VSHARING SYSTEM SIMULATOR

User Manual (ver. 2.0.0)

Table of Contents

1. Carsharing simulator overview

1.1.   Program installation and requirements
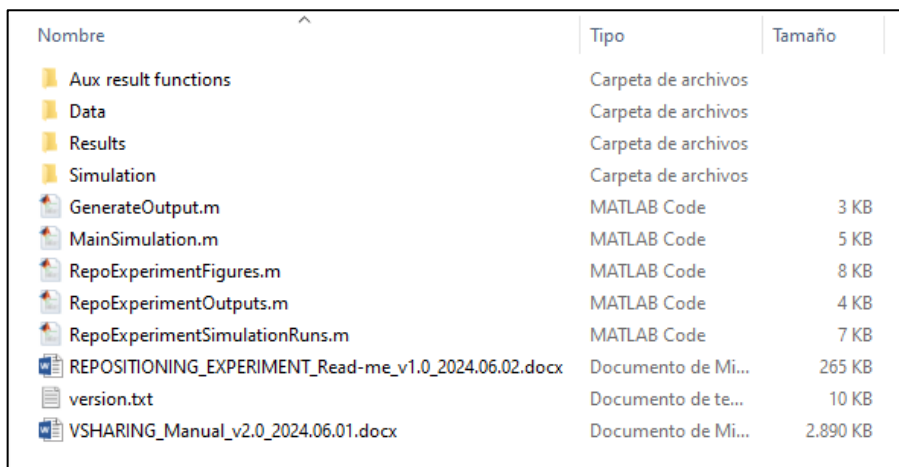
The Agent Based Vsharing Simulator is a program based in the Matlab coding language. It requires MATLAB on its version 9.10 (R2021a) or higher.

The installation of the simulator only requires to download the code files and execution is done through MATLAB.

1.2.   Program structure and files

The simulator consists of a folder (main folder, see Figure 1) with the following subfolders and files:

- **"Data" folder:** This folder contains all the input files of the simulator. Most of the inputs are mandatory, and they are preset with the default values. If the user needs to introduce new inputs, this must be done by modifying the files in this folder or subfolders. (See Section 2 for more information about the input files).
  This folder also includes a subfolder called "Aux_inputs". It includes auxiliary files to generate a zonification output and should not be removed or changed.

- **"Results" folder:** This folder stores all the generated results. If this folder does not exist, it will be created after the first simulation run. (See Section 6 for detailed information about the outputs.)

- **"Simulation" folder:** This folder stores all code files.

- **"MainSimulation" file:** Script that runs the simulation.

- **"GenerateOutput" file:** Script that runs the result post process code, generating all the results files.

- **"User Manual.docx":** This is the file of this User Manual to understand how to run the simulator.

Figure 1. Folder and file structure of the simulator.

1.3. Simulation and result generation process

The simulation process includes two blocks: i) the simulator setup and ii) the simulation time-step actions. These modules are independent inside the code. This provides additional flexibility to face future changes, without blocks interfering each other.

The generation of results is a completely independent process outside the simulator. Results are set with an independent code apart from that of the simulation. After the simulation is completed, the user must select which results wants to obtain as outputs, and run the postprocessing program. In this way, users can generate different results without running again the simulation.

To sum up, the whole process would be as follows:

1. Set all the simulation inputs in the "Data" folder. (See Section 2 for more information about the minimum necessary input files.)

2. Run the "MainSimulation" script file. When the simulation is completed, a Matlab object file with the name "YYYYMMDD_HHMMSS_Final_City_Variables.mat" will appear in the "Results" folder ("YYYYMMDD_HHMMSS" is the simulation date and time). It contains all the elements of the simulation. Inside the same folder, it will also appear a subfolder called "YYYYMMDD_HHMMSS_Final_City_Variables_input". It contains a summary of the simulation inputs.

3. Go back to the folder "Data" and open the excel file "outputs.xlsx". Select which results must be processed and returned for visualization. Be sure that the previous Matlab object file ("YYYYMMDD_HHMMSS_Final_City_Variables.mat") is correctly introduced. Save and close the file "outputs.xlsx".

4. Run the "GenerateOutput" script file. Results will be stored in a subfolder called "YYYYMMDD_HHMMSS_Final_City_Variables_output", with the same date prefix as the Matlab object file. It contains all processed results organized by type.

2. Inputs to the simulator

This section describes the inputs that are necessary to run the simulation. All the inputs must be included in the folder "Data".

The main inputs are:

- **"inputs.xlsx" MS Excel file:** This file includes all general car-sharing system parameters.

- "**stations.xlsx" MS Excel file:** This file includes a list of car-sharing stations, with their particular features.

- **The zonification:** One file defining the car-sharing service area and their operational sub-zones.

    o **Option 1: Full Zonification.** Include a shapefile zonification in .shp format. A default set of shapefiles is included as a reference. (They are contained in the folder "Data\SERVICE_AREA\WGS84_UTM_31N_ZONIFICATION".)

o **Option 2: Perimeter.** Alternatively, include a MS Excel file with the coordinates of the perimeter of the service area. According to this perimeter, the simulator will create a default zonification. A default perimeter file is included as a reference ("Data\SERVICE_AREA\boundary_UTM_31N.xlsx").

- **The "outputs.xlsx" MS Excel file:** This file is needed for selecting the desired results in the postprocess.

- **The Origin / Destination demand matrixes:** A set of demand matrices in .csv format. A set of default files are included as a reference. (They are contained in the folder "Data\DEMAND_OD\ FF_hourly_demand_0.5_perc".) These matrices are optional. If they are not included, the simulator will create them from the aggregated demand parameters.

- **The Generated Users array:** This input is optional. It must be a MATLAB array which includes a set of vehicle-sharing potential trips with its defining characteristics (i.e. starting time, origin, destination). If it's not given by the user, the simulator will generate its own random trips according to the demand patterns given. The simulator also always returns this array after a completed simulation in order to replicate experiments with the same set of potential trips.

- **The Initial Distribution of vehicles MS Excel file:** This file is not mandatory. It must be an Excel file with two columns, with the ID of each station and the initial number of vehicles on it. User must aware that the IDs are the right ones, and that the sum of all vehicles is equal to the total fleet.

Templates of all these files are included in the "Data" folder. These input files can be open and edited to change the simulation setup, but users must not change their name or path unless this possibility is clearly stated in this manual.

The following subsections explain in detail each one of the inputs.

2.1. The general inputs file ("inputs.xlsx")

The file "inputs.xlsx" is a mandatory input file that recaps all general parameters for the whole car-sharing system. It consists in an excel spreadsheet with all the necessary variables to run the simulation. Each variable appears with a small description and its units of measurement. **The values of all variables are editable, but the file "inputs.xlsx" cannot be renamed of moved from the "Data" folder**.

Next sections describe each group of input variables, and provide the default value used in the simulator for the Barcelona case study. For a detailed description of the estimation of the default values of these parameters, see the APPENDIX – Default parameter value estimation process.

2.1.1. General simulation parameters

Table 1. General simulation inputs

Parameter	Description	Default value
TotalTime (T)	Total duration of the simulation in minutes. It also defines the demand cycle. (See Section 3.2 for more information about setting the total simulation time.)	1440 min
WarmUpCycles	Number of warming up cycles in the simulation before obtaining results. This allows reducing the effects of initial conditions. (See Section 3.2 for more information about setting simulation time.)	2
verbose	Shows messages during the simulation. • *verbose = 0* => Messages disabled. Not recommended. • *verbose = 1* => Messages enabled.	1
rndSeed	Random seed code. Can be left blank and MATLAB will set one by default. (See Section 6.2.2.1 to obtain the random seed of a previous simulation.)	-

2.1.2. Service area and zonification parameters

These parameters are used to import or create the service area and the free-floating zones in the simulator. Two options are possible:

1. **Shape file →** Import a predefined shapefile with the full zonification. These can be exported from the most common GIS softwares.

2. **Perimeter →** Import a MS Excel file with the coordinates of a closed polygonal line defining the perimeter of the service area. In this case, the simulator will create a custom zonification, and will allow to export it as a shapefile.

Table 2. Service area and zonification inputs.

Parameter	Description	Default Value
ServiceArea	Selects the service area type of input to generate the zonification. • *ServiceArea = 0* => Perimeter in an excel file (.xlsx) • *ServiceArea = 1* => Full zonification in a shapefile (.shp)	1

If ServiceArea = 1		
ShapeFile	Path to shapefile. (See Section 2.3 for more information about the shapefile input format.)	See "Service area zonification" in Appendix
If ServiceArea = 0		
PerimeterFile	Path to perimeter file. (See Section 2.4 for more information about the perimeter input file format.)	See "Service area perimeter" in Appendix
OutputShape	Name of the output zonification shapefile when generated from perimeter. It will be stored automatically in the simulation inputs recap folder (see Section 6.2.2).	zonification_out.shp

Note that in case of selecting the shapefile option, only the input "ShapeFile" will be used. In contrast, if selecting the perimeter option, the "PerimeterFile" and "OutputShape" inputs will be used instead.

For more information about the definition of the service area and zonification process, see Section 4.1.

2.1.3. Demand parameters

Demand parameters are used to import or create the demand profile in the simulator. Like in the zonification process, two options are possible here:

1. **Import O/D demand matrices →** Import existing origin/destination demand matrices.

2. **Generate O/D demand matrices →** The simulator generates origin/destination demand matrices from the set of aggregated demand parameters.

Note that the first option requires the knowledge of the zonification, and therefore it is only possible if a shapefile (i.e. with zonification) was introduced.

Table 3. Demand generation inputs.

Parameter	Description	Default Value
OdmatKnown	Selects the type input of O/D demand matrices. • *OdmatKnown = 0* => Matrices not available. They will be estimated by the simulator from aggregated demand parameters. • *OdmatKnown = 1* => Matrices available.	1

If OdmatKnown = 1		
Odmat_prefix	Path to the prefix for the input O/D matrices .csv files. (See Section 2.5 for more information about the O/D matrix input file format.)	See "Demand matrices in .csv files" in Appendix
TimeReDemand (TD)	Duration for which each input demand matrix holds, in minutes.	60 minutes
If OdmatKnown = 0		
Odmat_out	Name prefix for the output O/D matrices .csv files when custom generated. They will be stored automatically in the simulation inputs recap folder (see Section 6.2.2).	matrix
TotalTripsDay (N)	Total number of potential trips, N, for the whole demand cycle, T (i.e. for the whole "TotalTime" duration) in the car-sharing system.	6481 trips[1]
ImbalanceAvg (Φ)	Average fraction of imbalanced trips, Φ. Average fraction origin trips without an equivalent return trip. Must be a value between 0 and 1.	0.226[1]
areaRet (π_t)	Fraction of service area where returned cars are higher than requested cars, π_t, resulting in a car accumulation at the end of the day.	0.47[1]
areaReq (π_q)	Fraction of service area where requested cars are higher than returned cars, π_q, resulting in a additional car requirement at the end of the day.	0.53[1]
ImbalancePattern	Average imbalance pattern within the service area • *Radial* => Imbalance is centered on a single point (focus), and evolves linearly in the radial direction from the focus. • *Flat* => Imbalance is defined by an axis (where the system is balanced) and evolves linearly in its perpendicular direction.	-
ImbCentre (X_C, Y_C)	The imbalance center (X_C, Y_C) coordinates in UTM. • If the pattern is 'radial', it defines the imbalance focus point. • If the pattern is 'flat', this input is not used. It can be left blank.	-
ImbDirection	Imbalance direction:	-

---

[1] Aggregated values obtained from the default O/D matrices inputs.

(φ)	• If the pattern is 'radial':
	
	    ○ *ImbDirection = 1* => if vehicles' attraction grows towards the focus point.
	
	    ○ *ImbDirection = -1* => if vehicles attraction diminishes towards the focus point.
	
	 • If the pattern is 'flat', this value is the positive angle between the North and the imbalance direction measured in sexagesimal degrees.	

TimeWeight ($W_{(t)}$)	Distribution of the total demand over the demand cycle. This is included as an array of weight factors. The number of elements in the array defines the number of periods, *t*, in which the demand cycle is divided. Each element is the weight of that period with respect to the others in the whole cycle. For example, an array [1;2] means that the demand cycle is divided in two periods, and in the second period the demand is the double than in the first one. (By default, an array of 24 values is proposed corresponding with the same hourly distribution of the example O/D matrices provided.)	[120; 84; 71; 60; 62; 82; 121; 275; 426; 372; 314; 322; 340; 393; 406; 386; 385; 396; 435; 425; 377; 288; 210; 131][1]
UsersKnown	Selects the trip generation input. • *UsersKnown = 0* => Trips are generated randomly according the demand patterns given. • *UsersKnown = 1* => Trips are read from a MATLAB array, which was generated on a previous simulation	0
If UsersKnown = 1		
UsersFile	Path to the trip array file. It includes a set of potential trips from a previous simulation. Note that for a correct experiment replication simulation times must also coincide (including warming-up).	-

Note that if the O/D demand matrixes are available, only the inputs "Odmat_prefix" and "TimeReDemand" are used. The rest of parameters in this group of inputs only will be used when matrices are not available.

For more information about the demand input process, see Section 4.3.

2.1.4. Stations' parameters

These group of parameters are used to import or generate the stations in the simulator. All inputs in this group are mandatory.

Table 4. Stations' inputs.

Parameter	Description	Default Value
TotalStat (S)	Total number of stations used in the car-sharing system.	48 stations
InputStationFile	Path to the "stations.xlsx" MS Excel file which includes the input stations' list. This file must exist, even if no station is to be used. In such case any default or empty file will suffice with no impact in the results. See Section 2.2 for more information about the stations' input file format.	Data/stations.xlsx (See "Station candidate list" in Appendix)
OutputStationFile	Name of the MS Excel file with the output stations' list. This list includes only the stations finally used in the simulation. It will be stored automatically in the simulation inputs recap folder (see Section 6.2.2).	stations_out.xlsx
defaultCapacity (s_m)	Default number of available parking slots (i.e. capacity) at parking stations. Must be an integer higher than 0. It is only used when the capacity value at the input file is void or for the generated stations within the simulator. Capacity values of 9999 or higher are considered as "infinite" for occupancy calculation purposes.	9999
defaultChargers (s_{m_e})	Default number of available e-vehicle charging slots at parking stations. Must be an integer smaller or equal than the default capacity. It is only used when the corresponding value at the input file is void or for the generated stations within the simulator. Values of 9999 or higher are considered as "infinity" for occupancy calculation purposes.	9999

For more information about the stations' input process, see Section 4.4.

2.1.5. Fleet parameters

These parameters are used to create the car fleet. All inputs in this group are mandatory.

Table 5. Fleet design inputs.

Parameter	Description	Def. Value
TotalCarsFF (m_{FF})	Total number of vehicles deployed initially on streets (i.e. FF => free-floating).	200 cars
TotalCarsSB (m_{SB})	Total number of vehicles deployed initially on stations (i.e. SB => station-based).	50 cars
streetFleetLimit ($\lim\_m_{FF}$)	Maximum number of vehicles that can be parked on streets (e.g. maximum allowance by municipality regulations). It must be an integer value higher or equal than "TotalCarsFF".	9999 cars
percEcars (e)	Fraction of electrical vehicles in the fleet. It must be a value between 0 and 1.	1
BatteryConsume (D_{ev})	Electrical vehicle autonomy distance for 80% of the battery level [km]	200 km
BatteryChargeTime (T_{ev})	Time required to charge 80% of the electrical vehicle battery [min]	240 min
minBatteryPerc (B_{min})	Minimum battery percentage to consider an electrical vehicle as available for service. Two options are possible here: • *minBatteryPerc = 0* => The simulator will calculate the minimum battery percentage necessary to make the longest possible trip in the service region and will consider this value as the default (See Section 4.2). • *minBatteryPerc > 0* => The simulator will take this user-defined value.	0
IniDistributionKnown	Selects the initial vehicle distribution input. • *IniDistributionKnown = 0* => Initial distribution is an optimal estimation (see __). • *IniDistributionKnown = 1* => Initial distribution is given.	0
If UsersKnown = 1		
IniDistributionFile	Path to the Excel file with the initial distribution given. User must aware that the IDs are the right ones, and that the sum of all vehicles is equal to the total fleet.	-

2.1.6. User parameters

These parameters define users' characteristics. These may result from the users' behavioral characteristics, from the city's private car performance, or from the car-sharing own system regulations. All inputs in this group are mandatory.

Table 6. User behavior inputs.

Parameter	Description	Default Value
Wmax (a)	Maximum access distance that users are willing to walk to access one vehicle (at origin or destination of the trip). Measured in meters.	400 meters
WalkSpeed (v_w)	Average walking speed in km/h. (A default value of 3km/h is suitable in most cases).	3 km/h
CarSpeed (v)	Average vehicle speed while traveling within the city [km/h]	15.3 km/h (car)
avgParkTime (τ_p)	Average time needed to park in the city [min]	6 minutes (car)
Trips_SB2FF	Types of trips allowed in the vehicle-sharing system: • *Trips_SB2FF = 0* => SB to FF trips are NOT allowed. • *Trips_SB2FF = 1* => SB to FF trips ARE allowed.	1
percParking (SB_PRK)	Fraction of users preferring to park at stations instead of on-street when both options are available. This represents a city average estimation. Note that there exists the possibility of using different values of this parameter for the different zones of the service region. This zone specific values are introduced through the zonification shapefile as the property "SB_PRK" (see Section 2.3).	0.43 (car)
penSB (a_{SB})	Perceived additional access cost of parking in a station. Considered as an additional distance in meters.	100 meters (car)
fullDest	Behavior pattern when not finding available parking on destination (under only-SB service conditions): • *fullDest = 1* => User looks for a new location. • *fullDest = 2* => User waits there until a parking spot is freed.	1
costLostFF (γ_{lost_FF})	FF Lost demand penalty. Perceived cost of not finding an available vehicle on-street. In €.	2.5 €/trip

costLostSB (γ_{lost_SB})	SB Lost demand penalty. Perceived cost of not finding an available vehicle in stations. In €.	7.93 €/trip

2.1.7. Repositioning parameters

The parameters in this group define the repositioning operations. All inputs are mandatory in this group, although they will not have any impact in case of selecting the "no repositioning" option. Any default value will suffice in this case.

Table 7. Repositioning inputs.

Parameter	Description	Default Value
repoMethod	Repositioning algorithm considered to assign tasks. In the case of car-sharing: • _repoMethod = 0_ => No repositioning considered. • _repoMethod = 1_ => Dynamic pairwise assignment repositioning For the case of bike-sharing: • _repoMethod = 0_ => No repositioning considered. • _repoMethod = 1_ => Dynamic pairwise assignment repositioning • _repoMethod = 2_ => Preemptive route optimization repositioning • _repoMethod = 3_ => Mixed routing optimization and dynamic pairwise modification.	1
repoTeams (k)	Number of repositioning teams working simultaneously.	13 teams
repoCapacity	Maximum number of vehicles carried per team. (In car-sharing must be equal to 1)	16 vehicles/team
repoSpeed (v_k)	Average speed of repositioning teams when riding a scooter, in km/h.	8.8 km/h
taskDuration (δ)	Average duration of the fixed tasks for every repositioning operation, in minutes. This considers all the operations to be done by the repo team when he is not moving (i.e. at the origin or destination of the repositioning movement).	6 minutes
costRepo (C_t)	Labor and equipment cost of repositioning employees prorated per hour.	21.54 €/h

2.2. The stations' list input file ("stations.xlsx")

The file "stations.xlsx" is a MS Excel spreadsheet including all the information about user defined stations. **Even if the user does not want to introduce any station, the file must exist**. The name and path of this file can be defined by the user, but the file must follow the default template and its path must be correctly indicated in the input file (see the previous section).

Figure 2. Example of station' list input file.

In the stations' list input file, each row defines a different station. In turn, each column defines a property or parameter associated to the stations. The different properties of stations are:

Table 8. Station' properties in "stations.xlsx" input file.

Column	Property	Description
Column A	*ID*	Station identifier. It can be any integer number up to 10000. Values from 10001 onwards are reserved for stations created by the simulator.
Column B	*Name*	Name of the station. It is optional. Stations created by the simulator have no name.
Column C	*X*	X coordinate in UTM of the station's location.
Column D	*Y*	Y coordinate in UTM of the station's location.
Column E	*Z*	Z coordinate in UTM of the station's location.
Column F	*Cap*	Number of parking slots of the station. Maximum number of car-sharing vehicles that simultaneously can park at the station.
Column G	*e-chargers*	Number of charging points. Maximum number of e-cars that can simultaneously recharge at the parking station. **The number of e-chargers must be always less or equal than the capacity of the station**. Otherwise, additional e-chargers will not be used.
Column H	*Info 1*	Free information column. Optional. The simulator does not read this information. So, it can be used to include any additional information of the station. In the template, the address of the station is included here.

Column I	*Info 2*	Free information column. Optional. The simulator does not read this information. So, it can be used to include any additional information of the station. In the template, the operator is included here.
Column J	*Info 3*	Free information column. Optional. The simulator does not read this information. So, it can be used to include any additional information of the station. In the template, the priority for implementation is included here.

Two important things must be considered when introducing the stations' input. First, the number of stations created in the simulator is determined by the "TotalStat" parameter. This is independent of how many stations there are in the "stations.xlsx" input file. "TotalStat" can be larger or smaller than the number of stations in the list, but **the simulator will always start generating stations from the first row to the last of the list**. So, **stations in "stations.xlsx" input file must be introduced with its priority order and without leaving blank rows in the middle**. The stations' generation process will stop when "TotalStat" number of stations is reached. If the number of stations in the "stations.xlsx" input file is not enough to reach "TotalStat", then the simulator will generate additional stations at optimal locations (see Section 4.4 for further details).

And second, **stations outside of the service area, will not be considered**. If this happens, the simulator will return a warning, will ignore the station, and will continue to check the next one in the list.

Finally, with respect to electrical vehicle recharging, note that all battery charging points at stations are equal. However, two possible configurations can be implemented in the simulator: i) distributed recharging at stations with domestic installations, and ii) Efficient recharging at a hub.

i. *Distributed recharging* is the default option. In this configuration, electrical vehicles are recharged at stations with the available number of chargers. No additional considerations are required.

ii. *Hub recharging* implies that electrical vehicles are moved to global recharging locations with efficient infrastructure by company employees (i.e. repositioning teams). After battery charging is complete, repositioning teams will return the vehicles to the streets or to the stations. Charging hubs must be included in the simulator as special stations with the following criteria:

   o Charging hubs must be included in the first rows of the "stations.xlsx" file, including the same parameters as a normal station.

   o The number of stations "TotalStat" in the "inputs.xlsx" file must consider the number of charging hubs in addition to the number of normal stations.

   o The default value of charging points on normal stations must be changed to 0.

   o Recharging time must be modified to account for the more efficient battery charging infrastructure at the hub.

