Bike-sharing repositioning experiment – Read-me file (v.1.0)

Note: All the files cited in this document and required to run the experiments in the project, including the source files of the Vsharing Simulator, and the corresponding manuals can be found here: https://github.com/EnriqueJM/Simulator_BSHARING

The purpose of this document is to provide a guide for replicating the simulation experiments presented in [Jiménez-Meroño & Soriguera, 2024a], aiming to compare different repositioning strategies in bike sharing systems (which could be extrapolated to other types of vehicles; e.g. car-sharing, scooter-sharing). The interested researcher should complement the reading of this document with a detailed understanding of the general-purpose user manual of the vehicle sharing simulator: i.e. Vsharing Simulator User Manual (ver. 2.0.0); [Jiménez-Meroño, E. & Soriguera, F. (2020)]. A summary, description, and potential applications of this simulator can be found in [Jiménez-Meroño & Soriguera, 2024b].

The folder and file structure of the Vsharing Simulator is shown in Figure 1.

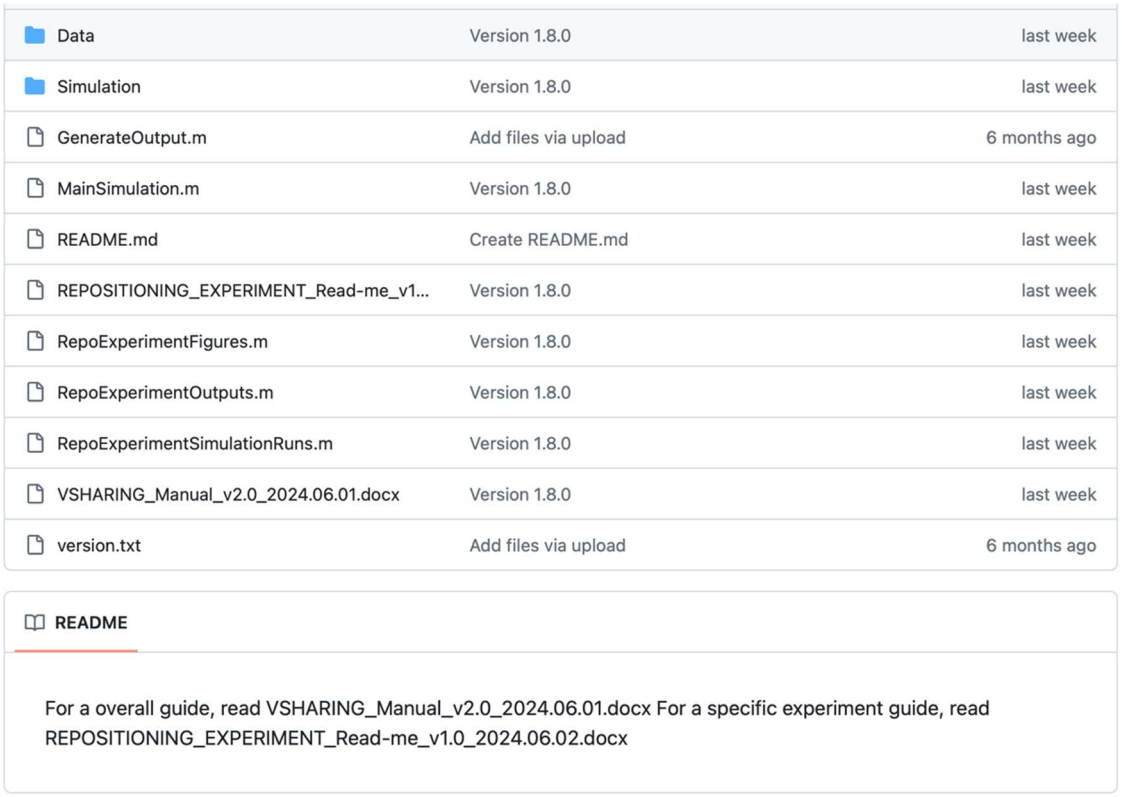

Figure 1. *Folder and file structure of the Vsharing Simulator project on GitHub.*

