# OpenOPorto: A Review of MatSim Model Creation Methodologies Applied to Porto, Portugal

Iohan Sardinha
Faculty of Engineering of University of Porto
Porto, Portgugal
up201801011@fe.up.pt

## ABSTRACT

MATSim is a popular and efficient multi-agent transport simulation that has been used in many scenarios worldwide. OpenOporto is the creation of a MATSim model for the city of Porto, Portugal. Alongside updating the methodology for creating MATSim models, that could hopefully be reproduced in other cities and regions of the world, considering that the book contains some out-of-date information by now, and information on model creation is scattered around the web and can be complicated to find for beginners trying to understand the simulation.

## KEYWORDS

MATSim, Microscopic Traffic Simulation, Activity Based Simulation, Porto, Portugal

## 1 INTRODUCTION

The exploration of mobility simulation is a pivotal subject, with contemporary cities providing a wealth of open data conducive to developing sophisticated models. MATSim, a microscopic activity-based transport simulator from the Technical University of Berlin, has found widespread application across diverse global scenarios. This tool serves as a cornerstone for crafting simulations that contribute to understanding urban and regional traffic dynamics, enabling predictions and comprehensive analyses in response to evolving conditions.

This paper presents the OpenOporto scenario, an endeavor to construct a model tailored to Porto, Portugal. Unlike conventional emphases on simulation outcomes, the primary focus of OpenOporto lies in the intricacies of model creation, leveraging openly available data. The overarching goal is to establish a methodology applicable to diverse scenarios for developing models spanning cities and regions. While existing literature, particularly the MATSim book, outlines a comprehensive process for model creation, OpenOporto seeks to streamline and update this methodology. The intention is not only to simplify the intricate process but also to address any outdated aspects resulting from the publication date of the MATSim book in 2016. OpenOporto aspires to contribute a refined and contemporary approach to urban mobility modeling through these efforts.

## 2 AVAILABLE RESOURCES

Developing OpenOporto, key literature, and existing MATSim scenarios served as foundational resources. The comprehensive guide provided by the MATSim book, authored by Horni et al. (2016)[1], served as a primary reference, elucidating the intricate functionalities of the simulation. Complementing this, the MATSim user guide[2], extracted from the book, synthesized essential information, forming a valuable compilation for our development endeavors.

One prominent influence on OpenOporto's development was the analysis of MATSim scenarios, particularly the exemplary OpenBerlin scenario[3]. This scenario, crafted by the MATSim development team, stands out as a notable creation, utilizing open data to simulate Berlin, encompassing the German capital and its surrounding cities. Another influential scenario, the Zurich Scenario[4], developed by the same team, played a significant role in our understanding, drawing references from the comprehensive MATSim book.

Drawing inspiration from recognized models, the Greater Toronto Area[5] and Santiago de Chile[6] scenarios were pivotal. Widely cited in various articles, these scenarios offered valuable insights into the nuanced model development process, enriching our understanding.

Exploring diverse perspectives, the North of France Scenario[7] presented intriguing analyses within MATSim simulations. It delved into hypothetical scenarios like inner city taxation for traffic reduction and free public transport. Additionally, the Barcelona Scenario[8] demonstrated an innovative approach to constructing population data from mobile communication data, though access to such data may not always be open.

In data acquisition, two primary sources played a central role in OpenOporto's development for the city of Porto. Leveraging the 2021 Portuguese Census from the National Institute of Statistics[9] provided a robust demographic foundation. Complementing this, the open data offered by Porto's municipal chamber through the Porto Digital Project[10] furnished a valuable dataset for our modeling efforts.

## 3 METHODOLOGY

MatSim has three required input files for defining the city model: config, network, and plans. Additionally, two more files are needed to add public transportation to the simulation: the transit schedule and the transit vehicles. The config file is the main one, fed to the program; it contains information about the other files and the simulation configurations, such as the number of iterations. The network file describes the world in which the agents will travel; it contains the information of the two components of a graph (Nodes and Links) and additional information relevant to the simulation. The transit vehicles have information about the public transport vehicles, such as their capacity, and the schedule contains the different public transport lines, routes, and timetables.

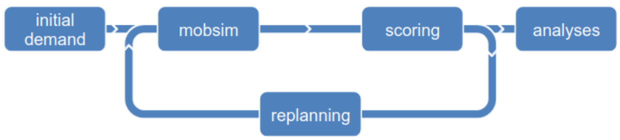


**Figure 1: The MATSim Life-cycle**

Finally, the plans contains the list of activities done by the population; each person has a list where each activity has a time to start and the way of locomotion to the next activity in the list. This is the core of the simulation; the agents can have a vast amount of information added to improve the simulation outcome. The simulation consists of the agents trying to follow their schedule of activities and re-planning the time of leaving, routes, and modes of transportation.

While constructing and experimenting with the model, the Simunto Via[11] visualizer was used; it is a simple-to-use tool to visualize the results of a MatSim simulation and is very helpful in creating the model. Nevertheless, Via is a commercial application that limits the visualization to five hundred total agents on its free license. Additionally, viewing the information graphically is more valuable in the creation process than in a full-scale analysis of all the simulation parts. Finally, a free alternative called OTFVis is referenced in the book as a bit harder to set up than Via and more experimental but has no limits on agents. OTFVis was not tested for the OpenOporto scenario, though.