2.3. The service area and zonification shapefile (.shp)

Users are allowed to introduce in the simulator their own zonification of the service region through a shapefile. Shapefiles are common outputs from GIS softwares. The name and the path of this shapefile must be introduced in the general inputs file "inputs.xlsx".

The service area and zonification input shapefile must have the extension ".shp", and follow the next requirements and recommendations:

i.   The zonification must cover the whole service area. Overlappings and gaps should be avoided as they could imply unexpected results.

ii. Zones can have any size and shape. However, it is advised to avoid narrow zones, and use zones whose approximate area is Wmax^2 (where "Wmax" is the user-defined maximum access distance to a vehicle).

iii. Zones can have any ID or numeration.

iv. The service area and zonification shapefile must be saved in UTM coordinates.

v.   If different on-street parking preferences (i.e. "percParking" parameter) are considered for the different zones, the zones must include a parameter in the shapefile called "SB_PRK", which depicts the percentage of users that would prefer to park in a station instead of on-street when both options are available. The value of this parameter at every zone must be between 0 and 1. Recall that the average of all "SB_PRK" must be included in the "input.xslx" file as the parameter "percParking".

Creating, editing, and mapping shapefiles is not a feature of the simulator. If this option is to be used, users should consider that they will need another GIS support software for the shapefile edition. Most of the GIS-based software can fulfill that purpose.

2.4. The service area perimeter file (.xslx)

If the zonification is not available as a shapefile, the simulator can create a custom zonification. In this case, the definition of the perimeter of the service area is the only required input.

This perimeter must be introduced through an MS Excel spreadsheet file with the same structure as in Figure 3. The first row is the header, and should not be edited. The next rows are the vertices defining a closed polygonal line which defines the perimeter of the service area. The different columns define each vertex ID (from 1 to N), and their UTM coordinates (X, Y, Z). Note that the elevation coordinate (i.e. Z) is not necessary.

	A	B	C	D
1	vertex ID	X	Y	Z
2	1	430923.570	4580123.745	0.000
3	2	429371.577	4580192.722	0.000
4	3	428267.938	4579732.873	0.000
5	4	425240.117	4582136.305	0.000

Figure 3. Example of the service area perimeter input file.

The name and the path to the service area perimeter file must be introduced in the general inputs file "inputs.xlsx". It is recommended to save this file inside a subfolder of the "Data" folder, together with the main input files.

If this option is used to define the service area, meaning that the zonification is created within the simulator, the program will return a shapefile with the created service area and zonification. This could be exported to any GIS-based software. The name and path for this file is given in the general inputs file "inputs.xlsx". For more information about this custom zonification process, see Section 4.1.2.

2.5.   The O/D demand matrix input (.csv)

If the zonification input was from a shapefile (i.e. with some pre-existent zones), the simulator allows introducing the demand O/D matrices according to this existing zonification. The demand O/D matrices represent the total number of **potential trips** generated during a simulation cycle (e.g. one day). Note that depending on the car-sharing system design (e.g. vehicle availability) it is possible that not all of these trips will be served, giving rise to some lost demand.

Different O/D demand matrices can be introduced for different periods of the simulation. This defines a set of matrices. The duration for which each matrix applies, which must be the same for all matrices in the set, is defined by the user as the input "TimeReDemand" in the general input file "inputs.xslx". It is strongly recommended that the parameter defining the simulation cycle time (i.e. the "TotalTime" parameter in the "inputs.xslx" file) is set as it follows:

$$"TotalTime" = \#of\ O/D\ matrices \cdot "TimeReDemand"$$

This implies consistency between the period for which the demand for the car-sharing system is defined and the simulation cycle time. Introducing a different value for "TotalTime" may not necessarily return an error, but it could bring unexpected results. In general, if a lower value is introduced for "TotalTime", only the first demand matrices of the set will be used. In contrast, if a higher value is introduced, the additional simulation time will consider a zero demand. For more information about the simulation cycle time, refer to Section 3.2.

Finally, in order to introduce a set of O/D demand matrices in the simulator, the following conditions must apply. An example is provided in Figure 4.

   i.   Demand matrices must be in .csv format.

   ii.   All files in the same demand set must be stored in the same folder.

   iii.   All files in the same demand set must have the same name, ended with an underscore (_) and a number suffix from 0 to N, where N is the number of matrices in the set. For instance, a valid name for the first matrix in the set could be "nameoftheset_0.csv".

   iv.   The path to the folder and the name of the matrices (e.g. "Data/folder/nameoftheset") must be introduced in the field "Odmat_prefix" in the general inputs file "inputs.xslx".

   v.   Matrices must be structured in three comma separated values: origin zone, destination zone, and total number of trips between both zones during the period.

   vi.   The numeration or ID of the zones must be the same as they appear in the shapefile previously introduced (see Section 2.3)

   vii.   The first row of the .csv file is always reserved for titles, and should not be edited.

Figure 4. Example of OD matrix csv input file.

2.6. The initial distribution of vehicles input (.xlsx)

If considered, user can introduce an initial distribution of vehicles in .xlsx format. The file must contain two columns with a name in the first row. This option is implemented only for the stations and not for the free-floating zones at version 1.8.0 of the Vsharing Simulator.

- ID → The first column must contain the ID of the station.
- Initial vehicles → The second column must contain the number of vehicles at that station at the beginning of the simulation.

Figure 5. Example of initial distribution of vehicles input file.

User must check that introduced IDs are correct and corresponds to an existing station inside of the service area. In case that this file is used to set the initial distribution of vehicles, the total number of SB vehicles input at "inputs.xlsx" no longer applies. Instead, the total number of vehicles at stations will be the sum of all initial vehicles in existing stations.

Consider also that this initial state is applied at the beginning of the warm-up period.

3. Simulation model

After the inputs to the simulator have been described in the previous section, here we describe the simulation model. This includes the treatment of space and time in the model and the definition and behavioral rules of all the simulation agents.

3.1. Spatial modeling

The car-sharing simulator considers continuous space inside the service region. All the agents and elements of the simulation are considered to be located at single points, defined by their position in X, Y coordinates.

The coordinates of all locations, including the inputs, must be in UTM system. Any UTM zone can be used, but the user must be aware of using the same zone for all inputs.

Distances in the model are calculated according to the L1 'Manhattan' metric:

$$dist(A, B) = abs|x_B - x_A| + abs|y_B - y_A|$$

3.2. Time modeling and simulation steps

Time in the simulation time is defined by:

     i.   Time step
    ii.   The demand update period
   iii.   Demand cycle

The time step defines the smallest unit of time in the simulator. Agents' status is updated every time step. The simulator considers a time step of 1 minute and this cannot be changed by the user. So, all simulation operations happen in a per minute basis (e.g. generating new users, moving active users, moving repositioning teams, updating the battery levels of e-cars, etc.) until the simulation ends.

The demand updated period deals with how often the demand rate changes. Demand in the simulator is defined by a set of demand matrices. This set of matrices are either inputs defined by the user, or estimated from aggregated values by the simulator. In any case, the demand update period defines the duration for which each O/D matrix holds. The demand update period is **defined by the user in the general inputs file (in minutes)** (i.e. the demand update period is defined by the "TimeReDemand" parameter in the "inputs.xslx" file). This value must hold for all matrices in the set. For example, if the demand update period is defined as 60 minutes, it means that all matrices are hourly O/D demand matrices.

Finally, the demand cycle defines the total duration of the whole set of O/D demand matrices, and corresponds to the total duration of the simulation. **The demand cycle is introduced by the user in the general inputs file (in minutes)** (i.e. the demand cycle is defined as the "TotalTime" parameter in the "inputs.xslx" file) **and its value should be equal to the demand update period multiplied by the number of introduced matrices**. For example, if the demand input is composed by a set of 24 hourly matrices, the demand cycle should be 24*60=1440 minutes.

$$Demand\ cycle = \#of\ O/D\ matrices \cdot Demand\ update\ period$$

Setting correctly the demand cycle time is fundamental when demand matrices are introduced by the user, because this defines how many matrix files the simulator will read. If the demand

cycle is too short, only the first matrices will be read. Otherwise, if the demand cycle is too long, the missing O/D matrices will be considered as zero.

3.2.1. Warm-up period and simulation time.

A complete simulation consists in several demand cycles, divided between the warm-up period and the simulation time:

i. *Warm-up period*: the warm-up period is a pre-defined time duration while the simulator runs with the only purpose of reaching a random, realistic, and stationary status. This allows washing out the effects of a deterministic (probably optimized) initial configuration. Results are not recorded during this warm-up period. **The duration of the warm-up period is defined by the user in the general inputs file. This represents the integer number of complete cycles considered for the warm-up**. At the end of each cycle only the current state of active objects is maintained. This includes:

- *Stations:* Current list of cars at each station and using recharging points.

- *Cars:* Current position, battery level, and status (i.e. idle, reserved, in trip, discharged).

- *Users:* Only for active users. All properties are kept (origin, destination, reserved car, current position).

- *Repositioning teams:* Current position, status (i.e. idle, moving on scooter to a car, returning the car to a new position), and unfinished tasks. Finished tasks are deleted.

ii. *Simulation time*: The simulation time starts just after the warm-up period, and it defines the period from which all simulation results are stored. It is always composed of one demand cycle and this cannot be modified by the user.

The typical setup would consider a demand cycle of one day, with a demand update period of one hour (i.e. requiring 24 O/D demand matrices). The warm-up period could be defined as 2 or 3 demand cycles (i.e. days).

In spite of this, other simulation scenarios are possible by changing the time and demand inputs. For instance, it is possible to obtain results for more than one day of simulation, or to consider that part of the day the system is closed.

a) *Obtaining results for more than one day.*

In this case, the user must redefine the demand cycle and replicate the O/D demand matrix. This can be done by implementing the following modifications to the inputs:

i. Set the "TotalTime" input in the file "inputs.xlsx" as the new demand cycle which should be set equal to the desired number of days of the simulation.

ii. The set of O/D demand matrices must cover the whole new demand cycle. If demand for each day is the same, just replicate the set of O/D matrices (.csv files). Paste them in the same folder as the original set, and rename them with the same name and consecutive numeration

For example, in case of having a set of 24 hourly matrices (numbered from 0 to 23), to obtain results for two days of simulation it would be necessary to create a copy of the .csv files and include them in the same folder with a numeration from 24 to 47. Then, the "TotalTime" input will be changed to 2880 minutes (48 hours).

Note that increasing the demand cycle also affects the warm-up period, because warm-up is defined as a number of cycles. So, it could be interesting to reduce the number of warm-up cycles, because now, each one represents more than one day.

b)   Considering closing times.

Some systems might be inaccessible during some hours every day (e.g. night hours). During this period, the system is closed to users' demand, but still some system maintenance operations are performed (i.e. battery recharging and repositioning operations).

The simulator can include this closing times by using void O/D demand matrices in the set. For example, in case of a closing time from 23:00 to 05:00, a valid set of 24 hourly matrices will include void matrices numbered from 0 to 4, non-zero demand matrices numbered from 5 to 22, and a last zeros matrix numbered as 23.

3.3.   Agents in the simulation

This car-sharing simulator is based on an agent modelling approach and object-oriented programming. So, each element in the simulation is defined as an object, and performs actions individually.

In this subsection, each type of agent is briefly described with its properties.

3.3.1. Stations

Stations are passive objects and do not perform actions. However, their properties are constantly being queried by other agents to perform their own actions. For example, users check the stations status for searching an available car in the car list, or repositioning teams check the charger list to allow recharging electric cars.

Table 9. Properties of the object "Station".

Property	Description
ID	Station identifier
Name	Name of the station (Optional)
X	X coordinate in UTM
Y	Y coordinate in UTM
Z	Z coordinate in UTM
zoneID	Identifier of the zone where the station belongs
capacity	Maximum number of parking slots
numChargers	Number of electric chargers in the station

nearestCharger	ID of the closest station with chargers
numCars	Current number of ICE (internal combustion engine) cars
numEcars	Current number of electric cars
accRequests	Accumulated number of car requests over time
accReturns	Accumulated number of car returns over time
optCars	Array of optimum number of cars in the station over time
listCars	List of car IDs currently in the station
listCharging	List of electric car IDs currently connected to battery chargers
vlistCars	Array of car lists over time in the station
vlistCharging	Array of car charging list over time in the station

3.3.2. Free floating zones

Free floating zones are equivalent to virtual stations. Therefore, most of their properties are common with stations.

Table 10. Properties of the object "Free-floating zone".

Property	Description
ID	Free floating zone identifier
X	X coordinate in UTM
Y	Y coordinate in UTM
Z	Z coordinate in UTM
zoneArea	Area of the zone in km^2
capacity	Maximum number of car-sharing vehicles that should be in the zone (default value = 9999)
numChargers	Number of electric chargers in the zone (default value = 0)
nearestCharger	ID of the closest station with chargers
numStations	Number of stations inside the zone
listStations	List of station IDs inside the zone
numStations_neig	Number of stations inside neighboring zones
listStations_neig	List of station IDs inside neighbor zones
accRequests	Accumulated number of car requests over time

accReturns	Accumulated number of car returns over time
optCars	Array of optimum number of cars in the zone over time
listCars	List of car IDs currently in the zone
vlistCars	Array of car lists over time in the zone

3.3.3. Vehicles

Vehicles are still passive agents, like stations and free-floating zones. Cars do not perform any actions by themselves, except reducing and increasing their battery levels if they are in use or in a charging list.

Table 11. Properties of the object "Vehicle".

Property	Description
ID	Car identifier
X	Array of X coordinates in UTM over time
Y	Array of Y coordinates in UTM over time
Z	Array of Z coordinates in UTM over time
status	Array of car status over time (0: idle, 1: reserved, 2: on-trip, 3: repositioning, 4: not enough battery)
isElectric	Determines if the car is electric or not
batteryLevel	Array of battery levels (in percentage) over time

3.3.4. Users

Users are active agents. They are created at every time step according to the O/D demand matrices. On creation, each user tries to reserve a car for his trip. If this is not possible, the user dies and stored separately as demand lost.

For more details about the user creation and movement process, read Sections 5.1and 5.2.

Table 12. Properties of the object "User".

Property	Description
X	Current X coordinate in UTM
Y	Current Y coordinate in UTM
Z	Current Z coordinate in UTM
XO	Origin of the trip X coordinate in UTM
YO	Origin of the trip Y coordinate in UTM

ZO	Origin of the trip Z coordinate in UTM
ZoneO	ID of the origin zone
CarZoneO	ID of the zone where the it is located the reserved car. Note that this may be different from the origin zone
XD	Destination of the trip X coordinate in UTM
YD	Destination of the trip Y coordinate in UTM
ZD	Destination of the trip Z coordinate in UTM
ZoneD	ID of the destination zone
CarZoneD	ID of the zone where the car will be returned. Note that this may be different from the destination zone
CarID	ID of the reserved car
StatO	ID of the station where the reserved car is located. If the reserved car is on the street, then StatO = 0, and the previous CarZoneO is used instead.
StatD	ID of the station where the car will be returned. If the car car is to be returned on the street, then StatD = 0, and CarZoneD is used instead.
Xpark	X coordinate in UTM of the parking location.
Ypark	X coordinate in UTM of the parking location.
tCreation	Time step when the user was created
tO2Car	Time step when the user reaches the reserved car at the origin of the trip.
tTrip	Time step when the trip with the car is finished and the car has been parked and returned.
tCar2D	Time step when the user arrives to the final destination point.

3.3.5. Repositioning tasks

Tasks are auxiliary objects in the repositioning process. They represent each one of the possible repositioning movements (i.e. artificially take/return one vehicle from a station or zone) in the system that can be assigned to an idle repositioning vehicle. Tasks are generated when the repositioning assignment process begins. If one task is assigned to a repositioning team, its relevant properties are copied into the team properties. All tasks are deleted right after the completion of the assignment process.

For more details about the task assignment process, read Section 5.4.

Table 13. Properties of the object "Task".

Property	Description
ID	Task identifier
X	X coordinate of the task in UTM (i.e. location where to take/return the vehicle)
Y	Y coordinate of the task in UTM (i.e. location where to take/return the vehicle)
Z	Z coordinate of the task in UTM (i.e. location where to take/return the vehicle)
taskStat	Shows if the task takes place in a station or in a free-floating zone
taskType	Shows reason of repositioning (1: recharging, 2: relocation)
taskStatID	Shows the ID of the station/zone where the task takes place
taskMovements	Number of vehicles to move. In car-sharing systems "taskMovements" is always equal to +1 (return one vehicle) or -1 (take one vehicle)
taskUtility	Utility value of completing the task in €

3.3.6. Repositioning teams

Repositioning teams are active agents. They are created at the beginning of the simulation and they perform repositioning tasks. They store all the assigned tasks for each demand cycle (e.g. 24 hours).

For more details about their movement and task assignment process, read Sections 5.4 and 5.5.

Table 14. Properties of the object "Repositioning team".

Property	Description
ID	Repositioning team identifier
X	Array of X coordinates in UTM over time
Y	Array of Y coordinates in UTM over time
Z	Array of Z coordinates in UTM over time
status	Array of repositioning time status over time (0: idle, 1: moving on scooter to a car, 2: returning the car to a new position)
vehicles	Array of the vehicles' ID repositioned over time
capacity	Maximum number of vehicles that the team can carry simultaneously. (Default value in carsharing systems is 1)

taskStat	Array of taskStat. Shows if the task took place at a station or at a free-floating zone
taskList	Array of taskList. Shows the ID of the station/zone where the task took place
taskType	Array of taskType. Shows reason of repositioning (1: recharging, 2: relocation)
taskMovements	Array of vehicles to move on each task. In car-sharing systems "taskMovements" is always equal to +1 (return one vehicle) or -1 (take one vehicle)
taskTime	Array of the ending times of the performed tasks
taskCurrent	Index of current task in the previous arrays
carID	ID of currently car being repositioned

3.3.7. City

City is an auxiliary object used as a container of all elements and objects of the simulation. For each simulation, a single City object is created as a Matlab file (.mat), which later is used to generate the selected outputs.

City objects contain the following properties:

Table 15. Properties of the object "City".

Property	Description
servArea	Contains the zonification of the service area as a struct array of matlab polygons. Each polygon represents a single free-floating zone.
zoneNum	Array containing the external ID of the service area zonification. It is used for a quicker search.
OD	Contains the OD matrices input. It is structured as a struct array of sparse matrices.
output	Postprocessing object. Only for calculation references.
numStations	Total number of stations in the system
vStations	Array of all station objects in the system (see Section 3.3.1)
numFreeFloatZones	Total number of free-floating zones in the system
vFreeFloatZones	Array of all free-floating zone objects in the system (see Section 3.3.2)
numCars	Total number of cars in the system
vCars	Array of all car objects in the system (see Section **Error! Reference source not found.**)

minBatteryLevel	Minimum battery level percentage for cars. Any car with a battery level below this threshold will be assigned status "discharged" and unavailable for user reservations. (See Section 4.2 for the estimation process of the minimum battery level).
numRepoTeams	Total number of repositioning teams in the system
vRepoTeams	Array of all repositioning teams objects in the system (see Section 3.3.6)
numUsers	Total number of active users in the system
vUsers	Array of all active users objects in the system (see Section 3.3.4)
numFinishedUsers	Total number of users who have finished their trips in the system during the simulation
vFinishedUsers	Array of all users who have finished their trips in the system during the simulation
notServicedUsers	Array of all not serviced users in the system during the simulation

4. Simulation setup modules

This section describes all the modules that perform operations just once and before the simulation time starts. Most of those modules create the fixed elements of the car-sharing system.

4.1. Creation of the free-floating zonification

This module divides the service region into several FF zones or virtual stations. These elements are not real infrastructure, but are used as an auxiliary element to control the distribution of cars on streets.

There are two possibilities of creation: i) import FF zones from a shapefile, and ii) create the FF zones within the simulator.

4.1.1. Import FF zones from a known zonification.

To use this option, a zonification shapefile (.shp) must be introduced as an input. The simulator will read the shapefile and will create the set of FF zones.

The simulator accepts zones of any shape and size. However, it is advisable to avoid narrow zones, while constructing them as uniform as possible and with an average area of $Wmax^2$ (where "Wmax" is the user-defined maximum access distance to a vehicle).

The easiness or difficulty of finding a parking spot on the street can be introduced as a different value for each zone. In this case, the zonification shapefile must include a property called "SB_PRK", which depicts the fraction of demand preferring to park on stations when both options (i.e. station and on-street parking) are available. If this property is not included, the simulator will take the default "percParking" parameter value for all zones.

See Section 2.3. for more information about the shapefile input.

4.1.2. FF zones from a square grid inside a given perimeter.

This option requires only the input of the perimeter of the service region. This is achieved through an excel spreadsheet, which includes the coordinates of the vertices of the closed polygonal line defining the perimeter (See Section 2.4 for more information about the perimeter input).

The simulator reads the perimeter and creates a uniform square grid, with an element area equal to Wmax^2. All the square elements of the grid, which will be defined as the FF zones, will have all four vertices inside the perimeter. Note that, for this reason, the final service region might be slightly smaller than the area defined by the perimeter input.

The percentage of SB parking demand in case of both parking options available (i.e. on-street or at station) will be the same for all zones and equal to the input "percParking" in the file "inputs.xslx" (see Section 2.1.6).