In order to run the vehicle sharing repositioning experiments, in addition to the core files of the simulator (see Section 1.2. in the Vsharing Simulator User Manual 2.0.0.), the following support files must be included in the Vsharing Simulator folder (see Figure 1):

- **"RepoExperimentSimulationRuns.m" file:** This file is a MATLAB script that runs a set of experiments testing all the proposed repositioning strategies under the same simulation conditions (service area, layout, potential demand trips…).

- **"RepoExperimentOutputs.m" file:** This file is a MATLAB script that, given a raw simulation results file, returns two Excel files with results equivalent to those presented in [Jiménez-Meroño & Soriguera, 2024a].
- **"RepoExperimentFigures.m" file:** This file is a MATLAB script that, given the Excel file with the simulation results processed and aggregated by station, returns the figures equivalent to those presented in [Jiménez-Meroño & Soriguera, 2024a]. The input Excel file is obtained from the "RepoExperimentOutputs.m" script (i.e. "TableFullEmpty.xlsx"; as described in the next sections).
- **"REPOSITIONING_EXPERIMENT_Read-me" file:** This is the present ReadMe file with the guidelines to run the experiments and process their results.

Part 1 – Simulation

1) Preparation of the inputs

Simulation time is set to 6 days, divided into a cycle of 3 days for warm-up and another cycle of 3 days for recording results. From our experience, the simulation duration cycle of 3 days, is an adequate selection. On the one hand, longer cycles would result in excessively large register files that cannot be stored to process the results later on, and also implies longer computational times, in part due to an excessive warm-up duration. Note that the Vsharing Simulator sets the minimum warm-up simulation duration to at least one cycle. On the other hand, shorter cycles might not be representative enough to obtain robust aggregate results later, and would need more experiment repetitions. In any case, other cycle durations could be considered if properly defined (check Section 3.2. of the Vsharing User Manual 2.0.0 for more information about the configuration of the simulation cycle duration.)

In order to set-up this 3+3-days simulation time, the following conditions must be fulfilled:

- If the O/D demand input comes from known matrices, it must be ensured that the matrices cover the whole simulation cycle time (in this case, 3 days).
- The input Excel file "input.xlsx" must be checked to introduce properly the simulation parameters:
    - Set: "TotalTime" = 4320 [min]. In order to set the 3-day duration cycle.
    - Set: "WarmUpCycles = 1. In order to set warm-up to a single cycle (defined as 3 days).
    - Set the rest of parameters as considered for the experiment. See Section 2.1 of the Vsharing User Manual 2.0.0 for more info about the simulation inputs file.

In order to replicate the same experiments, all inputs in "inputs.xlsx" file must remain constant for all the experiments, except for the random seed ("rndSeed"), which must be changed on each run of the simulation script to generate different trip arrays.

2) Run the simulation script *"RepoExperimentSimulationRuns.m"*

The script "*RepoExperimentSimulationRuns.m*" has been coded in order to ease the experiment repetition. The script will run four simulation experiments in a single execution:

- First, it runs the baseline simulation without repositioning. Potential trips are generated randomly, and a register of them is stored in an array (see Section 6.2.2.4. of the

Vsharing User Manual v.2.0.0). All data from the simulation is stored normally (see Sections 6.2.1. and 6.2.2. in the Vsharing User Manual v.2.0.0).

- After this, the script runs three additional simulations. Each one considers a different repositioning strategy of those considered in the analysis (i.e. *i)* Real-time reactive pairwise task assignment optimization; *ii)* Preemptive routing optimization, *iii)* Mixed). For all three simulations the potential trips are not generated randomly now, but read from the trip register of the previous baseline simulation. In this way, the script results in a set of results of four simulations with the same conditions and potential trips, but changing the repositioning strategy.

