes are also found during the rest of the day as a consequence of the load re-profiling process.

Total load-reprofiling for cluster 2.1 is increased from 6,51 kWh to 64,19 kWh, thus increasing the amount of flexibility to be used by the CA for portfolio optimization purposes. Similar results are obtained for the rest of clusters when base load and load re-profiling are compared. This information will be used as an input for the commercial aggregator and network operators for obtaining market results and performing grid validations respectively.

In Figure 4-11, total flexibility offered from each cluster is shown for scenarios 1 and 2. It can be seen how similarly to cluster 1, a higher amount of flexibility is offered from the rest of clusters. That is not the case for cluster 5, where total amount of flexibility is reduced in accordance with the lower number of prosumers belonging to this cluster. Specifically, flexibility is reduced from 25,48 kWh to 16,98 kWh. However, this reduction is counterweighted by the increase found in the rest of clusters, bringing up total flexibility from 53,69 kWh to 136,52 kWh, i.e. 154% higher.

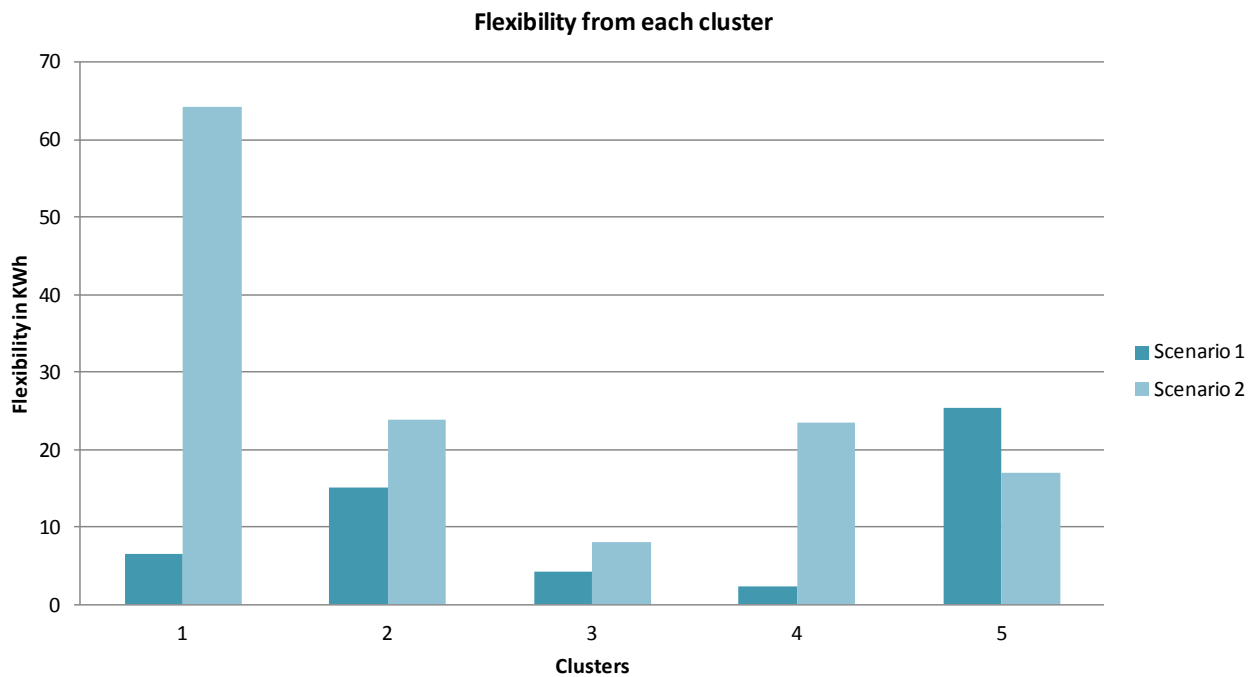


Figure 4-11: Flexibility offered from each cluster for scenarios 1 and 2

Finally, daily profits and savings for CA and prosumers respectively are shown in Figure 4-12.

From the CA perspective, it can be seen how potential profits are increased from Scenario 1 to Scenario 2. While total daily profit for Scenario 1 (2015) was 17,98 €, the figure is increased until 39.06 € for Scenario 2 (2020). Therefore, total profit increases with the amount of flexibility obtained from prosumers. However, it should be noticed that flexibility coming from different clusters show different profitability: while total increase in flexibility offered by cluster 4 is lower than the one offered from cluster 1, total profits obtained from the CA are higher.

Another effect that can be observed is the following: average profit per prosumer increases a 21,40 % from Scenario 1 to Scenario 2. While for Scenario 1 average profits are 0,94 €/user, it were 1,15 €/user for Scenario 2. From this, it can be concluded that higher penetration of RES and flexible loads lead to higher profits for CAs.

From the prosumers perspective, it can be seen how for all cases bill savings account for more than 10 %. Reason behind is that a savings threshold has been set for the CA when selecting the incentives to be activated. When scenarios 1 and 2 are compared, no major changes are observed excepting for cluster 2, where the higher amount of RES availability give more room to EV owners to shift their consumption, thus increasing the flexibility potential to be offered.

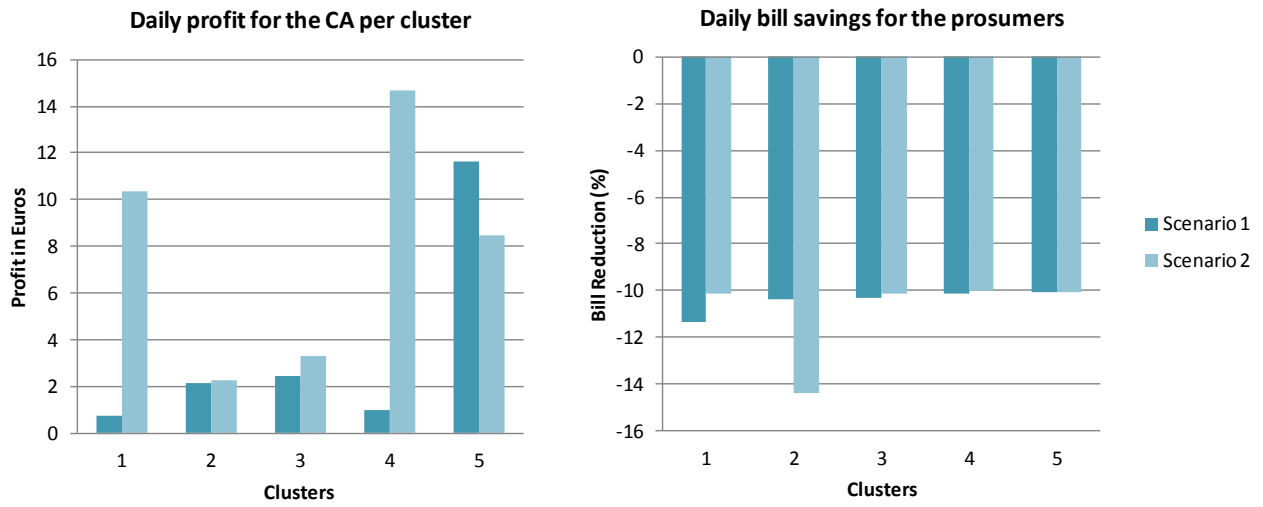


Figure 4-12: CA daily profit and daly savings from prosumers