The resulting zonification will be returned as an output of the simulator with .shp format (i.e. a shapefile). This allows users to make modifications on it for further experiments using an external GIS-based software.
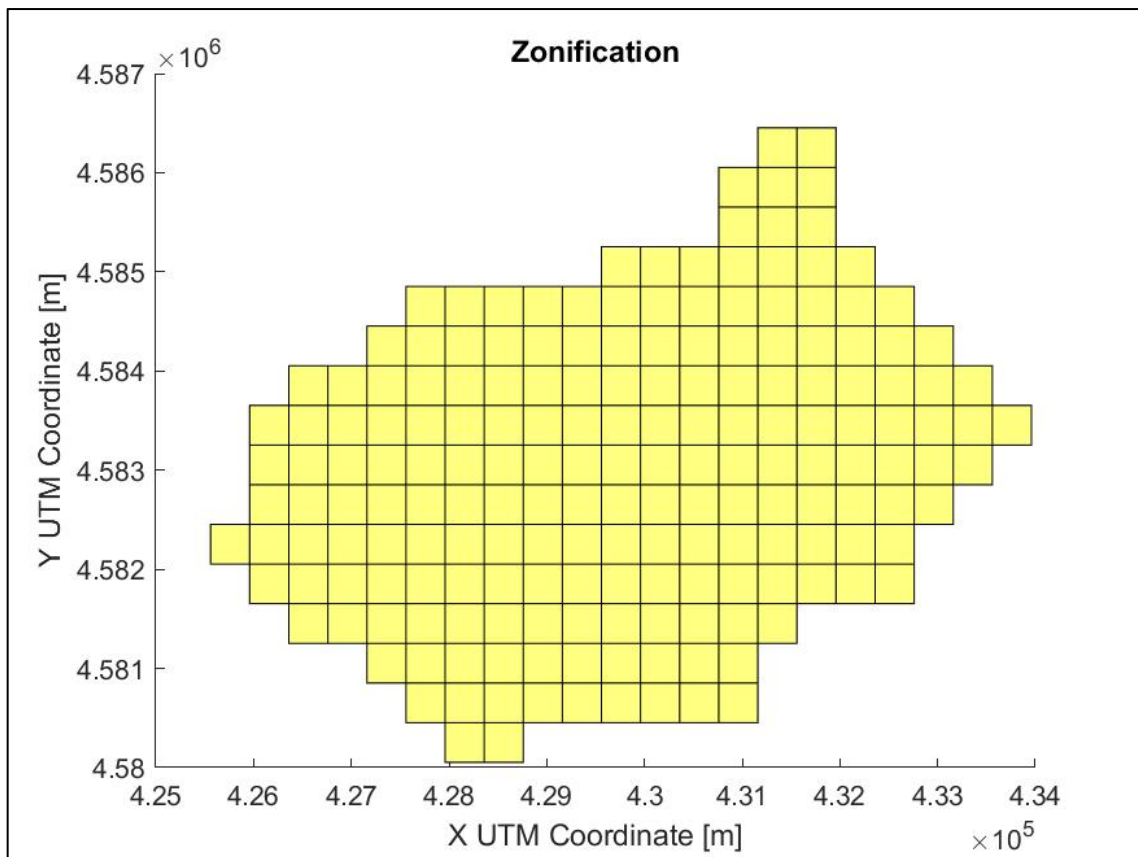

Figure 6. *Example of a square grid zonification created by the simulator.*

4.2. Estimation of the minimum battery level of electrical vehicles

The minimum battery level is a parameter that determines if an electrical car can be used by users. If a car is below this battery level, it will be considered as unavailable to users. Cars below

that level have a "discharged" status and are assigned the highest priority for being repositioned and recharged.

This minimum battery level is a user-defined parameter that can be changed in field "minBatteryPerc" of the "inputs.xlsx" file (see Section 2.1.5). The input must be a percentage, with values between 0 and 100. In spite of this, the simulator allows two possible options for setting this value:

i. *minBatteryPerc = 0* => If the user introduces a zero, the simulator will estimate the minimum battery level according to the following equation. The resulting value will be shown in the input recap file after the simulation.

$$minimum\ battery\ level\ \% = 100 \cdot 3.2 \cdot \frac{\sqrt{Total\ service\ area}}{Battery\ autonomy\ in\ km}$$

The previous equation determines the minimum battery level required to perform a trip crossing the whole service area. A safety factor of 3.2 is introduced to account for trip circuity and some additional battery reserve.

ii. *minBatteryPerc > 0* => If the user introduces a percentage different than zero, the simulator will take this value as the minimum battery level. (Be careful: this case also applies if setting a value very close to zero, e.g. 0.001.)

4.3. Creation of O/D demand matrices

There are three elements that define the time evolution in the simulator:

- *Time step*: The simulator considers a **time step of 1 minute and this cannot be changed by the user**.

- *Demand cycle*: Defines the total simulation period. By default, it is considered 24 hours (1440 minutes), but it can be changed by the user.

- *Demand update period*: It defines the duration of each demand matrix in the set. It must be the same for all demand matrices, and for being consistent, it must be equal to the demand cycle divided by the number of O/D matrices in the set.

The module of the simulation described in this section, creates the O/D demand matrices for a demand cycle.

The simulator offers two possibilities: i) import the OD demand matrices from .csv files, and ii) estimate the OD demand matrices from the aggregated demand parameters. Both options are explained in the following subsections.

4.3.1. Import O/D demand matrices as .csv files

This option requires of a sparse O/D matrix input in .csv format. The structure of each file must consist into three comma separated values: origin zone, destination zone, total number of trips during the demand update period. The name of the file must end with an underscore (_) and a numeration from 0 to N for each matrix in the set. Read Section 2.5 for more information about the demand matrix inputs.

If using this option, the demand update period must be equal for all matrices and must be explicitly introduced in the input file in minutes (i.e. in the "TimeReDemand" parameter included in the "inputs.xlsx", see Section 2.1.3).

This option is only available if a the zonification has been previously imported through a shapefile. The zone IDs of the sparse .csv files must reference the same ID on the zonification shapefile.

4.3.2. Estimate O/D demand matrices from aggregated demand values

This option estimates O/D demand matrices when only the demand aggregated parameters are available. These parameters include the total number of potential trips , N, during the total time of the simulation (i.e. the demand cycle, T), and the spatial imbalance and temporal evolution factors.

Temporal evolution is introduced through an array of temporal weights. The number of elements in the array defines the number of periods in which the demand cycle is divided. Each element is the relative weight of that period with respect the others in the whole cycle. For example, an array [1;2] means that the demand cycle is divided in two periods, and in the second period the demand is the double than that of the first one.

Spatial imbalance is introduced by the aggregated imbalance parameters and a shape pattern.

The aggregated imbalance parameters are three: i) The fraction of the service area in which there are more car requests than returns, π_q (i.e. generating areas); ii) the fraction of the service area in which there are more car returns than requests, π_t (i.e. attracting areas). Note that the sum of the two previous fractions must be less than or equal to 1); and iii) the average fraction of imbalanced trips, Φ.

Φ is the average imbalance considering both, generating and attracting zones, and can be expressed as;

$$\Phi = \Phi_t \pi_t + |\Phi_q| \pi_q$$

Where Φ_t and Φ_q are the average imbalance fractions over attracting and generating zones respectively. Note the absolute value operator applied to Φ_q, as Φ_q is defined negative. This is because the fraction of unbalanced trips is defined as returns minus requests, while in generating zones requests are predominant.

Note that because of the conservation of vehicles in the service region we have:

$$\Phi_t \pi_t = |\Phi_q| \pi_q$$

Meaning that vehicles in excess in one region correspond to a deficit of vehicles in another.

Then, considering the two previous equations we have:

$$\Phi = 2\Phi_t \pi_t = 2|\Phi_q| \pi_q$$

And finally, Φ_t and Φ_q can be obtained from the inputs Φ, π_t, and π_q as:

$$\Phi_t = \frac{\Phi}{2\pi_t}$$

$$\Phi_q = -\frac{\Phi}{2\pi_q}$$

In turn, the shape pattern determines how the imbalance is spread over the service region. Two shape patterns are considered in this module: i) flat pattern, and ii) radial pattern.

The flat pattern implies that the imbalance has the same dominant direction over the whole service area. Also, it is assumed that the dominant direction is perpendicular to an imbalance axis, defined as a straight line in the service region where the imbalance is null. If this pattern is selected, the dominant direction must be an input, introduced as the positive angle (φ) created with the north direction and the imbalance direction.

In contrast, the radial imbalance implies that imbalance, either an excess of returns or requests, accumulate with a focus. It is assumed that the imbalance is reduced gradually as one moves away from the focus. So, excess or requests around the focus area gets compensated by an excess of returns in the perimeter (or viceversa). If this option is selected, the coordinates of the focus must be given as an input (X_C, Y_C). For this radial pattern, the parameter, φ is set to 1 if the focus is a return focus ($\varphi = 1$) or set to -1 if it is a request focus ($\varphi = -1$).
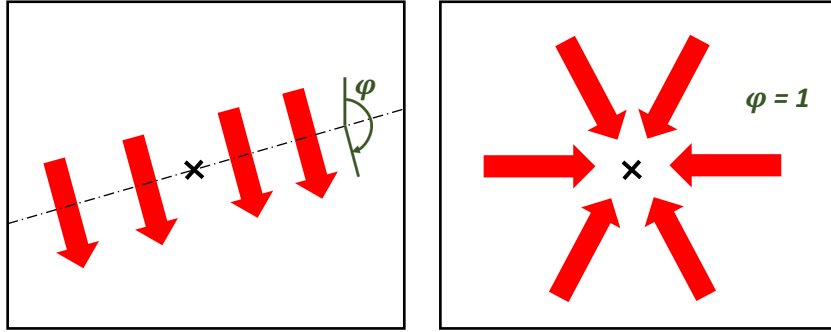

Figure 7. Example of flat (left) and radial (right, with $\varphi = 1$) imbalance patterns.

The simulator computes the distance from the centroid of each free-floating zone i, with coordinates X_i, Y_i, to the axis/focus, depending on the pattern selected, according to the following equation:

$$d(i,C) = \begin{cases} -\cos(\varphi) \cdot (X_i - X_C) - \text{sen}(\varphi) \cdot (Y_i - Y_C) & \text{if pattern is 'flat'} \\ \varphi \cdot \sqrt{(X_i - X_C)^2 + (Y_i - Y_C)^2} & \text{if pattern is 'radial'} \end{cases}$$

Note that in the flat pattern only the dominant imbalance direction is given, and X_C, Y_C are taken as the coordinates of the centroid of the whole service area (i.e. assuming that the axis goes through the centroid). However, the centroid of the service area acts only as an auxiliary point, since the fractions of π_t, π_q are known and define the particular location of the axis, which may not cross through the centroid. This means that $d(i,C)$ in this case needs to be corrected later on in the process.

Zones are ordered according to their values of $dist(i,C)$, and then classified as generating or attracting zones. This classification is done according to π_t, π_q. Assuming N to be the total number of zones, and π_q the fraction of service area with more requests than returns, then $N\pi_q$ will be the number of trip generating zones. These will be assigned to the $N\pi_q$ zones with the highest $d(i,C)$. Alternatively, the $N\pi_t$ zones with lowest distance will be classified as trip

attracting zones. If there is any zone left, it will be considered as equilibrated (i.e. balanced, with the same number of car requests and returns).

After this step distances are corrected. In order to do so, the average distance between the equilibrated zones, the limit subzone belonging to π_q (lowest distance of the set π_q), and the limit subzone belonging to π_t (highest distance of the set π_t) is computed. The resulting value is subtracted from all distances. That way, it is ensured that all distances in π_q will be positive, and all distances in π_t will be negative.

The total number of requests, $n_q(i,t)$, and returns, $n_t(i,t)$, for zone i and demand period t, is calculated from the total number of trips during period t, $n_{(t)}$, as follows:

$$n_{(t)} = N \cdot \frac{W_{(t)}}{\sum_t W_{(t)}}$$

Where here, N is the potential number of trips in the demand cycle, and $W_{(t)}$ represents the weight of the demand period t with respect to the other periods of the demand cycle.

If zone i belongs to π_t (i.e. attracting zone), then:

$$n_q(i,t) = n_{(t)} \cdot \left(\frac{r_{(i)}}{\sum_i r_{(i)}} \right) \cdot \left(1 - \frac{\Phi_t}{2} \cdot \frac{d(i,C) \cdot N\pi_t}{\sum_{i \in \pi_t} d(i,C)} \right)$$

$$n_t(i,t) = n_{(t)} \cdot \left(\frac{r_{(i)}}{\sum_i r_{(i)}} \right) \cdot \left(1 + \frac{\Phi_t}{2} \cdot \frac{d(i,C) \cdot N\pi_t}{\sum_{i \in \pi_t} d(i,C)} \right)$$

Where $r_{(i)}$ is the area of zone i. Note that $\sum_i r_{(i)} = R$ the area of whole the service region.

If zone i belongs to π_q (i.e. generating zone), then:

$$n_q(i,t) = n_{(t)} \cdot \left(\frac{r_{(i)}}{\sum_i r_{(i)}} \right) \cdot \left(1 + \frac{\Phi_q}{2} \cdot \frac{d(i,C) \cdot N\pi_q}{\sum_{i \in \pi_q} d(i,C)} \right)$$

$$n_t(i,t) = n_{(t)} \cdot \left(\frac{r_{(i)}}{\sum_i r_{(i)}} \right) \cdot \left(1 - \frac{\Phi_q}{2} \cdot \frac{d(i,C) \cdot N\pi_q}{\sum_{i \in \pi_q} d(i,C)} \right)$$

And if the zone is equilibrated:

$$n_q(i,t) = n_t(i,t) = n_{(t)} \cdot \left(\frac{r_{(i)}}{\sum_i r_{(i)}} \right)$$

Finally, the elements $n(i,j)$ of each cell of the matrix t are calculated as:

$$n(i,j,t) = \frac{n_q(i,t) \cdot n_t(j,t)}{n_{(t)}}$$

The resulting O/D matrices will be returned as an output of the simulator with .csv format. This allows users to make modifications and adjustments with more complex demand models using an external software.

4.4. Stations' location setup

The objective of the stations' location module is to determine the location of all car-sharing parking stations within the service region. This requires the input of the total number of stations in the system (i.e. the "TotalStat" parameter in the general inputs "inputs.xlsx" file; see Section 2.1.4), and the MS Excel file with the list of station candidate locations proposed by the user (i.e. the "stations.xlsx" file), if any. This list includes the ID of the candidate station, its location in UTM coordinates, its parking capacity, and the number of battery chargers available. A template file is provided (see Section 2.2).

With both inputs, the stations' location module of the simulator will create the desired number of stations. Assume that "TotalStat = S", so that the location of S stations needs to be determined. The module proceeds as follows:

- If there are enough stations in the candidate list, the first S stations in the list will be included in the simulator with their characteristics.

- If there are not enough stations in the candidate list, all candidate stations will be included. In addition, new stations will be generated. These new stations will have an ID from 10001 onwards, and a user-defined default capacity and number of chargers (i.e. parameters "defaultCapacity" and "defaultChargers" in the "inputs.xslx" file, see Section 2.1.4). The location of these new stations, which are not defined by the user, will depend on the following:

  o If there are free-floating zones without any station, new stations will be generated on their centroid until reaching the desired number of stations, S. The module will prioritize free-floating zones with more expected parking returns, with the objective of reducing recharging and repositioning costs.

  o Still, if there are not enough free-floating zones without stations to reach S, new stations will be generated at random locations at the zones with less station density. This aims to achieve a distribution of stations as uniform as possible, as station coverage is complete in this scenario.

The list of stations actually created within the simulator, either from the user defined candidate list or generated by the simulator, will be returned as an output in order to allow modifications for further experiments. The output file format will be an excel spreadsheet (.xlsx) with the same structure as the station candidate list input file. Note that the particular UTM coordinates of new generated stations will not necessarily correspond with feasible stations' locations.

After all stations are located within the service region, the simulator will automatically calculate all their properties, including the nearest battery charging and free-floating zones in the system.

4.5. Demand forecasting at stations and FF zones

This module of the simulator estimates the expected cumulative number of car requests and returns for each station and FF zone during the day. This prediction will be used to estimate the optimal car distribution and the required repositioning movements.

The prediction is based on the O/D demand matrices and is made in 4 steps that are repeated for all free-floating zones and stations and for each simulation time step (i.e. every minute). The 4 steps are:

1. *Calculate the expected number of returns and requests per minute in each zone.* For this prediction, it is assumed that the demand is constant over the whole demand update period, and that all the potential demand in the matrix will be served. Therefore, the expected requests per minute in zone i is the sum of the ith row of the O/D demand matrix, divided by the demand update period. Equally, the expected number of returns per minute is the sum of the ith column of the matrix, divided by the demand update period.

2. *Split the expected car requests between the FF system (i.e. reserved on street) and SB system (reserved at stations).* This split is made by considering the stations' coverage and the relative size of the SB fleet over the total. Considering the parameter "Wmax" as the maximum distance users are willing to walk to access one vehicle (see Section 2.1.6), the stations' coverage in zone i, $Coverage\_SB(i)$, is computed as:

$$Coverage\_SB(i) = \min\left\{1, \frac{s_{(i)} \cdot W_{max}^2}{r_{(i)}}\right\}$$

Where $s_{(i)}$ is the number of stations in zone i and $r_{(i)}$ is the area of zone i.

And the fraction of vehicle requests, $P_{q\_FF(i)}$ and $P_{q\_SB(i)}$ for FF and SB systems respectively, in zone i will be:

$$P_{q\_FF(i)} = 1 - Coverage\_SB(i) \cdot \frac{m_{SB}}{m_{SB} + m_{FF}}$$

$$P_{q\_SB(i)} = \frac{1 - P_{q\_FF(i)}}{s_{(i)}}$$

3. *Split the expected car returns between the FF system (i.e. reserved on street) and SB system (i.e. reserved on stations).* This split is made by considering the stations' coverage and the SB parking percentage on ideal conditions in zone i ("SB_PRK(i)"). Read Sections 2.1.6 and 2.3 for further details about this input. The fraction of demand returns FF and SB, $P_{t\_FF(i)}$ and $P_{t\_SB(i)}$, will be:

$$P_{t\_FF(i)} = 1 - SB(i) \cdot SB\_PRK(i)$$

$$P_{t\_SB(i)} = \frac{1 - P_{t\_FF(i)}}{s_{(i)}}$$

4. According to the expected number of requests and returns and the split fractions FF and SB, the cumulative vehicle forecasts can be obtained for all stations and FF zones.

This result is a property of the station/zone, which is used for estimating the optimal distribution of cars within the service region.

4.6. Initial distribution of the vehicle fleet

Vehicle creation and initial location depends on if they are FF or SB vehicles. FF vehicles will be created and located initially at their optimal location. The computation of this optimal distribution is achieved through a time-step module described in the next section (see Section 5.3). Electric and ICE cars are distributed evenly within the service region. SB vehicles could also be created and located initially at their optimal location, or can be created and located according

to an initial location given by the user. See section 2.6 for more details about the initial distribution input.

4.7. Initial distribution of repositioning teams

Repositioning teams will be created in the simulator and located initially at the centroid of the service region, with no task assigned and idle status.

5. Time-step modules

This section describes all the simulator modules that are required in order to perform operations every simulation time step (i.e. 1 minute). Some modules only will activate if necessary.

5.1. User creation and vehicle assignment

Car-sharing users are created every simulation time step, according to the demand matrix for the current period. For each O/D pair, a random number of users are created following a Poisson distribution whose average is the O/D average demand per minute (i.e. the simulation time step is of 1 minute).

For each user, the creation process is as follows:

- *Set origin and destination locations*. These are generated as random points inside the origin and destination zones respectively, following a spatial uniform distribution.

- *Check if there are stations near the final destination*. The simulator checks stations on the destination zone and its neighboring zones. Then, it is checked if the distance from any of these stations to the user final destination is less than the user-defined maximum access distance (i.e. parameter "Wmax" in the general inputs file "inputs.xlsx", see Section 2.1.6). An additional station penalty distance is applied in order to consider the additional access time experienced for entering or exiting from the parking station (i.e. parameter "penSB" in the general inputs file "inputs.xlsx", see Section 2.1.6).

- *Check if SB and FF systems are segregated*. Check if trips SB->FF are forbidden. (i.e. parameter "Trips_SB2FF" in the general inputs file "inputs.xlsx", see Section 2.1.6). If SB->FF trips are not allowed (i.e. "Trips_SB2FF = 0") and there are no stations near the final destination, then stations near the origin will not be considered, as they are not valid options for car assignment.

- *Otherwise, check if there are stations near the origin.* If SB->FF trips are allowed (i.e. "Trips_SB2FF = 1" and there is at least one station near the final destination, check all stations in the origin zone and its neighboring zones. Select the nearest station with available cars. The distance from this station to the origin of the user must fulfil the maximum access distance requirement (with the additional station penalty, "penSB") in order to be considered.

- *Find the nearest FF available car*. In order to be considered, the nearest FF available car must fulfil the maximum access distance requirement.

- *Assign a vehicle to the user.* Select the nearest between the two previous options (i.e. FF available car or SB available car) if both options are considered. For the SB option, the additional penalty distance is applied.

  - o If the FF car is selected, the car changes status to "reserved" and it is no longer available.

  - o If the SB option is selected, the user will check the car list at the station, and will select the first one which is available. If this car is an ICE car, the car is selected, the car changes status to "reserved" and it is no longer available. Alternatively, if this car is electric, the user will check the list and the available electric car with highest battery is reserved.

  - o If no car assignment is possible, the trip is cancelled. The user data is stored on a dead user array.

- *Select the parking preference.* According to the previous properties of the generated user and vehicle assignment, it is possible that only FF parking (i.e. no station near the final destination) or only SB parking (i.e. only SB=>SB trip is possible) are available. However, it is also possible that both parking options are available to the user. In this case, the parking selection is assigned randomly according to a Bernoulli trial where the probability of SB parking is set by the parameter "percParking" in the general inputs file "inputs.xlsx" (see Section 2.1.6). This parameter is substituted by the analogous "SB_PRK", in case of specific values for different destination zones.

- *Parking slot assignment.* If the parking preference is at a station, the nearest station to the final destination with available parking slots is selected. Alternatively, if the parking preference is on-street or there are no available parking spots at stations, a random location near the final destination will be selected. The time spent by the user to find this on-street final spot, and its particular location which will determine the final egress time will depend on the parking preference at destination (i.e. the general parameter "percParking" or the zone specific "SB_PRK). Note that higher preference for SB parking means that parking on-street is harder. The details of the on-street parking slot assignment are provided in a separate Section 5.1.1).

- *Compute the user's timers.* This is to compute the specific time steps when the user will reach the car at the origin of the trip, when he will reach the parking spot, and when he will reach the final destination.

  - o *Reaching the car at the origin of the trip*: To compute this timer, it is considered the access distance (i.e. the distance between the origin of the user and the location of the assigned vehicle, with the additional SB penalty, if the assigned vehicle is in a station) divided by the user walking speed (i.e. parameter "WalkSpeed" in the general inputs file "inputs.xlsx"; see Section 2.1.6).

  - o *Reaching the parking spot:* To compute this timer, it is considered the distance between the car location at the start of the trip and the parking position, divided by the car speed in the urban environment. If the parking selection is on-street, an additional parking time is included. The details of the calculation of this on-street parking time are provided in Section 5.1.1).

o *Reaching the final destination*: To compute this timer, it is considered the egress distance (i.e. the distance between the parking location and the final destination, including the SB penalty, if parking spot is in a station) divided by the user walking speed (i.e. parameter "WalkSpeed" in the general inputs file "inputs.xlsx"; see Section 2.1.6).