Note that for each set of experiment repetitions one important parameter is the error in the demand forecast. The "RepoExperimentSimulationRuns.m" script allows choosing between two demand forecast methods (i.e. "average" and "perfect" and introduce ad-hoc disruptions in order to generate different errors in the forecast (see Figure 2).

```
86  %-------------------------------------------------------------------------
87  %                          DEMAND FORECAST PARAMETERS
88  %-------------------------------------------------------------------------
89  % Demand forecast method
90
91 -  param.forecast_method = 'average';          % 'average' || 'perfect'
92 -  param.forecast_disruption = 0;             % 0~1
93
```

Figure 2. *Input of demand forecast options in "RepoExperimentSimulationRuns.m".*

Recall that the "average" forecast method assumes that the number of requests and returns for every time period simply follow the average daily demand rate for that zone of the service area. In turn, the "perfect" forecast method reads the number of requests and returns for every time period from a trip log (see section 6.2.2.4 of the Vsharing User Manual 2.0.0.).

After each run, the simulation returns the resulting average forecast error in requests and returns. Note that, since the resulting error depends on the actual performance of the simulator (users change their behavior and the stations used as the origin and destination of the trip depending on their availability), this cannot be estimated beforehand. So, a tuning process of trial & error would be necessary on the demand forecast parameters in order to obtain forecast errors in any given desired range. This may require running the simulation script several times.

Finally, recall that one of the first steps of the Vsharing simulator is to compute the optimal vehicle distribution over the stations, given the particular inputs defining the overall simulation, and in particular, the demand forecast. By default, the simulation will start the warm-up cycle considering this initial optimal distribution of vehicles at stations. This optimal distribution of vehicles for each simulation is stored by default in:

- Excel file "YYYYMMDD_HHmmss_initial_distribution_SB.xlsx"
- Folder "YYYYMMDD_HHmmss_Final_City_Variables_input"

As an example, the optimal vehicle distributions resulting from the experiments presented in [Jiménez-Meroño & Soriguera, 2024a] for the different error levels are available in the "Data/INITIAL DISTRIBUTION (repositioning experiments)" folder of the project. Also note that the experienced researcher may opt to overwrite the optimal distribution of vehicles and start

the warm-up of the simulation with a different ad-hoc distribution (see Section 2.6 of the Vsharing User Manual v.2.0.0).

3) Repetition

In order to get larger set of results and improve representativity, the process can be repeated by:

- Change the random seed ("rndSeed") in the input file to generate a different set of potential trips.
- Run the *"RepoExperimentSimulationRuns.m"* script again.

The process can be repeated until achieving the desired number of total days for each repositioning strategy.

In case new experiments with a different demand forecast error are desired, then tune the demand forecast error by:

- Change the demand forecast disruption parameter ("param.forecast_disruption") in the demand forecast parameters of the script (see Figure 2) to generate a different demand forecast error.
    - Note: The demand forecast method can also be changed from "average" to "perfect". This would require to change the inputs too, as the "perfect" forecast method requires a trip log, similar to the array of potential trips obtained as an output of the simulator (see section 6.2.2.4 of the Vsharing User Manual 2.0.0.). A log is provided to use the "perfect" forecast method (located in "Data/DEMAND_OD/20240210_023859_Generated_Users_FLAT.mat").
- Run the simulation script to obtain the resulting demand forecast error for the previous input.
- Repeat the two steps above as many times as needed until the forecast error is in the desired range.

Part 2 – Postprocessing of results

In this second part, the data obtained from the simulation runs are processed in order to obtain the desired results. Here, the postprocessing of the simulation outputs is limited to that necessary to generate the final results shown in the paper by [Jiménez-Meroño & Soriguera, 2024a]. However, the interested user could obtain other types of results through the general postprocessing script "GenerateOutput.m" as explained in Section 6 of the Vsharing User Manual 2.0.0.