Finally note that, after the creation of the user and if a car was assigned, the user location coordinates become the same as the ones for the car.

5.1.1. Determining the on-street parking location and additional parking time

The model used in the simulation considers that on-street parking is somehow scarce in the city, meaning that, in general, car-sharing users will not be able to park just in front of their final destinations. This might be due to the high on-street parking demand or due to parking regulations in the city.

For this reason, when the simulator requires to determine a particular on-street parking location, it will generate a random parking location and a random parking time. Both estimations are independent, although they both depend on the percentage of parking demand on the destination zone, which is set by the parameter "percParking" in the general inputs file "inputs.xlsx" (see Section 2.1.6). This parameter is substituted by the analogous "SB_PRK", in case of specific values for different destination zones.

Determining the parking location: it is assumed that users might need to walk larger distances than the ones they regularly accept at the origin, if on-street parking demand is very high at the destination zone and there is no SB option nearby. It is considered that this egress distance will be at most three times the maximum walking distance accepted by users in normal conditions (i.e. parameter "Wmax" in the general inputs file "inputs.xlsx" (see Section 2.1.6). So, the parking location will be generated randomly in a square of area W_{park}^2, centered at the final destination, and following an spatial uniform distribution, where W_{park} is defined as:

$$W_{park} = \min\left\{3 \cdot Wmax, \frac{Wmax}{1 - SB\_PRK}\right\}$$

Note that the likelihood of long egress distances grows with "SB_PRK", meaning that if there is a higher fraction of users who park SB when both options are available, this is indicative that finding an on-street parking spot is harder.

Determining the parking time: Analogously, it is expected that the parking time is higher at zones with higher parking demand. However, this parking time is considered independent from the obtained parking location. Note that users can spend a lot of time looking for available on-street parking by moving around their final destination point. Or, alternatively, they could also find an available spot quickly but further away from their final destination.

Therefore, parking time is randomly generated following an exponential distribution with an average value equal to the average parking time in the city (i.e. parameter "avgParkTime" in the general inputs file "inputs.xlsx"; see Section 2.1.6), weighted by the fraction of parking demand in that zone with respect to the global fraction for the whole service region. This is:

$$Parking\ time \sim Exp\left(avgParkTime \cdot \frac{SB\_PRK\ (in\ the\ zone)}{percParking\ (in\ the\ city)}\right)$$

This weighting takes into account that in zones with higher SB parking demand, parking on-street is harder, meaning that the time to find an on-street parking slot will also be higher. Recall that the global parameter "percParking" must be introduced in the simulator as the average of all the "SB_PRK" parameters introduced in the zonification shapefile. If the zonification is generated by the simulator or the shapefile does not contain the "SB_PRK" property, all zones will be assigned the same average SB parking fraction.

5.2. Users' movement and vehicles' status

This module compares the current simulation time step with the different timer properties of the users (i.e. arriving to the car, reaching the parking point, and reaching their destination) and changes the relevant status of users and assigned vehicles accordingly.

- *Arriving to the car*: When the user "arriving to the car" timer is reached, the car status changes from "reserved" to "in trip". The vehicle and the user location coordinates change to NaN. The car is also removed from the station/zone list, and from the battery charging list (when applicable).

- *Reaching the parking point*: When the user timer "Reaching the parking point" is fulfilled, the user and the assigned car location coordinates change to those of the parking point. The car status also changes as follows:

  o If the assigned car is electric and the battery level is below the minimum, the car status is changed to "discharged".

  o If the car assigned is an ICE car or an electric car with the battery level above the minimum threshold, the car status is changed to "available".

- *Reaching the final destination*: When user last timer is reached, and the user has arrived at his final destination, the user is stored into a finished users' array.

5.3. Calculation of optimal vehicle distribution

The optimal vehicle distribution over the service region is computed every time step. The process is split between the SB and FF systems. So, all available cars on parkings and on-street are accounted separately. The optimal distribution of each group is estimated according to the minimization of the cost of lost demand.

The optimal distribution process is adapted from Jimenez & Soriguera (2020)[2], and in the present simulator consists of the following steps:

1. Consider the available SB vehicle fleet at current time step, t (m_{SB_t}).

2. Assume all stations as empty ($m_{SB\_t(i)} = 0$). Calculate the improvement in the system if one car is added to the station i (i.e. Utility, $\mathcal{U}(+1)_{i,t}$). This is:

$$\mathcal{U}(+1)_{i,t} = Cost\big(m_{SB\_t(i)}\big) - Cost\big(m_{SB\_t(i)} + 1\big)$$

---

[2] Jiménez, E., & Soriguera, F. (2020). A new dynamic repositioning approach for bike sharing systems. *Transportation Research Procedia, 47*, 227-234.

(Note that, since this cost refers to lost demand, adding an additional car must translate into less cost, so that $\mathcal{U}(+1)_{i,t}$ would be positive.)

3. Select the station i with the highest $\mathcal{U}(+1)_{i,t}$, and add one car at station i ($m_{SB\_t(i)} = m_{SB\_t(i)} + 1$). If the station's i capacity is not reached, calculate again the new $\mathcal{U}(+1)_{i,t}$. Repeat step 3 until all available cars are distributed ($\sum_i m_{SB\_t(i)} = m_{SB\_t}$).

4. Repeat the whole process again with changing SB by FF free-floating zones.

Note that it could happen that all vehicles are reserved or unavailable (i.e. $m_{SB\_t} = 0$). So, in this case, the optimal distribution will not be updated at that time step.

5.4. Assignment of repositioning tasks

This module creates the repositioning tasks and assigns them optimally to idle repositioning teams. The process of task creation and assignment has the following steps:

1. *Check if there are idle repositioning teams*. The module is executed only if there at least one idle repositioning team waiting for a task assignment.

2. *Construct the pool of prioritized origin tasks (i.e. pick-up cars). Construct the pool of prioritized destination tasks (i.e. drop-off cars).* The definition of the priorities of the possible repositioning tasks will be provided in the next Section 5.4.1.

3. *Calculate the utility of all repositioning tasks.*

   a. *Origin tasks:* the utility $\mathcal{U}(-1)_{i,t}$ is defined as the cost of removing one vehicle from the origin zone/station i corresponding to the task. $\mathcal{U}(-1)_{i,t}$ is computed as:

   $$\mathcal{U}(-1)_{i,t} = Cost\big(m_{t(i)}\big) - Cost\big(m_{t(i)} - 1\big)$$

   Where $m_{t(i)}$ is the number of vehicles at zone/station i and time step t. Note that, since this cost refers to lost demand, removing one car must translate into a higher cost, so that $\mathcal{U}(-1)_{i,t}$ would be negative.

   b. *Destination tasks*: the utility $\mathcal{U}(+1)_{j,t}$ is defined as the cost of adding one vehicle to the destination zone/station j corresponding to the task. $\mathcal{U}(+1)_{j,t}$ is computed as:

   $$\mathcal{U}(+1)_{j,t} = Cost\big(m_{t(j)}\big) - Cost\big(m_{t(j)} + 1\big)$$

   Note that, since this cost refers to lost demand, adding one car must translate into a smaller cost, so that $\mathcal{U}(+1)_{j,t}$ would be positive.

4. *Define the total generalized cost, $C_{i,j,t}$, of each possible repositioning task (i.e. moving one vehicle from the origin i to the destination j of the task at current time step t). $C_{i,j,t}$ is defined as:

   $$C_{i,j,t} = \mathcal{U}(-1)_{i,t} + \mathcal{U}(+1)_{j,t} + C_{d_{i,j}}$$

   $C_{d_{i,j}}$ is de cost of the repositioning team moving from the origin i to the destination j of the task, obtained as:

$$C_{d_{i,j}} = C_t \frac{d_{i,j}}{v_k}$$

Where C_t is the cost of one repositioning team per unit time, $d_{i,j}$ is the distance from origin i to the destination j of the task, and v_k is the travelling speed of the repositioning team (see Section 2.1.7).

5. *Repositioning task creation by minimum generalized cost pairwise matching*. Assign origin tasks to destination tasks so that the sum of the generalized costs of all tasks, $C_{i,j,t}$, is minimum. If there remain some origin or destination tasks unassigned, they are discarded. Note that the number of origin and destination tasks may be different.

6. *Check the location, k, of all repositioning teams*. If one repositioning team is idle, its location will be the current position. If the repositioning team is busy, its location will be the end position of the last task assigned (i.e. where the repositioned car was dropped-off).

7. *Calculate the cost of moving repositioning teams to each possible origin of the tasks, $C_{d_{k,i}}$*. This is computed as the cost of moving a repositioning team from its considered location, k, to the origin of the repositioning task, i.

8. *Repositioning team assignment by minimum generalized cost pairwise matching.* Assign repositioning teams to tasks by minimizing the sum of all generalized costs, $C_{k,i,j,t}$, including the movement of repositioning teams to origin of tasks, and the generalized cost of the task. $C_{k,i,j,t}$ is defined as:

$$C_{k,i,j,t} = C_{d_{k,i}} + C_{i,j,t}$$

9. Tasks which are assigned to busy repositioning teams at time step t are discarded.

10. Tasks which are assigned to idle repositioning teams at time step t, are assigned to them. The origin task and the destination task are added into the assigned tasks list of the repositioning team. These tasks are represented by the ID code of the station or zone in which the car is picked/returned.

5.4.1. Determining repositioning task priorities

Repositioning tasks within the car-sharing system aim to two different objectives. On the one hand, repositioning must move discharged electrical vehicles from the streets to stations with available battery chargers. These operations are considered to have the highest priority in the system. On the other hand, repositioning operations must also balance the system, moving vehicles from where there are in excess to where there is a deficit. To that end, 3 priority levels are considered when defining the pool of repositioning tasks.

- *Priority 1: Battery recharging + balancing the system:* The first priority level includes those tasks which allow recharging the battery of electrical vehicles while at the same time rebalancing the system. For this Priority 1, the pool of origin tasks will be composed of all electrical cars whose battery level is below the minimum threshold (i.e. status is "discharged"). The utility $\mathcal{U}(-1)_{i,t}$ of these tasks will be set by default to maximum (i.e. 99999 €). The pool of destination tasks for the Priority 1 level will be composed only of stations with available battery chargers and with an inventory level below the optimum at time t. The number of destination tasks included in the pool for each station fulfilling

the previous requisite will be equal to the difference between the optimal and the actual inventory level. The utility for them is the expected revenue for locating the next extra car on that station, $\mathcal{U}(+1)_{j,t}$.

- *Priority 2: Battery recharging.* The second priority level only aims to reposition all the remaining discharged electrical vehicles left over after Priority 1. This priority considers again as origin tasks only the "discharged" electrical cars with same maximum utility for the task as in Priority 1. The difference is that Priority 2 does not aim to balance the system, and destination tasks are directly assigned to the closest station with available battery chargers. So, all recharging tasks will be assigned after Priority 2 if there are enough repositioning teams available.

- *Priority 3: System rebalancing.* The third priority level only aims to system rebalancing, as all the recharged vehicles must have been assigned in previous priority levels. Origin tasks in Priority 3 are stations/zones with a number of cars above the optimum inventory level at time t. The number of origin tasks per station/zone is the difference between the actual and the optimal inventory level. Utility is the expected revenue loss for removing the next car, $\mathcal{U}(-1)_{i,t}$. Destination tasks in Priority 3 level are stations/zones with a number of cars below the optimal inventory level. The number of tasks per station /zone is the difference between the optimal and the actual inventory levels. The utility for the destination task is the expected revenue for locating the next extra car on that station/zone, $\mathcal{U}(+1)_{j,t}$. In addition, if the number of FF cars on-street is over the maximum threshold considered by the policy makers, origin tasks will be restricted to FF vehicles in zones, and destination tasks will be considered only SB in stations. Otherwise, all tasks are considered.

Every time step with idle repositioning teams, the repositioning task assignment process is executed sequentially with task pools from Priority 1, Priority 2 and Priority 3 until there are not idle repositioning teams remaining.

5.5. Movement of the repositioning teams

Like users, repositioning teams' properties include time counters and the status. When the simulation time step exceeds a particular time counter, the repositioning team will perform an action that will depend on his status. Possible status for repositioning teams are: "idle", "moving with scooter to the car", and "returning a car". The actions performed in each status are described next:

- *Idle repositioning team*: Idle teams do not perform any actions. They just wait until tasks are assigned to them. They trigger the repositioning task assignment module.

- *Repositioning team moving with scooter to a car*:

  o Once the "idle" repositioning team is assigned one task, his status will change to "moving with scooter to the car".

  o When the time counter for this status is exceeded, it means that the repositioning team has arrived to the assigned car. Then, it changes his status to "returning a car" and the repositioning team is changed to the final location

of the destination task. If the destination is a free-floating zone, the considered position will be the centroid of the zone. The car position is changed to NaN.

- o If there are no cars on the station/zone (i.e. there was much more demand than expected and reserved them), the team aborts the returning tasks and becomes idle.

- *Repositioning team returning a car*:

    - o Once the time counter for the status "returning a car" is exceeded, it means that the repositioning team has dropped-off the car at the destination. The repositioning team status becomes idle.

    - o The car is included into the station/zone list. The car position changes to the destination point of the task. If the vehicle is electric and the battery level is below the minimum threshold, the car status changes to "discharged", otherwise, it changes to "idle".

5.6.   Electrical vehicle battery recharging

This module aims to simulate the battery recharging operations of electrical vehicles. This includes two processes: i) creating and updating the battery chargers lists at stations, and ii) update the battery level of electrical vehicles being recharged.

Charging lists at stations: These lists represent how many electric cars are connected to battery chargers in the station. Then, every simulation time step:

- If the battery level of any electrical car in the station list is 100%, this car is removed from the charging list.

- The electric cars with lowest battery level are included in the list until there are as many cars in the list as the number of chargers in the station.

Updating battery level of electrical vehicles: Battery level of electric cars is updated according to the following steps:

- Battery level of electrical cars with a "travelling" or "repositioning" status is reduced every time step by the proportional percentage for 1 minute of use. If battery level reaches zero, it stays at zero.

- Battery level of electrical cars included on any stations battery charging list is increased by the proportional percentage for 1 minute of recharge.

6. Outputs from the simulator

The generation of outputs and results from the simulator is an independent process from the simulation itself. This allows setting different outputs without the need of running the simulation again. In addition, it allows modifying the structure of contents of the output files, without interfering with the main modules of the simulator.

Given this context, the folder and file structure of the outputs program for the simulator consists of:

- **The "outputs.xlsx" MS Excel file:** This is the "output selection" input file. The file is located in the "Data" folder. It contains all the required inputs to set up the desired outputs

- **"GenerateOutput.m" script file:** This file runs the result post process code, generating all the results files.

- **"Results" folder:** In this folder, all results will be stored. If this folder does not exist, it will be created after the first simulation run.

6.1. The output selection input file ("outputs.xlsx")

The MS Excel file "outputs.xlsx" allows selecting which results will be returned after running the simulation postprocess program. Like all the input files, **the values of all variables are editable, but the file "outputs.xlsx" cannot be renamed of moved away from the folder "Data"**. The file is structured in several groups of inputs:

6.1.1. General output parameters

This group includes the general parameters defining where to save the results, and if graphical plots and additional information is desired.

Table 16. General result postprocessing parameters.

Property	Description
ResFile	Path to the results file from simulation (.mat file)
GeneratePlots	*GeneratePlots* = 0 => No graphical plots *GeneratePlots* = 1 => Generate graphical plots for every table
Verbose	*Verbose* = 0 => No additional information during the results post-process *Verbose* = 1 => Show additional information about the simulation

6.1.2. Selection of the detail level in the outputs

The inputs in this group allow the user to enable or disable different levels of aggregation in the results (see Table 17 and Table 18).

The maximum level of detail in the outputs of the simulation is to obtain results every hour for every station (i.e. SB system) and for every free-floating zone (i.e. FF system). Also, the global results for both systems can be obtained (i.e. results for each free-floating zone including all the stations inside the zone). If the "Hourly" option is activated, results are returned with this maximum level of detail.

These results could be aggregated for the total simulation time (e.g. 24 hours), but still distinguishing between stations/zones. This aggregation is selected by activating the time aggregation option, "AggT".

Alternatively, results could be aggregated in space, for the whole service area, but still distinguishing between hourly periods. This space aggregation is selected by activating the option "AggR".

Finally, by default, the simulator always yields an aggregated summary of results for the whole service region and for the whole simulation period. A template with relevant explanations of this results' summary report is presented in Section 6.2.4.

Table 17. Result aggregation options.

Property	Description
Hourly	*Hourly* = 0 => No hourly output *Hourly* = 1 => Generate results hourly
AggT	*AggT* = 0 => No time aggregation output *AggT* = 1 => Generate results aggregated by time
AggR	*AggR* = 0 => No space aggregation output *AggR* = 1 => Generate results aggregated by region

Table 18. Outputs from different aggregation options.

Notation	Description	Time	Space	Numerical result (for each variable)	Graphical result (for each variable)
"Hourly"	Detailed	Hour	Zone Station	Matrix $dim = TxN$	T x Contour plots
"AggT"	Space aggregated	Hour	Whole service region	Vector $dim = T$	1 x Time series
"AggR"	Time aggregated	Whole simulation time	Zone Station	Vector $dim = N$	1 x Contour plot
Summary		Whole simulation time	Whole service region	Number	-

Note: T is the duration in hours of the simulation time; N is the number of zones / stations in the service region.

6.1.3. Selection of the desired categories of results

With these options, the user can select which categories of results he desires to obtain. **If a category is selected, all results in the category will be generated**. In all categories, results for the SB and FF systems are generated separately. Global results (i.e. aggregating both the SB and the FF system) are also provided. The only exceptions are the "car status" in the "vehicle" category which is returned as percentages over the total fleet, and the "station occupancy"

which only considers stations. Further details about the detailed description and computation process of the results in these categories are presented in Section 6.1.4.

<p align="center">**Table 19.** Result category options.</p>

Category	Description

Vehicles	*Vehicles* = 0 => No output in the "Vehicles" category
	
	Vehicles = 1 => Summary of results about vehicles:
	
	    -Average number of available cars in stations/street/total. Status of cars is idle (including vehicles which are recharging batteries). [cars]
	
	    -Average number of cars in each status (i.e. idle, reserved, traveling, repositioning, to be recharged) [cars]

BatteryLevel	*BatteryLevel* = 0 => No output in the "BatteryLevel" category *BatteryLevel* = 1 => Summary of results about battery levels: -Average battery level of cars in stations/street/total [%]

StationOccupancy	*StationOccupancy* = 0 => No output in the "StationOccupancy" category
	
	StationOccupancy = 1 => Summary of results about station occupancy:
	
	    -Empty parking slots [slots] and station occupancy [%].
	
	    -Empty chargers [chargers] and charging occupancy [%].

Unbalance	*Unbalance* = 0 => No output in the "Unbalance" category *Unbalance* = 1 => Summary of results about unbalance: -Average vehicle unbalance in stations/street/total [cars]

Demand	*Demand* = 0 => No output in the "Demand" category
	
	Demand = 1 => Summary of results about demand:
	
	    -Number of trips generated at stations/street/total. [trips]
	
	    -Number of trips finished at stations/street/total. [trips]
	
	    -Number of unbalanced trips at stations/street/total. [trips]
	
	    -Number of lost trips (total: both systems together). [trips]
	
	    -Average access / egress distance at stations/street/total. [m]
	
	    -Total revenue of trips at stations/street/total. [€]

Repositioning	*Repositioning* = 0 => No output in the "Repositioning" category
	
	Repositioning = 1 => Summary of results about repositioning:
	
	    -Number of Priority 1 & 2 repositioning movements (i.e. electrical vehicles with discharged battery) taken / left from stations/street/total. [cars]
	
	    -Number of Priority 3 repositioning movements (i.e. system rebalancing) taken / left from stations/street/total. [cars]

6.1.4. Description of the output categories of results

6.1.4.1. Vehicles

This category contains two types of results related with vehicle number and status:

- *"Cars":* Accounts for the average number of <u>available</u> (i.e. status = idle, recharging) cars at a given station/zone/total over the considered time period. Recall that "total" means the number of available vehicles in the FF zone plus in the stations contained in this zone. Note that, since the minimum time period lasts 60 minutes, this average number of cars might be a non-integer value. For example, if during the 60-minute period a station has one available car during 30 minutes, and zero in the rest of the hour, this result will result as 0.5 average cars available in the station during that period. Units are [cars].

- *"CarStatus":* Accounts for the average number of idle, reserved, traveling, repositioning, and to be recharged cars over the considered time period. Since some of the status cannot be assigned to a particular system or location (i.e. moving cars are nor FF neither SB, and during the movement their location is undefined), this result does not consider any spatial discretization or split by system type. It accounts for the whole system fleet and service region. Units are [cars]. in the output summary file this variable is expressed as a [%] of the total vehicle fleet.

Additionally, there are three more aggregated results in the output summary file:

- Average number of e-cars in battery recharging mode at any time. Expressed as a [%] of the total <u>electric</u> vehicle fleet.

- Average distance travelled per vehicle and hour, without considering repositioning trips. Expressed in [km/veh·h].

- Average number of trips per vehicle and hour, without considering repositioning trips. Expressed in [trips/veh·h].

6.1.4.2. Battery Levels

This category only contains one result related to the battery level of electrical vehicles.

- *"Battery":* Accounts for the average battery percentage of electric cars at the station/zone/total and considered time period. Note that this result does not consider ICE cars or vehicles which are currently moving cars. To avoid confusions, if there are no electric cars in the selected region or time period, an empty value is returned instead of a zero.

6.1.4.3. Station Occupancy

The simulation model does not consider any parking capacity limitation at FF zones. In addition, it is assumed that charging infrastructure it is only deployed at stations. Therefore, results in this category, related to stations' and battery chargers' occupancy, only apply to the SB system.

There are four types of results in this category:

- *"Empty Slots":* Accounts for the average number of empty parking slots at a given station over the considered time period. If the capacity of the station is set as infinity (i.e. input 99999), no result is returned. Units are [slots].

- *"Occupancy":* Accounts for the average percentage of occupied parking slots at a given station over the considered time period. If the capacity of the station is set as infinity (i.e. input 99999), no result is returned. Expressed in [%] of station's capacity.

- *"Empty Chargers":* Accounts for the average number of empty battery charging slots at a given station over the considered time period. If the number of chargers in the station is set as infinity (i.e. input 99999) or zero, no result is returned. Units are [chargers].