1) Classification of results

Simulation results files and folders must be classified carefully. It advisable to rename the simulation result file and folder according to the repositioning strategy considered, the demand forecast error, and the random seed, which can be obtained from the input summary file. It is also advisable to group several results files and folders into a main folder that indicates the repositioning strategy and the demand forecast error.

2) Check the output generation variables “output.xlsx” excel file

As a previous step to running the scripts, the input parameters for the postprocessing must be checked in the “output.xlsx” excel file. Note that, since the specific “RepoExperimentOutput.m” script will be run to generate specific results according to those presented in [Jiménez-Meroño & Soriguera, 2024a], many options will in the “output.xlsx” excel file will be limited.

- *SIMULATION variables* => Is it possible to turn on/off the additional information shown in the MATLAB command window during the postprocess. The rest of variable fields are not used in the “RepoExperimentOutput.m” script. In particular, the simulation data file name (“ResFile”) is not introduced here but directly in the script, as detailed in the next section (see Figure 3).
- *DETAL LEVEL and CATEGORY variables* => These variables are not used in the “RepoExperimentOutput.m” script, and changes will not have any effect. The script already implements which categories and detail level returns.
- *USER COST and AGENCY COST variables* => These variables must be filled in accordance to the considered simulation inputs in “inputs.xlsx”.

For more information about the “output.xlsx” file, check Section 6.1. in the Vsharing User Manual 2.0.0.

3) Run the postprocessing script “RepoExperimentOutput.m”

The script “RepoExperimentOutput.m” has been coded to ease the task of generating the customized results according to those presented in [Jiménez-Meroño & Soriguera, 2024a]. The script must be run for each simulation result data file. In order to ease the process, the simulation data filename must be introduced directly in the corresponding line of the script.

```
43    %% SIMULATION RESULTS DATA FILE PATH
44    %-------------------------------------------------------------------------
45    %          INTRODUCE HERE FILE NAME AND PATH OF SIMULATION DATA
46    %-------------------------------------------------------------------------
47
48 -  results_filename = 'Results\PAIRWISE - Err36\pwise_e36_sim1_Final_City_Variables.mat';
49
50
```

Figure 3. *Input of simulation data result file in “*RepoExperimentOutput*.m”.*

The script will automatically create an output folder with the postprocessed performance results of the vehicle-sharing system. The process is analogous to that explained in Section 6.2.3. of the Vsharing User Manual 2.0.0. Therefore, the structure will be equivalent, and the folder’s name will be also automatically generated from the simulation data filename with the suffix “_output”.

Two results are generated and stored inside the outputs folder:

- *System-level results* => These are the overall KPIs for the whole system. They are stored in the summary excel file “Table_summary.xlsx”.
- *Station-by-station results* => These are KPIs that consider a single station. In this particular case, the calculated KPIs are the % of time full and % of time empty for each station. They are stored in an excel file called “TableFullEmpty.xlsx”.

4) Aggregation

Results should be aggregated in order to consider the whole set of days with same repositioning strategy and demand forecasting errors. The aggregation procedure is different depending on the type of results:

- *System-level results* => For system-level KPIs, the values from the result summary files "Table_summary.xlsx" are aggregated manually. Average values must be obtained as weighted averages, where the weights are the total demand for each considered period.
- *Station-by-station results* => In the case of station-by-station KPIs, a new excel file must be created with the same structure to the station-by-station results "TableFullEmpty.xlsx". A way to do this is to duplicate one of the station-by-station excel files and substitute manually the KPIs values with their corresponding aggregated values (i.e. average values obtained as a demand-weighted average).

5) Run the figure creation script "RepoExperimentFigures.m".

The "RepoExperimentFigures.m" script creates two figures from the aggregated station-by-station results. These are the geographical representation of the [%] of time the stations remained full or empty during the simulation duration (see Figure 4), and which can also be found in [Jiménez-Meroño & Soriguera, 2024a].

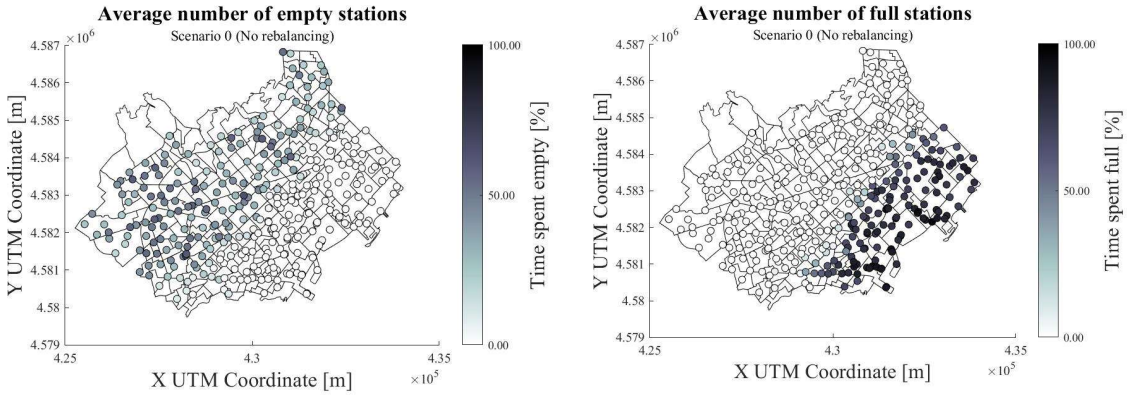

Figure 4. *Input of simulation data result file in "*RepoExperimentOutput*.m".*

The required user inputs to run the "RepoExperimentFigures.m" script include:

- *Simulation data filename* ("results_filename") => This is used to obtain the stations' IDs and locations. Note that any simulation data file from the experiment would suffice, since in all cases the stations remain the same.
- *Figure name* ("figure_filename") => Any character string. It is advisable to introduce a name that identifies the repositioning strategy and demand forecast error. The figure outputs will include the suffix "_full" or "_empty" to their names.
- *Station-by-station result file* ("table_filename") => Path and name to the aggregated "TableFullEmpty.xlsx" Excel file created in the previous step to aggregate station-by-station KPIs.
- *Table subtitle* ("table_subtitle") => Any character string to print as subtitle of the figure. It is advisable to introduce here something that identifies the repositioning strategy or whatever that defines the experiment.

```
%% SIMULATION RESULTS DATA FILE PATH
%--------------------------------------------------------------------------
%           INTRODUCE HERE FILE NAME AND PATH OF SIMULATION DATA
%--------------------------------------------------------------------------

results_filename = 'Results\PAIRWISE - Err36\pwise_e36_sim1_Final_City_Variables.mat';
figure_filename = 'pwise_e36';
table_filename = 'Results\PAIRWISE - Err36\AggregatedTableFullEmpty_pwise_e36.xlsx';
table_subtitle = 'Scenario 1 (Real-time pairwise assignment)';

```

Figure 5. *Inputs of the figure creation script "*RepoExperimentFigures*.m".*

The script will automatically create the map figures and save them in the "Results" folder.

REFERENCES

- Jiménez-Meroño, E. & Soriguera, F. (2024a) "Optimization of bike-sharing repositioning operations: A reactive real-time approach." Submitted for publication to *EURO Journal on Transportation and Logistics*.
- Jiménez-Meroño, E. & Soriguera, F. (2024b). "Agent-based simulation of vehicle-sharing systems". *Journal of Simulation*. In press. https://doi.org/10.1080/17477778.2024.2304549
- Jiménez-Meroño, E. & Soriguera, F. (2020) Vsharing Simulator User Manual (ver. 2.0.0).