- *"Occupancy Chargers":* Accounts the average percentage of occupied battery charging slots on a given station over the considered time period. If the number of chargers in the station is set as infinity (input higher than 9999) or zero, no result is returned. Expressed in [%] of station's battery charging capacity.

Note that in case of spatially aggregated results for this category (i.e. "*AggR* = 1") only the stations with a defined capacity or number of battery chargers (i.e. different than zero or infinity) will be considered. In case of not having any of them, no result is returned.

6.1.4.4. Unbalance

This category only contains one type of result in relation to the car-sharing system unbalance:

- *"Unbalance":* System unbalance is defined as the average difference in the time period considered between the actual vehicle inventory level minus the optimal inventory level at the station/zone/total. Therefore, system unbalance is positive if there are vehicles in excess, and negative if there is a deficit of vehicles. Note that the vehicle inventory level only includes available vehicles (i.e. "idle" and "recharging" status). Units are [cars].

6.1.4.5. Demand

The "Demand" category contains up to six types of results related to how the demand is served by the car-sharing system. These are:

- *"Demand Origin":* Accounts for the total number of trips generated at a given station/zone/total over the considered time period. One trip is generated when a car has been assigned. Units are [trips].

- *"Demand Destination":* Accounts for the total number of trips attracted by a given station/zone/total over the considered time period. One trip is attracted when a car has been assigned a parking spot at the considered location. Units are [trips].

- *"Demand Unbalance":* Accounts for the total number of unbalanced trips at the station/zone/total over the considered time period. Demand unbalance is defined as the average difference in the time period considered between the number of vehicle returns and the number of vehicle requests at the station/zone/total. Therefore, demand unbalance is positive if there are more returns than request, and negative if there are more requests than returns. Units are [trips].

- *"Lost trips":* Accounts for the unserved demand in the whole system (i.e. total; FF and SB systems together) over the considered time period. Recall that the unserved demand

cost in the output summary is calculated considering different penalty values if the user had a station nearby (SB) or not (FF) (See Section 6.1.5). Units are [trips]. In the output summary file "Lost trips" are also expressed as a [%] over the total potential demand.

- *"Access Distance":* Accounts for the average access distance of all trips generated at a given station/zone/total over the considered time period. Units are [m].

- *"Egress Distance":* Accounts for the average egress distance of all trips attracted at a given station/zone/total over the considered time period. Units are [m].

- "*Fare Trips":* Accounts for the total revenue generated by all trips starting from a given station/zone/total over the considered time period. This result needs as an input the unitary fare for each trip (See Section 6.1.5). Units are [€].

Additionally, three more aggregated results belonging to the demand category are included in the output summary file:

- *Average usage time per trip*. Only including the time the vehicle is moving (i.e. status is "traveling"). Considering all the trips finished over the service region and in the total simulation time. Units are [minutes].

- *Average distance per trip*. Considering all the trips finished over the service region and in the total simulation time. Units are [km].

- *Average vehicle reserved time per trip*. This is the time elapsed between the vehicle is assigned to a user, and the user actually arrives to the vehicle. Considering all the trips started over the service region and in the total simulation time. Units are [minutes].

6.1.4.6. Repositioning

This category contains up to four types of results related to the repositioning operations.

- *Number of origin repositioning tasks* (i.e. vehicle taken) at a given station/zone/total over the considered time period. Units are [cars].

- *Number of destination repositioning tasks* (i.e. vehicle left) at a given station/zone/total over the considered time period. Units are [cars].

The previous results are split between Priority 1 & 2 repositioning operations (i.e. battery recharging) and Priority 3 operations (i.e. rebalancing movements).

In addition, five more aggregated results are included in the output summary file. These are:

- *Average repositioning rate in the system*. Average number of cars repositioned per hour over the service region and in the total simulation time. Units are [operations/h].

- *Average repositioning rate per team*. Total number of cars repositioned per hour and team over the service region and in the total simulation time. Units are [operations/h·team].

- *Average percentage of time the repositioning teams are in each status*. Average percentage of time the repositioning teams are travelling with scooter, returning the car, and idle. For the complete service region and in the total simulation time. Expressed as a [%]

- *Percentage of each task type*. Percentage of the total operations performed belonging to Priority 1 & 2 (i.e. battery recharging) and Priority 3 s (i.e. rebalancing movements). Expressed as a [%].

- *Percentage of tasks according to the type of Origin – Destination.* The origin / destination of any repositioning task can be a station (i.e. SB system) or a free-floating zone (i.e. FF system). This result accounts for the percentage of tasks with the following origin – destination systems: FF=>FF, FF=>SB, SB=>FF, SB=>SB. Expressed as a [%].

6.1.5. Unitary cost and auxiliary outputs parameters

Table 20 and Table 21 present the required unitary cost parameters, used to calculate the cost and revenue metrics by monetizing the outputs of the simulator, and some additional auxiliary parameters. These auxiliary parameters are needed because the simulation and results postprocess are independent programs, and **some parameters need to be listed twice in the "inputs.xlsx" and in the "output.xlsx" files**. User should check that both files use the same values. Details on the default value estimation of these parameters are included in the appendix to this user's manual (i.e. APPENDIX – Default parameter value estimation process).

Table 20. Auxiliary users' parameters for the result generation.

Parameter	Description	Default Value
WalkSpeed (v_w)	Average walking speed in km/h. (A default value of 3km/h is suitable in most cases).	3 km/h
CarSpeed (v)	Average car speed while traveling within the city [km/h]	15.3 km/h
costTimeUser (β)	Average users' value of time when accessing the system. Perceived cost of time lost on access and egress.	11.40 €/h
costLostFF (γ_{lost_FF})	FF Lost demand penalty. Perceived cost of not finding a car on-street. In €.	2.50 €/trip
costLostSB (γ_{lost_SB})	SB Lost demand penalty. Perceived cost of not finding a car in stations. In €.	7.93 €/trip

Table 21. Auxiliary agency parameters for the result generation (unitary costs).

Parameter	Description	Default Value
costCarElectric $(\gamma_{electric})$	Prorated cost of vehicle fleet acquisition and insurance, for electric vehicles. It includes the cost of additional stock needed to cover vehicle unavailability due to maintenance and repairs.	0.33 €/car·h
costCarICE (γ_{ICE})	Prorated cost of vehicle fleet acquisition and insurance, for ICE vehicles. It includes the cost of additional stock	0.23 €/car·h

	needed to cover vehicle unavailability due to maintenance and repairs.	
costStat (γ_{STAT})	Prorated cost per station. This cost includes the possibility of having a fixed time cost per station, because of the agreement to participate in the car-sharing system.	0 €/station·h
costParkingSB (γ_{S_SB})	Prorated cost (i.e. renting) of one parking spot in a station. This cost does not consider any penalty for allowing to park in different stations.	0.25 €/slot·h
costParkingFF (γ_{S_FF})	Prorated cost (i.e. renting) of one parking spot on-street.	0€/slot·h
costCharger ($\gamma_{S_{CH}}$)	Prorated cost (i.e. renting) of an electrical vehicle battery charger. To be added to the prorated cost of the parking spot in a station.	0.05 €/slot·h
costOperative (γ_a)	Administrative and control costs prorated per vehicle.	0.24 €/h·veh
costMaintCar (γ_m)	Vehicle maintenance cost prorated per 100 km.	4.39 €/100km
costFuelElectric (γ_{c_ICE})	Energy consumption cost prorated per 100 km. For ICE cars.	6.47 €/100km
costFuelICE ($\gamma_{c\_electric}$)	Energy consumption cost prorated per 100 km. For electric cars.	3.58 €/100km
costRepo (C_t)	Labor and equipment cost of repositioning employees prorated per hour.	21.54 €/h
avgFare (F)	Average car-sharing fare.	0.27 €/min

6.2. The "Results" folder

All the simulation outputs are saved in the "Results" folder. For each simulation and postprocessing process, three elements are generated. The name of all three elements automatically includes the prefix "YYYYMMDD_HHMMSS", where YYYYMMDD and HHMMSS are the date and time of the simulation, respectively.

The three main elements generated by the simulation in the "Results" folder, and which are later used for postprocessing are:

- The Matlab file ("YYYYMMDD_HHMMSS_Final_City_Variables.mat"), which contains all the elements and information of the performed simulation in Matlab format. This file is the main input for the postprocessing process (see Section 1.3).

- A folder named "YYYYMMDD_HHMMSS_Final_City_Variables_input". This folder is generated after running the simulation. It contains a compilation of figures and text files summarizing the introduced inputs. In particular, the incuded files are:

  - The text file "YYYYMMDD_HHMMSS_rawInputs.txt". This text file recaps all the inputs used in the simulation, for later reference.

  - The text file "YYYYMMDD_HHMMSS_aggDemandParameters.txt". This text file recaps all the aggregated demand parameters that characterize the O/D demand matrices used in the simulation.

  - A figure called "DemandHistogram.jpg". It depicts the total number of potential trips for each demand matrix period.

  - A figure called "Zonification.jpg", depicting the service area zonification and their ID number.

  - The list of stations actually used. (By default, called "stations_out.xlsx")

  - Three demand map figures ("DemandRequests.jpg", "DemandReturns.jpg" and "DemandImbalance.jpg"). They depict the spatial distribution of the potential demand at origin (i.e. requests), demand at destination (i.e. returns) and demand imbalance (i.e. returns-requests), in order to visualize which zones have more demand generation and attraction.

  - The generated O/D demand matrices in the folder "custom_OD_mat".

  - The generated zonification shapefiles in the folder "custom_shp".

- A folder named "YYYYMMDD_HHMMSS_Final_City_Variables_output". This folder is generated after running the results' postprocess. It contains several subfolders according to each category of results and the result summary file. In particular, they are:

  - The file "Table_summary.xlsx". This file is the result summary file and it includes tables of KPIs and costs associated to the simulation run.

  - Up to six subfolders, one for each result category (as considered and described in Section 6.1.3). Each one of these subfolders will include an excel file ("Tables_categoryname.xlsx") with all the results and their respective representation in a figure (if selected).

The next sections describe each one of these elements in more detail.

6.2.1. The "YYYYMMDD_HHMMSS_Final_City_Variables.mat" Matlab file

The "YYYYMMDD_HHMMSS_Final_City_Variables.mat" is a Matlab object file, which contains an object of the class "City". This object is a container of all stored agents, elements, and properties of the simulation (See Section 3.3.7).

This Matlab element allows generating all the possible outputs with any desired detail level. Therefore, this "City" element is the main input for the results' postprocessing and it allows to run different results generations without running again the simulation.

This file can be renamed (i.e. its path is set by the parameter "ResFile" in the output selection input file "outputs.xlsx", see Section 6.1), but then, the following results generated will be created in a new folder with the new name.

6.2.2. The "YYYYMMDD_HHMMSS_Final_City_Variables_input" folder
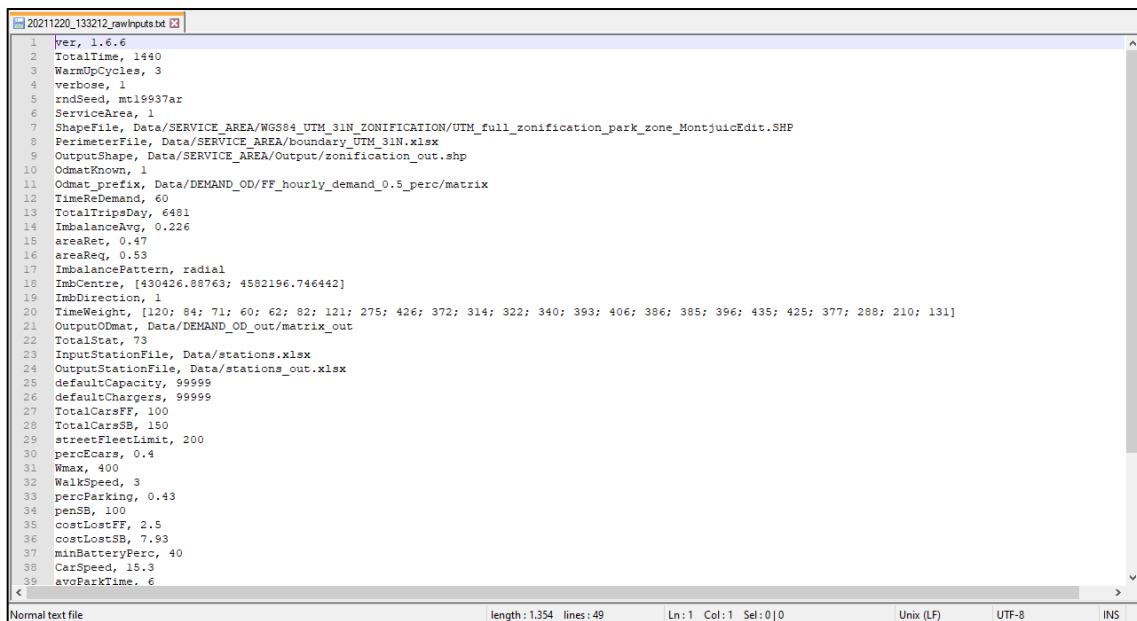
This folder contains a recap of the used inputs in the simulation. It is automatically generated after a simulation run, without need of a postprocess. It includes the following files.

6.2.2.1. The "YYYYMMDD_HHMMSS_rawInputs.txt" text file

The "YYYYMMDD_HHMMSS_rawInputs.txt" is a text file which recaps all the inputs used in thee simulation. The name of the file will be always follow the same structure, where YYYYMMDD and HHMMSS are the date and time of the simulation, respectively.

In addition, this text file will include:

- The random seed used, even if this parameter was left blank in the general inputs file

- The minimum battery level considered, even if the option 0 (i.e. automatic calculation) was selected in the general inputs file (See Section 4.2 for more information about this estimation).


```
20211220_133212_rawInputs.txt
1   ver, 1.6.6
2   TotalTime, 1440
3   WarmUpCycles, 3
4   verbose, 1
5   rndSeed, mtl9937ar
6   ServiceArea, 1
7   ShapeFile, Data/SERVICE_AREA/WGS84_UTM_31N_ZONIFICATION/UTM_full_zonification_park_zone_MontjuicEdit.SHP
8   PerimeterFile, Data/SERVICE_AREA/boundary_UTM_31N.xlsx
9   OutputShape, Data/SERVICE_AREA/Output/zonification_out.shp
10  OdmatKnown, 1
11  Odmat_prefix, Data/DEMAND_OD/FF_hourly_demand_0.5_perc/matrix
12  TimeReDemand, 60
13  TotalTripsDay, 6481
14  ImbalanceAvg, 0.226
15  areaRet, 0.47
16  areaReq, 0.53
17  ImbalancePattern, radial
18  ImbCentre, [430426.88763; 4582196.746442]
19  ImbDirection, 1
20  TimeWeight, [120; 84; 71; 60; 62; 82; 121; 275; 426; 372; 314; 322; 340; 393; 406; 386; 385; 396; 435; 425; 377; 288; 210; 131]
21  OutputODmat, Data/DEMAND_OD_out/matrix_out
22  TotalStat, 73
23  InputStationFile, Data/stations.xlsx
24  OutputStationFile, Data/stations_out.xlsx
25  defaultCapacity, 99999
26  defaultChargers, 99999
27  TotalCarsFF, 100
28  TotalCarsSB, 150
29  streetFleetLimit, 200
30  percEcars, 0.4
31  Wmax, 400
32  WalkSpeed, 3
33  percParking, 0.43
34  penSB, 100
35  costLostFF, 2.5
36  costLostSB, 7.93
37  minBatteryPerc, 40
38  CarSpeed, 15.3
39  avgParkTime, 6
```

Figure 8. Example of the "YYYYMMDD_HHMMSS_rawInputs.txt" text file.

6.2.2.2. The "YYYYMMDD_HHMMSS_aggDemandParameters.txt" text file

The "YYYYMMDD_HHMMSS_aggDemandParameters.txt" text file is a compilation of aggregated values that summarize and characterize the demand input. All values are calculated statistically from the O/D matrices used by the simulator, either if they are user-provided or custom-generated. Note that, in the latter case, these values could be slightly different from the aggregated input values.

The following values are included. By notation, i, j are zone indexes belonging to the service area ($i, j \in R$). And t is a time index defining each matrix. Therefore, matrix elements are defined as $n_{i,j,t}$, and the area of each zone is r_i.

- The total number of potential trips during the demand cycle, N. They are calculated as the sum of all elements of all O/D matrices.

$$N = \sum_{\forall i} \sum_{\forall j} \sum_{\forall t} n_{i,j,t}$$

- The total area of the service region, R.

$$R = \sum_{\forall i} r_i$$

- The average potential demand density, λ. Computed as the total number of trips during the demand cycle, N, divided by the product of the demand cycle duration, T, and service area, R. This is:

$$\lambda = \frac{N}{R \cdot T}$$

Note that demand density can be different at different times of the day, resulting in:

$$\lambda_t = \frac{N_t}{R \cdot TD}$$

Where $N_t = \sum_{\forall i} \sum_{\forall j} n_{i,j,t}$, and TD is the duration of each matrix.

- The fractions of the service area in which there are more car requests than returns, π_q (i.e. generating areas), and more car returns than requests, π_t (i.e. attracting areas).

In order to compute these fractions, first it is necessary to define the total number of requests and returns per zone and time period:

$$n_{q(i,t)} = \sum_{\forall j} n_{i,j,t} \qquad n_{t(j,t)} = \sum_{\forall i} n_{i,j,t}$$

Then, the unbalance is calculated as the difference between returns and requests and each zone is assigned to the corresponding subregion.

$$n_{t(i,t)} - n_{q(i,t)} > 0 \quad \rightarrow i \in R_{t(t)}$$

$$n_{t(i,t)} - n_{q(i,t)} < 0 \quad \rightarrow i \in R_{q(t)}$$

Each fraction is computed as the aggregated area of each subregion type divided by the total service area.

$$\pi_{t(t)} = \frac{\sum_{i \in R_{t(t)}} r_i}{R}$$

$$\pi_{q(t)} = \frac{\sum_{i \in R_{q(t)}} r_i}{R}$$

Finally, the global average is obtained as a weighted average where the weights are the demand density at each time period:

$$\pi_t = \frac{\sum_{\forall t}\left(\pi_{t(t)} \cdot \lambda_t\right)}{\sum_{\forall t}\lambda_t}$$

$$\pi_q = \frac{\sum_{\forall t}\left(\pi_{q(t)} \cdot \lambda_t\right)}{\sum_{\forall t}\lambda_t}$$

- The average fraction of imbalanced trips, Φ. The process of calculation of this value is split between both subregions R_q and R_t. For each one of them the hourly average demand unbalance fraction ($\Phi_{q(t)}$ and $\Phi_{t(t)}$) is the difference between the hourly attracted and generated trips, divided by the hourly demand in the subregion. This is:

$$\Phi_{q(t)} = \frac{\sum_{i \in R_{q(t)}}\left(n_{t(i,t)} - n_{q(i,t)}\right)}{R \cdot \pi_{q(t)} \cdot \lambda_t}$$

$$\Phi_{t(t)} = \frac{\sum_{i \in R_{t(t)}}\left(n_{t(i,t)} - n_{q(i,t)}\right)}{R \cdot \pi_{t(t)} \cdot \lambda_t}$$

By definition, $\Phi_{q(t)}$ is negative and $\Phi_{t(t)}$ is positive.

Then, the global average unbalance fractions can be calculated considering the hourly demand density and the subregion areas as weighting factors.

$$\Phi_q = \frac{\sum_{\forall t}(\Phi_{q(t)} \cdot \pi_{q(t)} \cdot \lambda_t)}{\sum_{\forall t}\left(\pi_{q(t)} \cdot \lambda_t\right)}$$

$$\Phi_t = \frac{\sum_{\forall t}(\Phi_{t(t)} \cdot \pi_{t(t)} \cdot \lambda_t)}{\sum_{\forall t}\left(\pi_{t(t)} \cdot \lambda_t\right)}$$

Finally, the average demand unbalance fraction is calculated as:

$$\Phi = \Phi_t \cdot \pi_t + \left|\Phi_q\right| \cdot \pi_q$$

- The temporal coefficient of variation of demand, $CV_{temp(\lambda)}$. This is calculated as the standard deviation of the demand densities, λ_t, obtained from O/D matrices at time t, normalized by the average demand density.

$$CV_{t(\lambda)} = \frac{\sigma_t(\lambda_t)}{\lambda}$$

- The spatial coefficient of variation of demand, $CV_{s(\lambda)}$. In order to calculate this parameter, we define the dimensionless imbalance at each zone and time period as:

$$\varphi_{i,t} = \frac{n_{t(i,t)} - n_{q(i,t)}}{\lambda_t \cdot r_i \cdot TD}$$

Where TD is the fixed duration of time periods t.Then, the standard deviation is calculated for each matrix $\left[\varphi_{i,t}\right], \forall i, t$ and weighted by the demand density at period t. Note that the obtained result is already normalized.

$$CV_{s(\lambda)} = \frac{\sum_{\forall t} \sigma_t(\varphi_{i,t}) \cdot \lambda_t}{\sum_{\forall t} \lambda_t}$$

6.2.2.3. O/D demand matrices characterization figures

In total, there are five figures that characterize temporally and spatially the demand matrices used during the simulation. In particular, these figures are:

- *A time series plot of the hourly potential demand evolution* in the whole service region and complete demand cycle. Note that time series bars will be associated with the duration of each O/D demand matrix (e.g. 60 minutes). See one example in Figure 9.

- *Map of the service region* with the zonification and stations' location. (Filenames: "Zonification.jpg" and "Stations.jpg"). See one example in Figure 10.

- Three contour plot maps representing the following demand variables for each zone and for the whole simulation time. See one example in ***Figure 11***.

    o Total potential trip generation per zone. (Filename: "DemandRequests.jpg")

    o Total potential trip attraction per zone. (Filename: "DemandReturns.jpg")

    o Total potential trip imbalance, which is the result of the potential trip attraction minus potential trip generation. (Filename: "DemandImbalance.jpg")
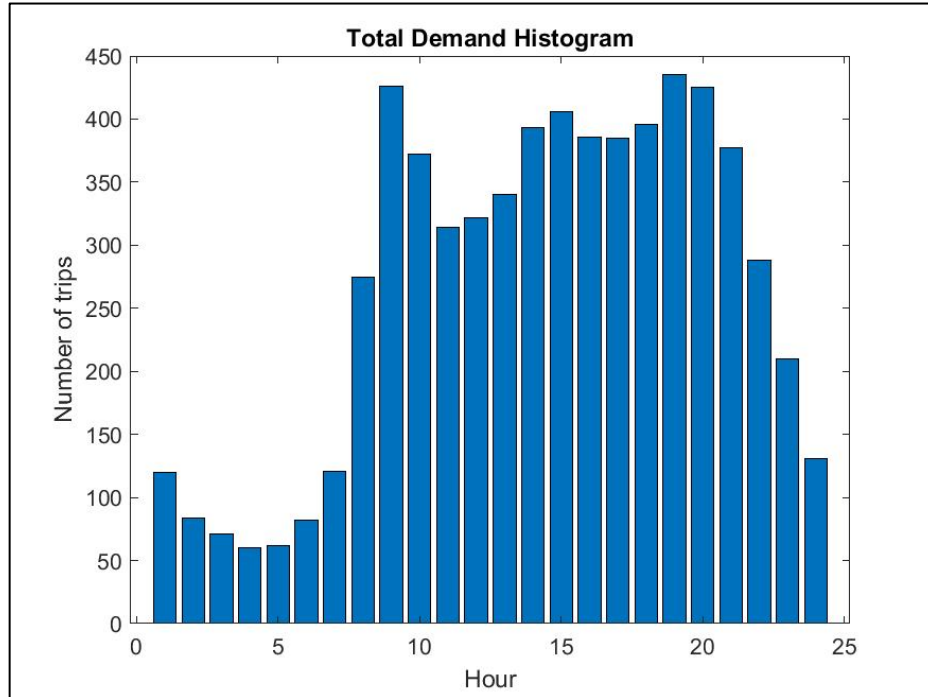
Figure 9. *Example of demand time series plot.*
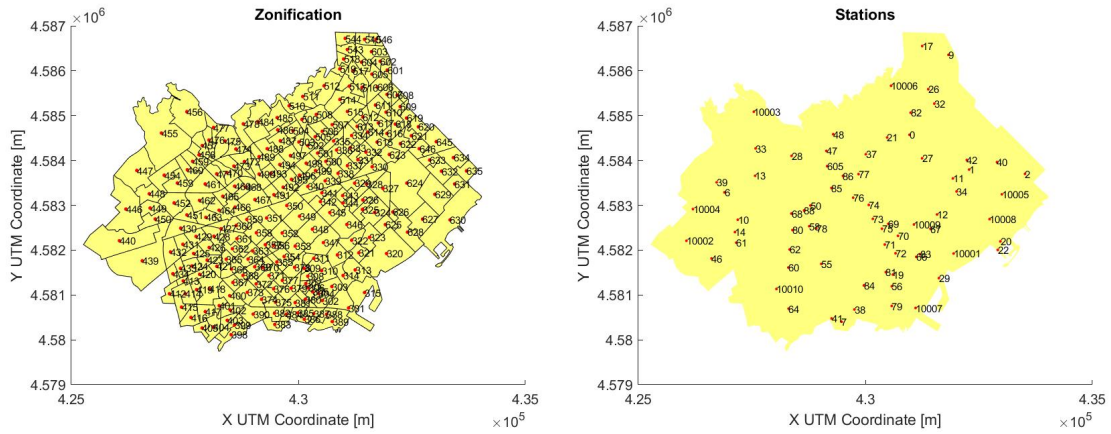
Figure 10. *Map of the service region with the a) Zonification and b) Stations' location.*
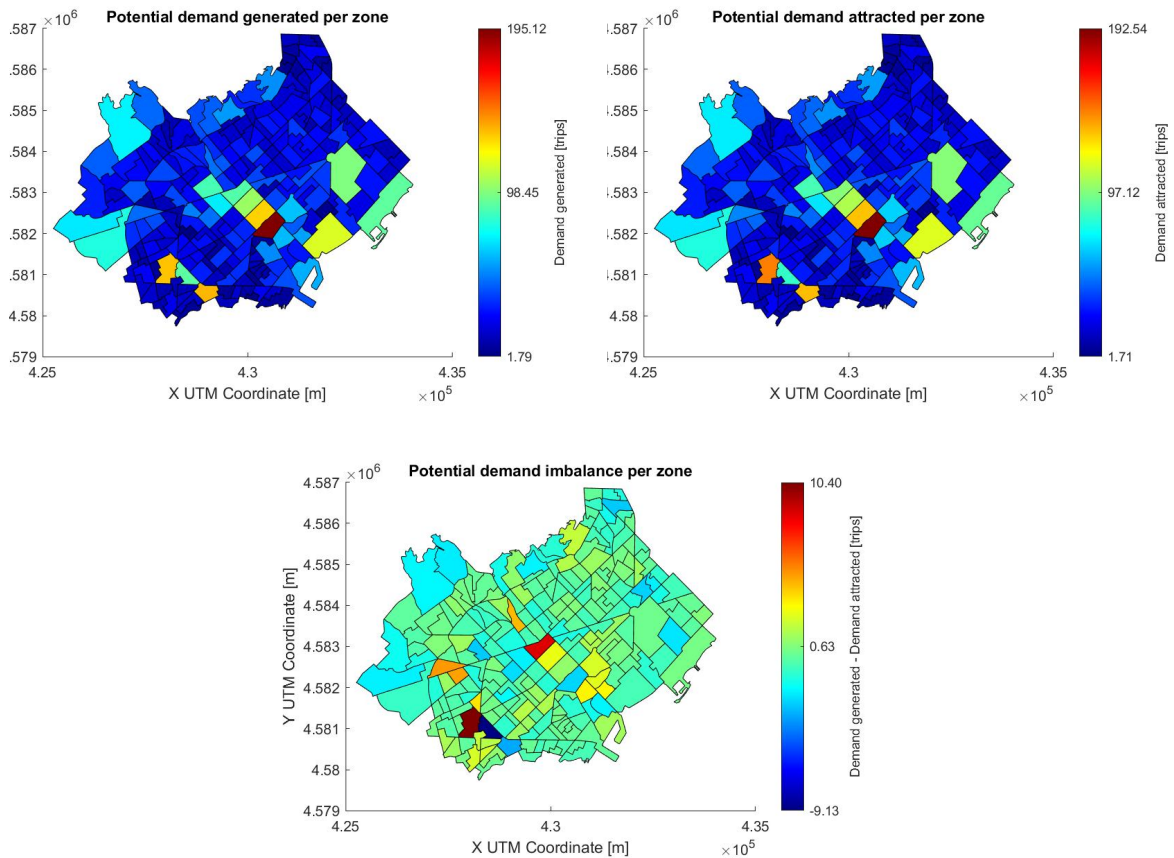
Figure 11. *Map of the service region representing a) the trip generation (i.e. requests); b) the trip attraction (i.e. returns); and c) the demand imbalance (i.e. returns-requests).*

6.2.2.4. The "YYYYMMDD_HHMMSS_Generated_Users.mat" MATLAB file

This file contains a MATLAB array with the defining parameters of all potential trips in the simulation, including the warm-up time. This file can be introduced as an input in following simulations in order to replicate the same potential trip demand.

6.2.2.5. The "YYYYMMDD_HHMMSS_initial_distribution_SB.xlsx" excel file

This file contains the information about the initial distribution of SB vehicles at the beginning of the warm-up. This distribution is defined by the ID of the station, and the initial number of vehicles.

6.2.3. The "YYYYMMDD_HHMMSS_Final_City_Variables_output" folder

This folder contains all the results selected as outputs by the user. Results are structured in subfolders for each category (i.e. Vehicles, BatteryLevel, StationsOccupancy, Unbalance, Demand, Repositioning), which contain the results for the category and for the selected levels of detail.

In general, all results are split in three segments, which are indicated in their file names:

- *"InStation" => Only SB system.* These results are organized by stations and only consider the SB system. These results ignore everything related to the FF system.

- *"InZones" => Only FF system.* These results are organized by free-floating zones and only consider the FF system. These results ignore everything related to the SB system.

- *"InTotal" => Total.* Aggregated results of the SB and FF systems. These results are organized by free-floating zones, and consider what happens in the FF system and also in all the SB stations contained in the zone.

Additionally, there are three possible levels of aggregation of these categories results. The aggregation level is also indicated in the file names as a suffix:

- *"Time" or "dt" => No aggregation.* These are the detailed results, and do not consider any spatial aggregation and time is divided in blocks of 60 minutes each. Therefore, results are presented hourly for each station/zone/total. Note that, if simulation time is not an integer number of hours, the last block will be of less than 60 minutes. These detailed results can be visualized as a contour plot in a map figure for each single time block (e.g. 24 hourly contour plot maps).

- *"AggT" => Time aggregation.* These results are aggregated in time, meaning that they represent the whole simulation time. Still, they do not consider any spatial aggregation, and are presented for each station/zone/total. These time aggregated results can be visualized as a single contour plot in a map figure for the whole simulation time.

- *"AggR" => Spatial aggregation.* These results consider the aggregation of all zones/stations and return the value for the whole service region. Time is not aggregated, and it is still divided in blocks of 60 minutes each. Therefore, results are presented hourly for the whole service region. These space aggregated results can be visualized as time series plots.

Finally, the simulator also provides an aggregated summary of all the results in the simulation. These results consider the whole service region and the whole simulation period (e.g. 24 hours). The output summary file "Table_summary.xlsx" is presented separately in the main "YYYYMMDD_HHMMSS_Final_City_Variables_output" folder. This aggregated summary will be described in detail in Section 6.2.4.

Each subfolder corresponding to a particular category of results, includes different visualization formats. These are described next.

6.2.3.1. Excel files

This output consists on an excel file with all the numerical results for the category. The different tabs of the spreadsheet contain the different results and aggregation levels selected. These are the numerical results from where all figures are generated.

Figure 12. *Example of excel table output.*

6.2.3.2. Map figures

These outputs are .jpg figures generated for almost all possible results that consider a spatial discretization, either if time is discretized hourly (i.e. results with the suffix "Time" or "dt") or aggregated for the total simulation time (i.e. suffix "AggT").

In these figures, stations or zones are color scaled according to the value of the represented result. If the result considers only the SB system, these figures will represent the stations as big dots in the map. However, if the result considers the FF system or both (i.e. total aggregated FF and SB results), the figures will represent the zones color coded. Figure 13 and Figure 14 show examples of these type of map figures.
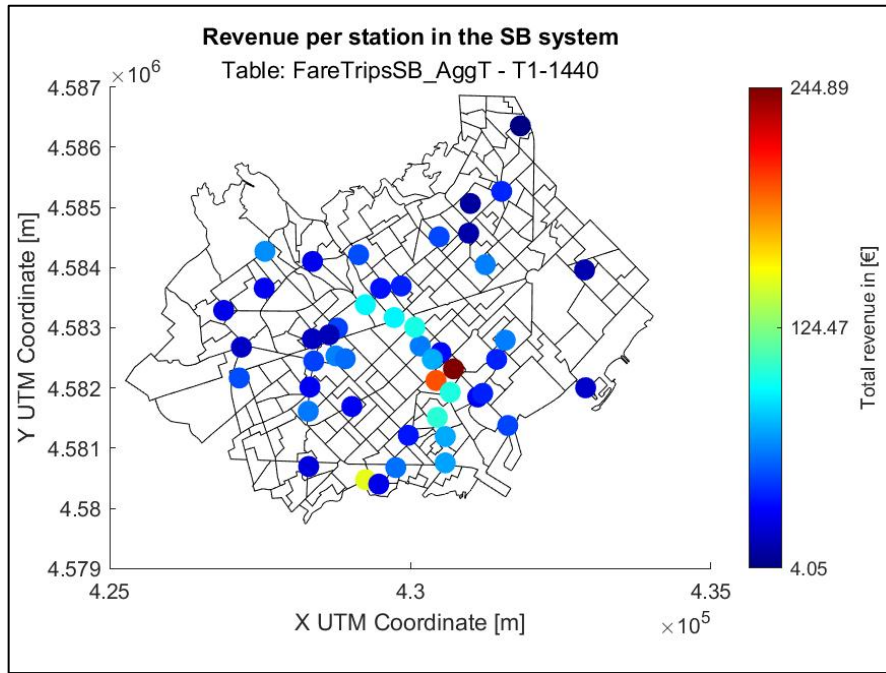
Figure 13. *Example of SB map figure (sorted by stations).*
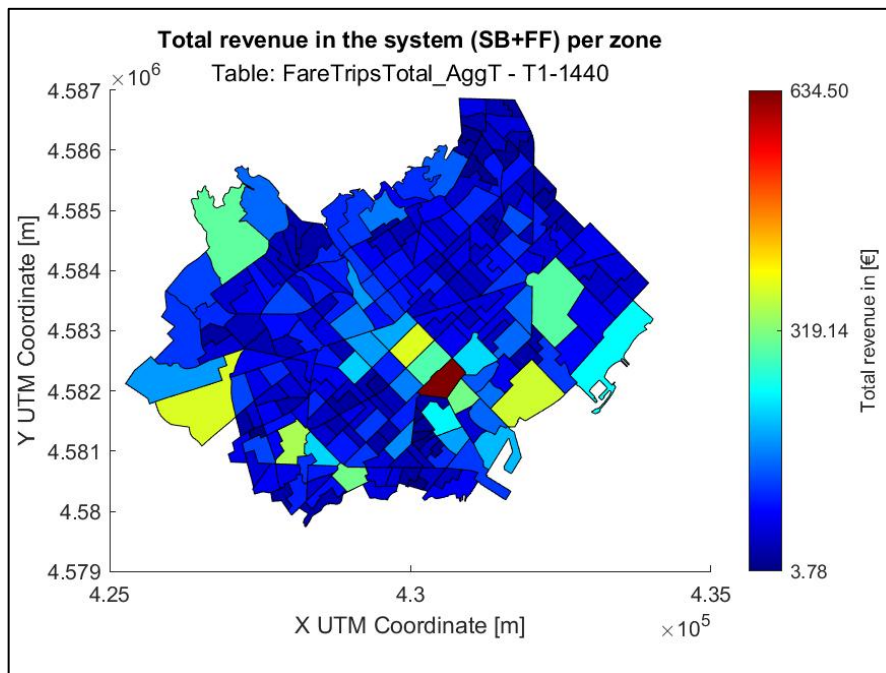
Figure 14. *Example of zone-based map figure (sorted by zones).*

6.2.3.3. Time series

These outputs are generated for almost all possible results that consider a spatial aggregation over the whole service area (suffix "AggR"), but still present time discretization in hours.

In all cases, the horizontal axis depicts the simulation time in hours (e.g. 24 hours), while the vertical axis depicts the considered result.

Figure 15 shows an example of a time series plot for the result "Batterly level".
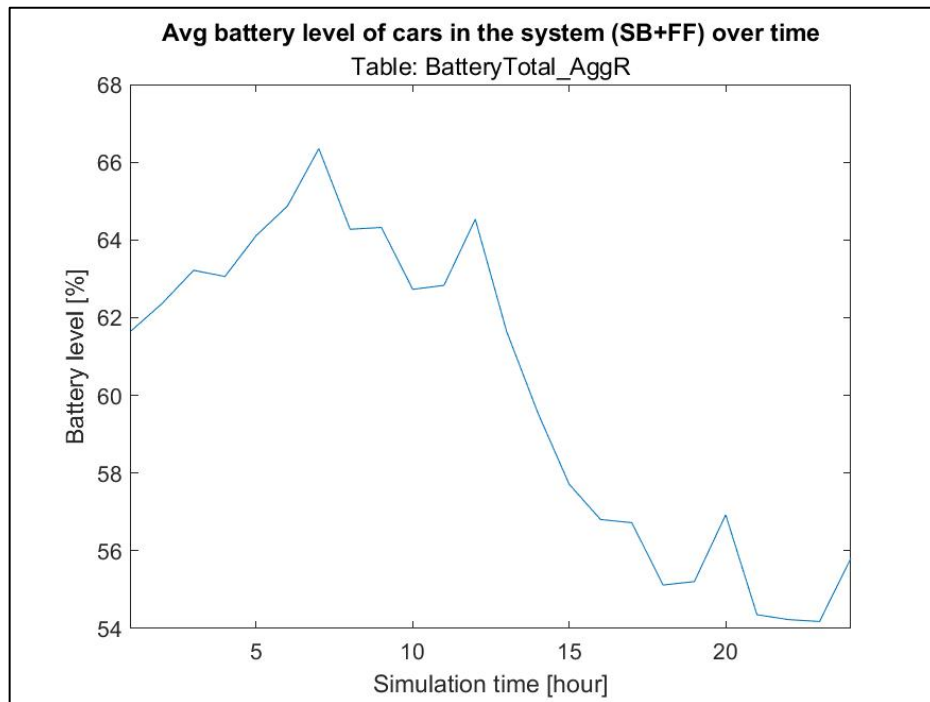
Figure 15. *Example of time-series figure (total battery levels).*

6.2.3.4. Car status pie charts

This visualization output is used to show the average percentage distribution of time that vehicles spend in each status. They can be generated for any 60-minute period (i.e. hourly plots) or for the whole simulation time (e.g. 24 hours). See Section 6.1.4.1 for more details.
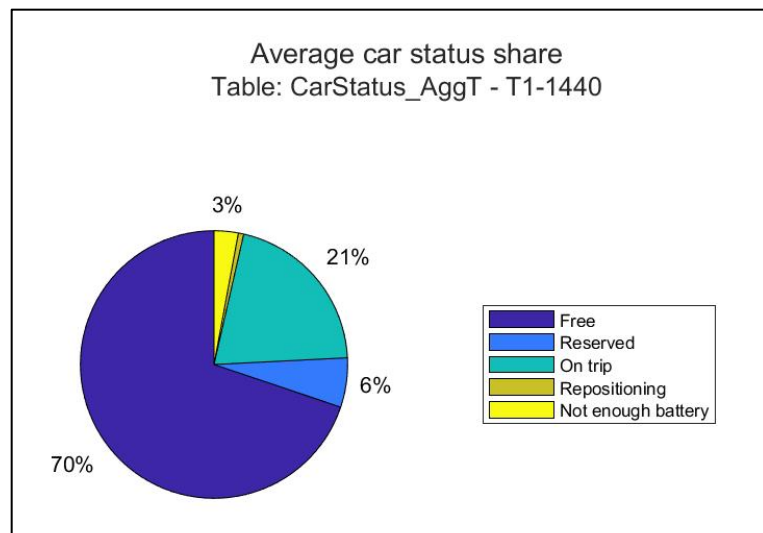
Figure 16. *Example of a pie chart for the vehicles' average time in each status.*

6.2.4. The output summary file "Table_summary.xlsx"

This output file is included in the "YYYYMMDD_HHMMSS_Final_City_Variables_output" folder. It contains the summary of all results from the simulation, including all categories, plus additional results regarding the costs of the system. It skips all the details of the spatial and temporal distribution of results, and presents the main outputs aggregated for the whole

simulation period and for the global service region. This represents the main summary of results that could support decisions at the strategic level.

The aggregated output summary file consists of a .xlsx MS Excel spreadsheet summarizing the main KPI and cost results. These are:

- *The key performance indicators (KPIs)*. These tables include the aggregated version of all results detailed in Section 6.1.4, still divided by categories. All results in this table are aggregated values for the whole service area and total simulation time (e.g. 24 hours). Results are split between the FF and SB systems when relevant.

- *Costs table.* This table includes all the cost components of the car-sharing system, including the agency and user costs, **in units of [€/h].** Table 22 describes the details of all the cost categories included. The unitary costs considered are detailed in Table 21. Auxiliary agency parameters for the result generation (unitary costs).

Table 22. Description of system costs in the output summary file.

Type of cost	Description
Agency costs	
Fleet cost	Amortization costs of the vehicle fleet. It is obtained as the product of the number of electric and ICE vehicles times its respective unitary costs, γ_{ICE} and $\gamma_{electric}$.
SB Stations cost	Agreement costs of using parking stations. It is obtained as the product of the number of stations times its unitary agreement costs prorated per unit time, γ_{STAT}.
SB parking cost	Parking costs of vehicles at stations. It is obtained as the total time spent by car-sharing vehicles parked at stations, times the on-street parking cost per unit time, γ_{S_SB}, and divided by the total simulation time.
SB charger cost	Cost of using the battery charging infrastructure at stations. It is obtained as the total time spent by electrical car-sharing vehicles charging batteries at stations, times the charger cost per unit time, γ_{S_CH}, and divided by the total simulation time.
FF parking cost	Parking costs of vehicles on-street. It is obtained as the total time spent by car-sharing vehicles parked on-street, times the on-street parking cost per unit time, γ_{S_FF}, and divided by the total simulation time.
Total infrastructure cost	Total infrastructure costs incurred by the agency to lay out the system. It is obtained as the sum of the previous fleet, Stations, SB parking, SB charger and FF parking costs.
Administrative and control cost	Total administrative and control cost to run the car-sharing system. It is obtained as total car-sharing vehicle fleet times the unitary administrative and control cost per vehicle and hour, γ_a.
Maintenance cost	Total maintenance cost of the vehicles in the car-sharing system. It is obtained as the total distance traveled by car-sharing vehicles, times

	the maintenance cost per 100 km, γ_m, divided by 100 and by the total simulation time.
Energy cost	Total cost of the energy consumed by the vehicles in the car-sharing system. It is obtained as the total distance traveled by car-sharing vehicles, times the energy cost per 100 km, γ_c, divided by 100 and by the total simulation time. This computation is done separately for ICE and for electric vehicles, because they have different unitary energy costs.
Total operative cost (w/o repo)	Total operative cost for running the system (without considering the repositioning cost, which is computed separately). It is obtained as the sum of administrative and control, maintenance and energy costs of the system.
Repositioning costs	Cost of the repositioning teams in the car-sharing system. It is obtained as the product of the number of repositioning teams working simultaneous in the system, times their cost per unit time, C_t.
Users' costs	
Access cost	Total access and egress cost incurred by users in the car-sharing system. It is obtained as the total cumulative access and egress distance experienced by all users divided by the walking speed, and multiplied by the average users' value of time, β. In order to obtain units of [€/h] this value needs to be divided by the total simulation time.
FF lost demand cost	Total cost of the penalties incurred by FF car-sharing users which do not find any available vehicle. It is obtained as the total number of FF lost trips times its unitary penalty, γ_{lost_FF}. In order to obtain units of [€/h] this value needs to be divided by the total simulation time.
SB no service cost	Total cost of the penalties incurred by SB car-sharing users which do not find any available vehicle. It is obtained as the total number of SB lost trips times its unitary penalty, γ_{lost_SB}. In order to obtain units of [€/h] this value needs to be divided by the total simulation time.
Revenues	
Revenues of SB system (i.e. in stations)	Total revenues of the SB car-sharing system. It is obtained as the product of the trip fare times the number of trips served from stations (SB). In order to obtain units of [€/h] this value needs to be divided by the total simulation time.
Revenues of FF system (i.e. in street)	Total revenues of the FF car-sharing system. It is obtained as the product of the trip fare times the number of trips served from streets (FF). In order to obtain units of [€/h] this value needs to be divided by the total simulation time.
Total costs and revenues	
Total agency cost	Total costs incurred by the agency. It is obtained as the sum of all the agency costs: fleet, operative and repositioning costs.

Total users' cost	Total costs incurred by the users. It is obtained as the sum of all the users' costs: access and no service penalties.
Total system generalized cost	Total costs of the car-sharing system. Sum of the agency and the users' costs.
Total system revenue	Total revenues of the car-sharing system. It is obtained as the sum of the SB and FF revenues, or equivalently as the product of the trip fare times the total number of trips served. In order to obtain units of [€/h] this value needs to be divided by the total simulation time.
Generalized cost per trip	Generalized cost of each trip. It is obtained as the total system generalized costs divided by the total number of trips served.
Revenue neutral fare	It represents the required fare so that the revenues of the system are enough to cover all the agency costs (i.e. the resulting agency profit is zero).
Profit	Agency profit obtained in the operation of the car-sharing system. It is obtained as the difference between the total revenues and the total agency costs.

Note: all cost results are expressed in units of €/h, except the "generalized cost per trip" and the "revenue neutral fare" which are expressed in €/trip.

APPENDIX – Default parameter value estimation process

This appendix describes the assumptions made and the estimation procedure for all the default values in the input parameters to the simulator. For all the parameters, the Barcelona case study is considered. In some instances, various estimation procedures are possible. In such cases, one of the estimations is selected, although the alternatives are also presented.

A1.   Service Area

Service area zonification
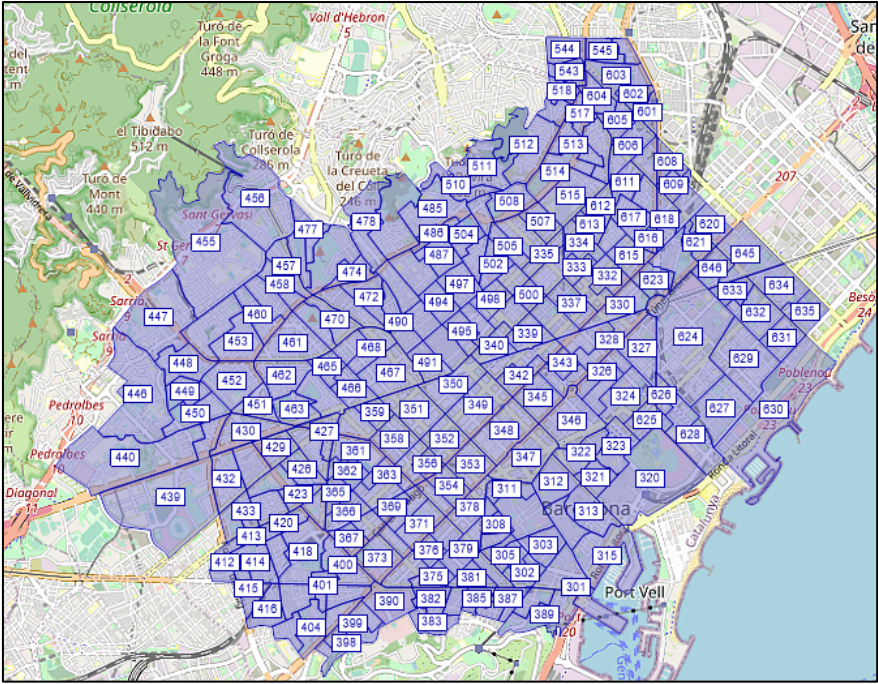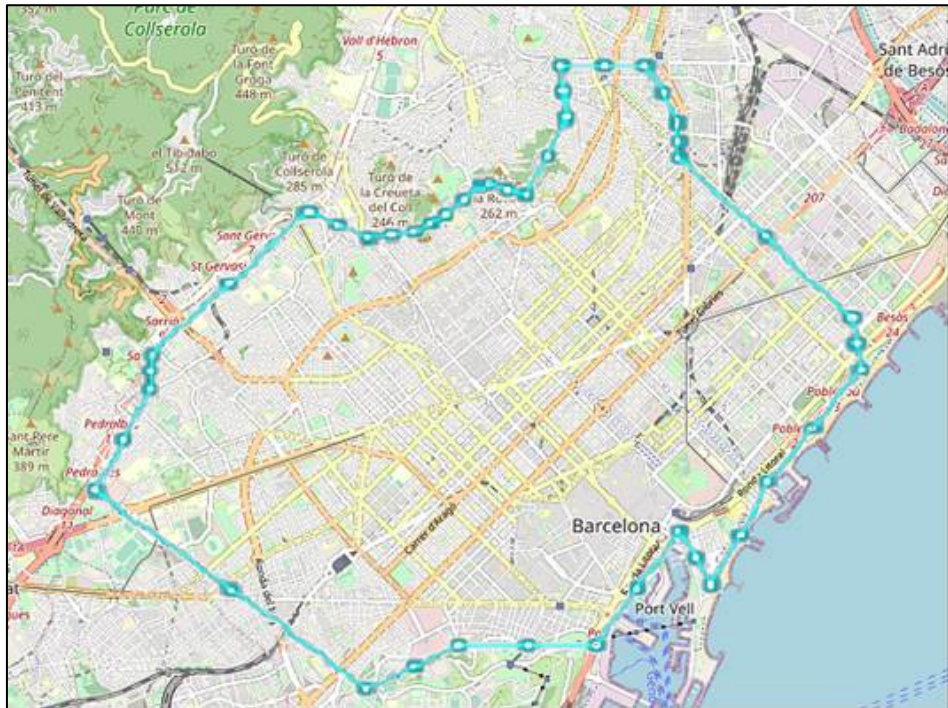
	ShapeFile	"UTM_full_zonification_park_zone_MontjuicEdit.SHP"	Source: [A1]
Def.	Service area zonification for the carsharing system: Central Barcelona.		
Estimation process	[A1] provided a shapefile for the Barcelona intra-rondes region and considering a zonification with 384 subzones. The following changes to that shapefile were applied: • Reduction to a subset of 228 subzones, considering only central Barcelona. • The subzone of Montjuic is reduced to only Plaça Espanya and Fira Barcelona, where the demand is concentrated. • The attribute depicting parking desirability on station (SB_PRK) was added to all subzones and set to the average value of 0.43. • The coordinate system is set to WGS84 UTM projection. The resulting shapefile (the shapefile .shp and its auxiliary files) is stored in the folder "Data/SERVICE_AREA/WGS84_UTM_31N_ZONIFICATION". Figure A1 below depicts the resulting zonification, which in total represents a total area of $R = 35.54 km^2$ **Figure A1**. Service region map.		

Service area perimeter

PerimeterFile	"boundary_UTM_31N.xlsx"	Source: -

Def.	Perimeter of the service area for generating a custom zonification.
Estimation process	Using a web mapping platform (e.g. Google Maps), the coordinates of several points were obtained defining approximately the same area as the previous zonification shapefile. Coordinates were transformed to WGS84 UTM and stored in an excel file as in the following figure.

	A	B	C	D
1	vertex ID	X	Y	Z
2	1	430923.570	4580123.745	0.000
3	2	429371.577	4580192.722	0.000
4	3	428267.938	4579732.873	0.000
5	4	425240.117	4582136.305	0.000

Figure A2. Perimeter of the service region map.

The excel file is saved in the folder "Data/SERVICE_AREA".

Figure A3. Representation of the perimeter of the service region.

Demand matrices in .csv files

Odmat_prefix	FF_hourly_demand_0.5_perc/matrix	Source: [A1], [A4], [A5], [A6], [A7]
Def.	OD matrices input. O.5% market penetration.	
Estimation process	[A1] provides hourly origin-destination (OD) matrices associated to the previously defined zonification shapefile. These matrices include all trips longer than 1.4 kilometers (made either by public or private transportation modes). The source of the mobility data is due to Kineo and was measured during a weekday in 2017. Matrices were adapted in order to only consider the 228 selected subzones and a market penetration of 0.5%. The resulting set of files are saved with the name "matrix_H.csv", where H is a number from 0 to 23 depicting the corresponding hour of the day. The set of matrices were stored in the subfolder "Data\DEMAND_OD\FF_hourly_demand_0.5_perc".	
Note	With respect to market penetration, [A4] found a modal penetration of **0.37% for car2go car-sharing system in Ulm** (Germany), the first free-floating car-sharing system implemented in the world. However, more recent works point out that free-floating car-sharing demand is drawn from a different pool of users than those in Ulm, and therefore, potential demand could be way higher [A5]. So, most of the recent works rely on forecasting rather than observed values. [A6] studied current tendencies and users' behavior and concluded that in **London car-sharing, modal penetration could be stablished as 3.8% on the medium to long term**. Given these references, a modal penetration rate of 0.5% is selected as a conservative default value. This is consistent with the modal penetration rate observed for the car-sharing service in the city of Milan [A7]. In addition to this one, two additional sets of matrices are provided with a lower and higher market penetration bounds. In particular, a 25% and a 139% of the default value. The latter one corresponds to the demand level of motorbike sharing in Barcelona.	

Demand update time

TimeReDemand (TD)	60 minutes	Source: [A1]
Def.	Duration of each matrix .csv file.	
Estimation process	Directly from [A1]. The matrices provided have a duration of 60 minutes each.	

A2.   Stations

Station candidate list

InputStationFile	"stations.xlsx"	Source: [A1]

<table>
<tr>
<td rowspan="1" style="writing-mode:vertical-lr">Def.</td>
<td>List of candidate stations.</td>
</tr>
<tr>
<td style="writing-mode:vertical-lr">Estimation process</td>
<td>

The selection of parking stations was provided by [A1] in a .json file. The input format was changed to an excel spreadsheet.

The file information is the following:

- Column A: Station ID.
- Column B: Name.
- Column C: X coordinate in UTM system.
- Column D: Y coordinate in UTM system.
- Column E: Z coordinate in UTM system. Set to zero by default.
- Column F: Capacity of the station. Set to 9999 by default.
- Column G: Number of chargers. Set to 9999 by default.
- Column H: Address of the station (when available).
- Column I: Station operator company.
- Column J: Priority of implementation, as considered in [A1].

</td>
</tr>
</table>

	A	B	C	D	E	F	G	H	I	J
1	ID	Name	X	Y	Z	Cap	e-chargers	Info 1	Info 2	Info 3
2	50	SABA-Aparcament SABA Travessera de GrÃ cia 43	428790.116	4582996.201	0.000	9999	9999	address	SABA	-1
3	55	SABA-Aparcament BAMSA ValÃ¨ncia - CalÃ bria	429028.594	4581693.708	0.000	9999	9999	address	SABA	-1
4	56	SABA-Aparcament BAMSA Illa Raval	430582.304	4581193.462	0.000	9999	9999	address	SABA	-1
5	57	SABA-Aparcament BAMSA Barceloneta	432196.040	4581240.359	0.000	9999	9999	address	SABA	-1
6	58	SABA-Aparcament BAMSA Londres-Villarroel	428762.953	4582533.697	0.000	9999	9999	address	SABA	-1
7	60	SABA-Aparcament BAMSA Josep Tarradellas II	428303.106	4581612.804	0.000	9999	9999	address	SABA	-1
8	61	SABA-Aparcament BAMSA Joan GÃ¼ell	427156.528	4582166.398	0.000	9999	9999	address	SABA	-1
9	62	SABA-Aparcament BAMSA Tarradellas II	428329.850	4582014.465	0.000	9999	9999	address	SABA	-1
10	64	SABA-Aparcament BAMSA Vilardell	428309.992	4580695.967	0.000	9999	9999	address	SABA	-1
11	66	SABA-Aparcament SABA Catedral	431120.857	4581850.607	0.000	9999	9999	address	SABA	-1
12	67	SABA-Aparcament SABA LluÃs Companys	431435.112	4582476.158	0.000	9999	9999	address	SABA	-1
13	68	SABA-Aparcament SABA Pau Casals	428370.560	4582816.695	0.000	9999	9999	address	SABA	-1
14	69	SABA-Aparcament SABA DiputaciÃ³	430506.216	4582587.933	0.000	9999	9999	address	SABA	-1
15	70	SABA-Aparcament SABA PlaÃ§a Urquinaona	430717.901	4582318.660	0.000	9999	9999	address	SABA	-1
16	71	SABA-Aparcament SABA Rambla Catalunya	430427.214	4582124.925	0.000	9999	9999	address	SABA	-1
17	72	SABA-Aparcament SABA PlaÃ§a de Catalunya	430664.042	4581930.353	0.000	9999	9999	address	SABA	-1
18	73	SABA-Aparcament SABA GrÃ cia III (Pg de GrÃ cia, AragÃ³-Rosse	430164.106	4582697.922	0.000	9999	9999	address	SABA	-1
19	74	SABA-Aparcament SABA Pau Claris	430074.646	4583008.112	0.000	9999	9999	address	SABA	-1
20	75	SABA-Aparcament SABA Gran Via II (Pg de GrÃ cia, Gran Via-Ari	430365.634	4582477.477	0.000	9999	9999	address	SABA	-1

Figure A4. "station.xlsx" file.

A3.   Car parameters

Electric vehicle battery autonomy

BatteryConsume (D_{ev})	$200\ km$	Source: [A41]
Definition	Maximum distance that a fully charged e-car can travel.	
Estimation process	According to SEAT Mii technical specifications, fully charged autonomy is 250 km. The 80% of this autonomy is considered. **NOTE:** Considering that car-sharing trips will be mostly urban trips, probably this value is very optimistic.	

Electric vehicle battery charging time

BateryChargeTime (T_{ev})	$240\ min$	Source: [A41]
Definition	Time required to charge the battery of an e-car from empty battery up to 80%.	
Estimation process	According to SEAT Mii electric specifications: • On AC 7.2kW: 4 hours for 80%. AC charger is considered by default. • On DC 40kW: 1 hour for 80%	

A4.   User parameters

Maximum walking distance

Wmax **(a)**	$400\ m$	Source: [A2], [A17], [A18], [A19], [A20]
Def.	Maximum distance users accept to walk in order to reach a car-sharing station.	

Estimation process	From different literature sources:
	
	 • [A17] on a survey of car2go in Hamburg.
	     o 20% => $a = 300m$
	     o 60% => $a = 500m$
	     o 20% => $a \geq 500m$
	
	 • For station-based bike-sharing systems it is found:
	     o $a = 300m$ [A18], without empirical justification.
	     o $a = 300m$ in Velib (Paris) [A19]
	     o $a = 250 - 300m$ in Bixi (Montreal) [A19]
	     o $a = 350m$ Bicing (Barcelona) [A2]
	
	 NOTE: In station-based bike-sharing systems in operation, the maximum walking distance is obtained from the density of stations. Actually, these empirical data are not the maximum users would be willing to walk, but what they actually walk given the existing density of stations.
	
	 • Most studies use distance thresholds of 400 m for access to bus stops and 800 m for metro or railway stations [A20]. Because people are willing to walk longer distances for accessing transport facilities for longer trips and the average distance for bike trips is very short, the impedance threshold for accessing bike- sharing stations should be lower than that used for car-sharing facilities [A19].
	
	 • [A21] found that the often-used walking range is shorter than 600 m, and very few exceed 1200 m.
	
	 Given these values reported in the literature, and considering that a car-sharing system represents higher added value than a bike-sharing system, **a distance of 400 m is considered** as the maximum distance users are willing to walk to access a vehicle.

Average walking speed

WalkSpeed v_w	3 km/h	Source: [A22]
Definition	Average walking speed in the city of Barcelona for pedestrians.	
Estimation process	According to the [A22], the average walking speed is 1 m/s (i.e. 3,6 km/h). This value is rounded down to account for the additional lost time at intersections and pedestrian crossings. $$v_w = 3km/h$$	

Average circulating speed of cars

CarSpeed (v)		15.3 km/h	Source: [A8]
Definition		Average driving speed for cars in the city of Barcelona. It does consider delays at traffic lights and congestion.	
Raw data	Average measured speed	22.9 km/h	Source: [A8]
	Definition	Average speed in the streets of Barcelona, measured with punctual detectors. It includes neither delays at traffic lights, nor the new regulations in speed limits.	
Raw data	Effective driving time factor	0.66	Source: [-]
	Definition	Fraction of time that cars are driving. It is assumed that 1/3 of the circulating time is lost due to multiple delays (e.g. congestion, intersections, traffic lights)	
Estimation process	The average circulating speed is the result of multiplying the measured speed by the effective driving time factor in order to account for the average delays. This is: $$v_c = 22.9 \frac{km}{h} \cdot 0.66 = 15.267 \ km/h$$ For comparison purpose, note that the average taxi speed in Barcelona in 2014 was 18.6 km/h [A9]. This value includes metropolitan trips, like trips to the airport.		

Average parking time

avgParkTime (τ_p)	6.5 min	Source: [A11]
Def.	Time spent by drivers in order to find an available parking slot and park the vehicle.	
Estimation process	According to a survey in 19 major European cities [A11], the average urban driver spends 6.5 minutes looking for available parking slot near his workplace. $$\tau_p = 6.5 \ min$$ **NOTE:** This value varies largely depending on the parking policies and city district. It could even rise up to 20 minutes (considering users looking for free parking in a congested city as in [A12]). Such scenarios are considered improbable in a car-sharing system, where at the end of the day, stations with reserved parking spots are available. The possibility of accepting free-floating parking reservations, would improve the performance of car-sharing systems [A13], [A14].		

Average demand fraction of parking in station

percParking (SB_PRK)	0.43	Source: [A45]
Def.	Fraction of demand that prefers to park on stations when both, on-street and station options are available.	
Estimation process	Figure A5 shows a table of [A45], which analyzes parking preferences of users travelling with private vehicle. Then, only users travelling to a different city from hometown are considered (column "A un alter municipi"). Users with preference of parking on street are taken from the row "Carrer" (46.6%). Users with preference of parking on stations are taken from the row "Reservat en destinació" (34.8%). The rest of the values are not considered. Therefore: $$SB_{PRK} = \frac{34.8}{34.8 + 46.6} = 0.43$$		

Lloc d'aparcament del cotxe	Municipi de destinació	
	Al municipi de residència	A un altre municipi
Carrer*	37,1%	46,6%
Propietat, lloguer o concessió	40,6%	5,2%
Reservat en destinació (carrer/subterrani)*	11,4%	34,8%
Pàrking de pagament (per hores o minuts)	3,1%	6,2%
Intercanvi (estació de tren o d'autobusos)*	..	1,1%
Altres**	7,3%	6,0%
Total	**100%**	**100%**

* Inclou: pagament i gratuït.
**Inclou: no aparca, només para un moment; altres llocs; i no ho sap, no contesta.

Figure A5. "Lloc d'aparcament de vehicle privat." EMEF 2019. [A45]

Value of penalties

costLostSB, costLostFF $(\gamma_{lost\_SB}, \gamma_{lost\_FF})$	$\gamma_{lost\_SB} = 7.93 \ €/trip$ $\gamma_{lost\_FF} = 2.50 \ €/trip$	Source: [A25], [A26], [A27]
Def.	Average cost perceived by the users when facing service unavailability.	

		Average public transport fare	1.14 €/trip	Source: [A25]
Raw data	**Definition**	Individual public transport fare in Barcelona. T-Casual card provides 10 trips at a cost of 11.35€. It is assumed that car-sharing users will use a T-Casual card in case of needing to travel using public transportation options.		
		Average taxi fare	7.93 €/trip	Source: [A26], [A27]
	Definition	According to [A26] the average taxi trip in Barcelona (not considering metropolitan or airport trips) resulted in a fare of 7.29 € in 2015. This value has been updated according to the increases on regulated fares between the 2015-2020 period [A27]. The resulting value is 7.93 €/trip.		

Estimation process	For station-based service, the taxi fare is considered as the no-service penalty. Station-based users perceive the car-sharing system as reliable. In case of unavailability, users suffer a significant penalty because of lost times and the need to look for an alternative transportation mode. Taxi service is the equivalent option to defray such penalties.$$\gamma_{lost\_SB} = 7.93 \ €/trip$$However, users of the free-floating service do not perceive such high penalty, as they already acknowledge that availability is somehow random and mostly, they use the system in an opportunistic way. They take advantage of the service if available, but they do not rely on it as a regular option. According to the literature, approximately 80% of users will take public transportation as an alternative if car-sharing service is not available [A17], [A28]. The rest, will take their own car, pay for a taxi, or cancel the trip.In order to estimate the no-service penalty in this case, we consider that 80% of the unmet demand will pay a public transport ticket, and 20% will pay the taxi fare. Note that the penalty of using a private car, or not travelling, might be assimilated to the paying for a taxi option.$$\gamma_{lost\_FF} = 2.50 \ €/trip$$

User value of time

costTimeUser β	11.40 €]	Source: [A23], [A24]
Def.	The average value of time perceived by the population when accessing a public transportation system by walking.	
Estimation process	Directly from [A23]: $$\beta = 11.40 \text{ €}$$ This estimation is consistent with the research of the US Department of Transportation [A24] adapted to the Spanish context.	

A5. Repositioning parameters

Average scooter speed of repositioning staff

repoSpeed (v_k)	$8.8\ km/h$	Source: [A16]
Definition	Average speed at which the repositioning staff moves from one vehicle to the next. It is considered that they travel with an electric scooter (e.g. SEAT EXS KickScooter).	
Estimation process	Directly from [A16]: $$v_k = 8.8\ km/h$$	

Average time spent on fixed repositioning operations

taskDuration (δ)	$6\ min$	Source: [-]
Definition	Fixed time required for performing the necessary operations at the start and end of one rebalancing operation (i.e. while the vehicle is stopped). This includes locking/unlocking the vehicle, folding/unfolding the electric scooter, administrative tasks, and a light cleaning of the car. The travelling time (either with the vehicle or with the scooter is not included in this fixed time).	
Estimation process	It is considered an average of 6 minutes including all operations. $$\delta = 6\ min$$ **NOTE**: There exist differences between the required operations for electric and fuel vehicles. A fraction of fuel vehicles will require refueling, which will be performed while repositioning. In contrast electric vehicles require to park in a parking slot equipped with battery charging equipment and plug the car. These differences are neglected due to the lack of reliable measurements and assuming that they represent a small part of the average time spent in fixed operations. Therefore, this time is assumed to be independent of the vehicle type.	

Cost of repositioning staff

costRepo (C_t)	21.54 €/h	Source: [A29], [A38]
Definition	Hourly cost of a repositioning team, including one employee and an the e-scooter acquisition and maintenance.	

Raw data	Average hourly income	14.32	[€/h]	Source: [A38]
	Definition Average income for an hour of work. According to [A38] the gross average hourly income for machine operator workers in Catalonia was 14.32 €/h in 2017.			
	Cost of equipment acquisition	599	[€]	Source: [A29]
	Definition Cost of SEAT EXS KickScooter.			
	Useful life of equipment	1.7	[years]	Source: [-]
	Definition Estimated useful life of an e-scooter. Residual value is considered null.			

Estimation process	Sum of the labor costs and the prorated cost of equipment acquisition. This total cost is increased by an inefficiency factor of 1.5. It is assumed that 1/3 of the working time is lost. $$C_t = 1.5 \cdot \left(Labor\ cost + \frac{Scooter\ acquisition\ cost}{Useful\ life} \right)$$ $$C_t = 1.5 \cdot \left(14.32\ €/h + \frac{599\ €}{1.7 \cdot 365 \cdot 24\ h} \right) = 21.54\ €/h$$ **NOTE:** 1) The prorated cost of equipment acquisition could be neglected compared to labor costs. 2) Technical specifications of SEAT EXS KickScooter state an autonomy of 25km. This value could be restrictive if repositioning operations require travelling a lot of distance during the day. Scooter batteries could be charged while traveling with the car. Also, spare batteries could be considered if necessary.

A6. Agency parameters (outputs.xlsx)

Prorated acquisition cost per electric vehicle

CostCarElectric $(\gamma_{electric})$	0.33 €/car · h			Source: [A29], [A30], [A31]	
Def.	Hourly depreciation cost of each electric vehicle in the fleet.				
Raw data	Price of one new e-vehicle	17.900	[€]	Source: [A31]	
	Def.	Market price of a SEAT Mii electric model. This is a retail price, without considering any subsides.			
	Percentage depreciation of e-vehicle	50.6	[%]	Source: [-]	
	Def.	Percentage of value lost by the vehicle after its useful life. Estimated through average depreciation of vehicles on several second-hand market websites. Only e-cars with mileage between 20.000 and 50.000 km were considered.			
	Useful life of an e-vehicle	5	[years]	Source: [A29]	
	Def.	Useful life of an e-vehicle operated in a car-sharing system. According to VW-SEAT [A29], the useful life of vehicles in a car-sharing system can be 4 - 5 years considering a daily use of 130 km. Actually, daily use in current car-sharing systems is much lower (between 15 and 25 km per day) [A15]. Here, it is assumed a useful life of 5 years, corresponding with the EV battery warranty period.			
	Insurance cost	943.40	[€/car·year]	Source: [A30]	
	Def.	Cost of insurances per car and year. According to [A30], a cost of 1000 CHF/car·year is considered and transformed into €.			
	Ratio fleet size/available e-cars	1.05	[-]	Source: [A30]	
	Definition	Ratio accounting for the additional stock needed to cover vehicle unavailability due to maintenance and repairs. [A30] states that car-sharing vehicles are unavailable 5% of the time due to maintenance. This value includes cleaning, reparations, refueling and relocations, and is based on experience of car-sharing operators in Zürich.			

Estimation process	The fixed hourly cost of the fleet is the result of prorating the depreciation by the total useful life. In addition, we include a factor in order to consider the additional stock of vehicles necessary to replace the fleet when they are unavailable due to maintenance tasks.

$$\gamma = \xi \cdot \left(\frac{Price\ of\ car \cdot Percentage\ depreciation}{Useful\ life\ (t_c)} + Insurance \right)$$

NOTE: In case of considering the residual value of vehicles as a fixed value, apply the following formula instead:

$$\gamma = \xi \cdot \left(\frac{Price\ of\ car - Residual\ value}{Useful\ life\ (t_c)} + Insurance \right)$$

Prorated acquisition cost per electric vehicle

CostCarICE (γ_{ICE})	$0.23 \, €/car \cdot h$	Source: [A30], [A31]

<table>
<tr><td rowspan="2">Definition</td><td colspan="4">Hourly depreciation cost of each fuel vehicle in the fleet.</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td rowspan="12">Raw data</td><td colspan="2">Price of one new fuel vehicle</td><td>9.900</td><td>[€]</td><td>Source: [A31]</td></tr>
<tr><td>Definition</td><td colspan="4">Market price of a SEAT Mii Ecomotive model. This is a retail price, without considering any subsides.</td></tr>
<tr><td colspan="2">Percentage depreciation of a fuel vehicle</td><td>50.6</td><td>[%]</td><td>Source: [-]</td></tr>
<tr><td>Definition</td><td colspan="4">Percentage of value lost by the vehicle after its useful life.

Considered the same as in the electric model</td></tr>
<tr><td colspan="2">Useful life of a fuel vehicle</td><td>5</td><td>[years]</td><td>Source: [-]</td></tr>
<tr><td>Definition</td><td colspan="4">Useful life of a fuel vehicle operated in a car-sharing system.

Considered the same as in the electric model.</td></tr>
<tr><td colspan="2">Insurance cost</td><td>943.40</td><td>[€/car·year]</td><td>Source: [A30]</td></tr>
<tr><td>Definition</td><td colspan="4">Cost of insurances per car and year.

According to [A30], a cost of 1000 CHF/car·year is considered and transformed into €.</td></tr>
<tr><td colspan="2">Ratio fleet size/available e-cars</td><td>1.05</td><td>[-]</td><td>Source: [A30]</td></tr>
<tr><td>Definition</td><td colspan="4">Ratio accounting for the additional stock needed to cover vehicle unavailability due to maintenance and repairs.</td></tr>
</table>

Estimation process	Same as in the electric vehicle option: $$\gamma = 1.05 \cdot \left(\frac{9.900€ \cdot 0.506}{5 \, years} + 943.40€/year \right) = 2042.54€/year = 0.233 \, €/h$$

Parking cost on station

costParkingSB (γ_{S_SB})	$0.25\ €/parking \cdot h$	Source: [A32], [A33]
Definition	Hourly cost of parking on station (rent).	
Estimation process	Equivalent cost of renting a private parking in Barcelona. The network of SABA and B:SM parkings offer subscriptions for 180.00 €/month, which equals 0.25 €/parking·h.	

Parking cost on street

costParkingSB (γ_{S_FF})	$0.00\ €/parking \cdot h$	Source: [A32], [A33]
Definition	Hourly cost of parking on station (rent).	

Raw data	Public parking fare	0.67	[€/h]	Source: [A37]
	Definition	On-street parking fare in the municipality of Barcelona. In Barcelona there is a zone-based parking fare system. The car-sharing service area in central Barcelona, mainly belongs to the parking area called "Area Blava". The "Area Blava" fare is 2.50 €/h which only applies during 45 hours a week (peak hours). The average hourly fare is then: $2.50 \cdot (45/24 \cdot 7) = 0.67\ €/h$			

Estimation process	For the present implementation, it is assumed that public parking fees are subsidized. However, it may exist a maximum number of car-sharing vehicles parked on-street. $$\gamma_{S\_FF} = 0.00\ €/h$$ **NOTE:** Some studies suggest that paying for the public parking fee (or a fraction) for on-street parking reservations could improve the performance of car-sharing. In addition, this cost could be compensated by extra revenue [A14]. This could be operative scenario to explore.

Depreciation cost of charging installation

costParkingSB (γ_{CH})		$\gamma_{CH} = 0.05 \ €/charger \cdot h$	Source: [A25], [A26], [A27]
Def.	Hourly cost of using a charging point.		
Raw data	Installation cost of charging point	-	Source: [A25]
	Definition — This cost depends on the typology of the charging point. Slow and medium speed is usually preferred for a domestic use (i.e. regular charger). Prices for them are obtained from [A34]. • 7.4 kW AC installation: 862.80 € • 22 kW AC installation: 922.80 € An additional 20% is applied to consider installation overrun. Quick charging points are currently used for public charging stations. According to [A35], the average investment cost for these installations (including maintenance over all its useful life) is: • 62.5 kW DC installation: 95 000.00 € **NOTE:** [A35] uses data obtained before 2012. According to [A36], these data was still valid in 2018.		
	Useful life of charging point	-	Source: [A26], [A27]
	Definition — Average time that a battery charging point remains operative. For domestic use (regular charger) the 2 year Wallbox warranty period is considered. For quick charging point, [A35] considers an average useful life of 12.5 years.		
Estimation process	This estimation considers the installation of the battery charging point prorated over its useful life. $$\gamma_{CH} = \begin{cases} 0.05 \ €/h & regular\ battery\ charger \\ 0.93 €/h & fast\ battery\ charger \end{cases}$$		

Administrative costs

costOperative (γ_a)	$\gamma_a = 0.24 \; €/car \cdot h$	Source: [A30]
Definition	Overhead costs of the car-sharing system prorated per vehicle. It includes administrative costs, management costs, human resources management, advertising, etc.	
Estimation process	[A30] estimates an overhead cost of 10.24 USD per vehicle and day. Data comes from US transportation agencies. This value is mostly a labor cost, so it must be corrected by considering the salary level in Spain with respect to the US. The average yearly income for administrative jobs used as the conversion factor (obtained from [A38] and [A39]). Then: $$\gamma_a = 10.24 \; \text{USD/veh} \cdot \text{day} \cdot \frac{24 \, h}{1 \, day} \frac{\text{Labor cost US}}{\text{Labor cost Spain}}$$ $$\gamma_a = 10.24 \; \frac{\text{USD}}{\text{veh} \cdot \text{day}} \cdot 24 \, h \cdot \frac{21937.18 \frac{€}{\text{year}}}{38990 \frac{\text{USD}}{\text{year}}} = 0.24 \; €/veh \cdot h$$	

Maintenance costs

costMaintCar (γ_m)	$\gamma_m = 4.39 \, €/100km$	Source: [A30]
Definition	Cost of car maintenance prorated per 100 kilometers.	

	Raw data			
	Maintenance labor cost per 100 km		1.70 €/100km	Source: [A30], [A38], [A40]
	Definition	Labor cost of vehicle maintenance. [A30] considers a labor cost of 6.40 CHF/100km for the maintenance of car-sharing vehicles, based on the experience in Zürich (Switzerland). This value is a labor cost, so it must be corrected considering the salary level in Spain with respect to Zwitzerland, and using the relative median yearly salary as the conversion factor. The median gross yearly salary in Spain is 21937.18 € [A38], and in Switzerland this is 82392.00 CHF [A40]. Then, the corrected maintenance labor cost is: $$6.40 \, CHF/100km \cdot \frac{21937.18 \, €/year}{82392.00 \, CHF/year} = 1.70 \, €/100km$$		
	Tire cost per 100 km		1.89 €/100km	Source: [A30]
	Definition	This represents the prorated cost of tire replacement. [A30] considers a cost of 2 CHF/100km. This cost is converted to € to obtain the 1.89 €/100km.		
	Average cleaning cost per 100 km		0.80 €/100km	Source: [A30], [A38], [A40]
	Definition	Cost of cleaning the vehicles. According to [A30], cleaning represents a cost of 3 CHF/100km. This is mainly a labor cost, so the median salary conversion factor is applied. $$3 \, CHF/100km \cdot \frac{21937.18 \, €/year}{82392.00 \, CHF/year} = 0.80 \, €/100km$$		

Estimation process	The maintenance cost per trip is computed as the sum of previous costs per 100 km distance: $$\gamma_m = 1.70 \, € + 1.89 \, € + 0.80 \, € = 4.39 \, €/100km$$

Energy consumption cost (electric)

costFuelElectric ($\gamma_{c\_electric}$)	$\gamma_{c\_electric} = 3.58 \text{ €}/100km$	Source: [A41], [A42]
Definition	Energy consumption per 100 km of an e-car in the car-sharing system.	

	Vehicle consumption	14.9 kWh/100km	Source: [A41]
	Definition	Energy consumption from the technical specification of the SEAT Mii electric. **NOTE:** Considering that car-sharing trips will be mostly urban trips, probably this value is very optimistic.	
	Electricity price	0.2403	[€/kWh]	Source: [A42]
	Definition	Price of electricity for domestic consumption in Spain in 2019. It includes fixed costs and taxes.	

Estimation process	The battery charging cost is obtained as the energy consumption per trip multiplied by the average electricity price in Spain. $$\gamma_{c\_electric} = 14.9 \; kWh/km \cdot 0.2403 \text{ €}/kWh = 3.58 \text{ €}/100km$$

Energy consumption cost (ICE)

costFuelICE (γ_{c_ICE})	$\gamma_{c\_ICE} = 6.47 \,€/100km$	Source: [A43], [A44]
Def.	Fuel consumption per 100 km of a combustion engine car-sharing vehicle.	

	Vehicle consumption	5 liters/100km		Source: [A43]
	Definition	Gasoline consumption obtained from the technical specifications of SEAT Mii Ecomotive. Urban consumption is considered.		
	Gasoline price	1.293	[€/liter]	Source: [A44]
	Definition	Average price of gasoline in Spain in 2019. It includes fixed costs and taxes.		

Estimation process	The fuel cost is obtained as the fuel consumption per trip multiplied by the average gasoline price in Spain. $$\gamma_{c\_ICE} = 5 \; liter/100km \cdot 1.293 \,€/liter = 6.47 \,€/100km$$

Service fare

avgFare (F)	$0.27 \,€/min$	Source: [A10]
Def.	Average fare users pay to use the car-sharing system.	

Estimation process	According to [A10], the <u>minimum</u> fares for using the one-way car-sharing systems are: • Vienna => 0.26 €/min • Copenhagen => 0.27 €/min • Paris => 0.24 €/min • Germany (Berlin, Hamburg, Munich, Frankfurt) => 0.19 €/min • Budapest => 0.29 €/min • Italy (Rome, Milan and Turin) => 0.19 €/min • Amsterdam => 0.26 €/min • Madrid => 0.19 €/min These fares are increased depending on the time of the day and the vehicle availability at a given location, in order to promote user self-rebalancing movements. For instance, the average fare in Madrid is estimated to be 0.27 €/min, and this is considered as the most representative for the Barcelona case study. Then: $$F = 0.27 \,€/min$$

A7.   Sources

A1. InLab, UPC. (2019). "*Parking Stations Layout for a Car Sharing System Project*". Final Report of the Phase 0.1 of project entitled Management Model of an Electric Car-Sharing System. CARNET Barcelona - Cooperative Automotive Research Network.

A2. Soriguera, F. and E. Jiménez. (2020). A continuous approximation model for the optimal design of public bike-sharing systems. *Sustainable Cities and Society*, 52, 101826.

A3. Soriguera, F. and E. Jiménez. (2017). *Management Model of Electric Vehicle Sharing*. Final Report. Cátedra SEAT-UPC.

A4. Firnkorn, J., and Müller, M. (2011). What will be the environmental effects of new free-floating car-sharing systems? The case of car2go in Ulm. *Ecological economics*, *70*(8), 1519-1528.

A5. Kopp, J., Gerike, R., and Axhausen, K. W. (2015). Do sharing people behave differently? An empirical evaluation of the distinctive mobility patterns of free-floating car-sharing members. *Transportation*, *42*(3), 449-469.

A6. Le Vine, S., Lee-Gosselin, M., Sivakumar, A., and Polak, J. (2014). A new approach to predict the market and impacts of round-trip and point-to-point carsharing systems: case study of London. *Transportation Research Part D: Transport and Environment*, *32*, 218-229.

A7. Bruglieri, M., Colorni, A. and Luè, A. (2014). The vehicle relocation problem for the one-way electric vehicle sharing: an application to the Milan case. *Procedia - Social and Behavioral Sciences*, 111, 18-27.

A8. Ajuntament de Barcelona. (2018). *Dades básiques de mobilitat Barcelona. Informe 2017.* Available online at http://hdl.handle.net/11703/111727 (Accessed March 2020).

A9. CENIT. (2014). *Observatori del Taxi. Document annual 2014*. Institut Metropolità del Taxi.

A10.   Car2Go Madrid. (2020). Website https://www.car2go.com/ES/en/madrid/ (Accessed March 2020).

A11.   Conduent (2016). *Keeping our cities moving: The European Urban Transportation Survey*. Available online at: https://downloads.conduent.com/content/usa/en/ebook/european-urban-transportation-survey.pdf

A12.   Mangiaracina, R., Tumino, A., Miragliotta, G., Salvadori, G., and Perego, A. (2017). Smart parking management in a smart city: Costs and benefits. In *IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*. 27-32.

A13.   Kaspi, M., Raviv, T., and Tzur, M. (2014). Parking reservation policies in one-way vehicle sharing systems. *Transportation Research Part B: Methodological*, *62*, 35-50.

A14.   Balac, M., Ciari, F., and Axhausen, K. W. (2017). Modeling the impact of parking price policy on free-floating carsharing: Case study for Zurich, Switzerland. *Transportation Research Part C: Emerging Technologies*, *77*, 207-225.

A15.   Sprei, F., Habibi, S., Englund, C., Pettersson, S., Voronov, A., and Wedlin, J. (2019). Free-floating car-sharing electrification and mode displacement: Travel time and usage patterns from 12 cities in Europe and the United States. *Transportation Research Part D: Transport and Environment*, *71*, 127-140.

A16.   Liu, M., Seeder, S., and Li, H. (2019). Analysis of e-scooter trips and their temporal usage Patterns. *Institute of Transportation Engineers. ITE Journal, 89*(6), 44-49.

A17. Herrmann, S., Schulte, F., and Voß, S. (2014). Increasing acceptance of free-floating car sharing systems using smart relocation strategies: a survey based study of car2go Hamburg. In *International conference on computational logistics*, 151-162, Springer, Cham.

A18. Lin, J.R. and Hui Yang, T.A. (2011). Strategic design of public bicycle sharing systems with service level constraints. *Transportation Research Part E,* 47(2), 284-294.

A19. García-Palomares, J.C., Gutiérrez, J. and Latorre, M. (2012). Optimizing the location of stations in bike-sharing programs: a GIS approach. *Applied Geography*, 35, 235–246.

A20. Transportation Research Board. (2013). *Transit Capacity and Quality of Service Manual*. Report 165 - TRB's Transit Cooperative Highway Research Program (TCRP). Third Edition.

A21. Millward, H., Spinney, J. and Scott, D. (2013). Active-transport walking behavior: Destinations, durations, distances. *Journal of Transport Geography*, 28, 101–110.

A22. Generalitat de Catalunya. (2017). *Web de Mobilitat*. Available online at: http://mobilitat.gencat.cat/es/serveis/mitjans_de_transport/a_peu/ (Accessed March 2020).

A23. Autoritat del Transport Metropolità. (2017). Personal communication.

A24. U. S. Department of Transportation. (2011). *The value of travel time savings: Departmental guidance for conducting economic evaluations. Revision 2*. Office of the Secretary of Transportation. Available online at: https://www.transportation.gov/sites/dot.dev/files/docs/vot_guidance_092811c.pdf (Accessed January 2017).

A25. Transport Metropolità de Barcelona. (2020). Fare information website. Available online: https://www.tmb.cat/en/barcelona-fares-metro-bus/single-and-integrated/t-casual (Accessed March 2020).

A26. Autoridad Catalana de la Competencia. (2018). *Estudio sobre el sector del transporte de viajeros en vehículos de hasta nueve plazas: el taxi y los vehículos de alquiler con conductor*. Available online at: http://acco.gencat.cat/web/.content/80_acco/documents/arxius/actuacions/20180628_e studi_taxi_veh_lloguer_esp.pdf

A27. Institut Metropolità del Taxi (2020). Website. http://taxi.amb.cat/s/imet.html (Accessed March 2020).

A28. Ampudia-Renuncio, M., Guirao, B., and Molina-Sanchez, R. (2018). The impact of free-floating carsharing on sustainable cities: analysis of first experiences in Madrid with the university campus. *Sustainable Cities and Society*, *43*, 462-475.

A29. Volkswagen-SEAT. (2020). Personal communication.

A30. Bösch, P. M., Becker, F., Becker, H., and Axhausen, K. W. (2018). Cost-based analysis of autonomous mobility services. *Transport Policy*, *64*, 76-91.

A31. SEAT (2020). Website www.seat.es (Accessed March 2020).

A32. B:SM (2020) Website www.bsmsa.cat (Accessed March 2020).

A33. SABA (2020) Website. www.saba.es (Accessed March 2020).

A34. Wallbox (2020) Website www.wallbox.com (Accessed March 2020).

A35. Schroeder, A., and Traber, T. (2012). The economics of fast charging infrastructure for electric vehicles. *Energy Policy*, *43*, 136-144.

A36.   Zhang, Q., Li, H., Zhu, L., Campana, P. E., Lu, H., Wallin, F., and Sun, Q. (2018). Factors influencing the economics of public charging infrastructures for EV–A review. *Renewable and Sustainable Energy Reviews*, *94*, 500-509.

A37.   Ajuntament de Barcelona. (2020). Area Blava Website https://www.areaverda.cat/en/blue (Accessed March 2020).

A38.   Idescat. (2017). *Gross annual wage and earnings per hour. By sex and type of occupation*. Available online at: https://www.idescat.cat/indicadors/?id=anuals&n=10403&col=3&lang=en (Accessed March 2020).

A39.   U.S. Bureau of Labor Statistics. (2020). *May 2018 national occupational employment and wage estimates*. Available online at: https://www.bls.gov/oes/2018/may/oes_nat.htm#43-0000 (Accessed March 2020).

A40.   Office for Economy and Labour. Kanton Zürich. (2020). *Lohnbuch Schweiz 2019*. Wages Book 2019.

A41.   SEAT. (2020). *SEAT Mii electric technical specifications document*. Available online at: https://www.seat.es/content/dam/countries/es/seat-website/compra-tu-seat/descarga-catalogos/cars-models-brochure-KF1-NA-October-2019-electric.pdf

A42.   Eurostat. (2020). *Electricity prices by type of user*. Available online at: http://ec.europa.eu/eurostat/tgm/refreshTableAction.do?tab=table&plugin=1&pcode=ten00117&language=en (Accessed March 2020)

A43.   MotorEU. (2020) Website. *Technical specification of SEAT Mii 1.0 Ecomotive*. Available online at: https://motoreu.com/es/seat-mii-1-0-ecomotive-consumo-ficha-tecnica-15157

A44.   European Comission – Energy Policy. (2020). *Consumer prices of petroleum products inclusive of duties and taxes*. Available online at: https://ec.europa.eu/energy/observatory/reports/latest_prices_with_taxes.pdf

A45.   Ajuntament de Barcelona. (2021). *Enquesta de Mobilitat en dia feiner. EMEF 2019*. Available online at: https://bcnroc.ajuntament.barcelona.cat/jspui/bitstream/11703/119315/4/EMEF_2019_Informe_AMB.pdf