The primary sources for the file creation are the MatSim Book and User guide and the T.U.Berlin course slides and recorded classes. The written materials are some years old and have some deprecated information, and the class recordings can be exhaustive in finding information due to their length. The process followed to create OpenOporto intends to update some of the information while pointing out what is still up-to-date to provide an updated step-wise methodology for creating MatSim models.

## 3.1 Network

To create the network, first, it is necessary to have a source of this data; thankfully, Open Street Map(OSM)[12] has updated information for most of the world's road networks. This information is not directly available to download; thus, the best place to find it is through Geofabrik, as suggested in the book. It provides easy access to OSM data divided by regions of the world. The data size may vary from place to place; some countries may have only the whole country's data, and others may have subdivisions. If the region is small enough, it can be exported directly from the OSM site, but this only works for tiny areas and is not covered in this process.

For OpenOporto, the smaller region available in Geofabrik is Portugal. The downloaded file tends to be very large since it contains all the information for that region available in OSM. Adding additional information that can make the simulation and the whole process slower and more complex is unnecessary. So, it is good practice to crop the region, in this case, the city of Porto. The book suggests using Osmosis, a command line client for managing .osm

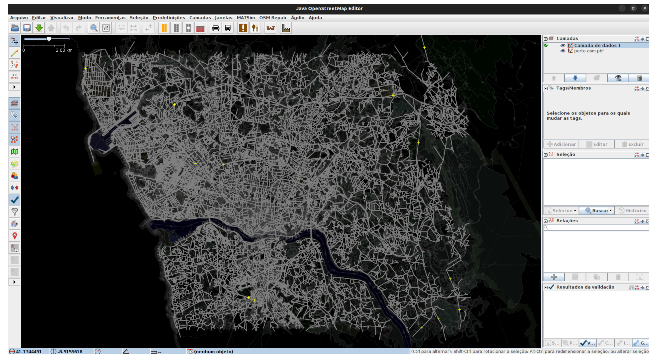

**Figure 2: A Screenshot of the conversion from OSM to MATSim network in JOSM**

files. However, Osmosis was discontinued in 2018 due to the creation of Osmium Tool[13] and has problems running on newer versions of Java.

Before using Osmium to crop the file, finding the limits of the area to be cut is necessary. This can be quickly done directly on the OSM site; by searching for a place by its name, it should show an outline on the map with the limits of that region. In the case of Porto, this works for the city, its neighborhood, and other important administrative subdivisions without much trouble. Nevertheless, this could be harder for regions with less available information. With the wanted area highlighted or centered on the map, it is possible to click on the export button. This action should give information on the latitudes and longitudes of the rectangle containing the wanted area. In the case of the regions without much information, there is an option for manually selecting the location.

With the bounding box information, it is straightforward to run the osmium extract command as shown below:

```
osmium extract -b west,south,east,north
    input.osm.pbf -o output.osm.pbf
```

Finally, to convert from OSM format to MatSim XML network, the book recommends using a class available at their matsim-code-examples GitHub. However, this code uses a class deprecated in the currently available version of the examples repository. Another class called 'RunCreateNetworkFromOSM' is also available in the code examples, but upon trying with the Porto, cropped data raised exceptions, not running properly or generating the output. To generate the network for the OpenOPorto scenario, JOSM[14], an open software to edit OSM data, was used. It has a plugin between its plugins called 'matsim,' which adds the option to convert between the formats simply. It is important to note that the plugin uses JavaFX, and after installing it, the user needs to run JOSM with the path to it to work correctly, as in the example below:

```
java -jar --module-path
    /usr/share/openjfx/lib --add
    modules=javafx.base, javafx.controls,
    javafx.fxml, javafx.graphics,
    javafx.media, javafx.swing, javafx.web
    josm.jar
```

It is also important to note that Geofabrik offers the OSM files in the '.osm.pbf' file format, which is not a problem for Osmium but is not readable by JOSM by default, so alongside the MatSim plugin, it is also necessary to install a plugin called 'pbf.' With both plugins installed, the OSM file can be opened, and the new MatSim tab can be used to convert the layer to a MatSim XML network that can then be saved. The result can be loaded into Via for testing.

## 3.2 Public Transport

The main suggestion from the book and the T.U.Berlin's course for adding public transport is to use GTFS format, which is widely used worldwide. Porto Digital has a portal for open data that offers the wanted files for the city metro and the bus lines. Both sources cite a repository in MATSim's GitHub called 'GTFS2MATSim' that can convert between the formats.

After running the converter, two main issues were found: mismatched map projections and wrong stop times. The first issue originated from the JOSM network conversion; the resulting network was created using Web Mercator(EPSG:3857) coordinates, although OSM data should be in World Geodetic System(WGS84) coordinates. This appears to occur since the JOSM plugin does not have information about the system, and it defaults to EPSG:3857 since it is possible to convert to WSG84 if another reference layer is added with OSM. However, doing this generated errors during the simulation, so the solution was to convert the GTFS files to EPSG:3857; this kind of issue is common, and MATSim code has a class for converting between coordinates, that is already present in the conversion tool, so it is just a matter of setting the coordinate systems codes. Via can be used to test if the networks generated match by simply loading both networks in two separate layers.

The second issue is that the schedule generated contained departures with unsupported times after midnight. These issues seem to come from the GTFS schedule containing these times; however, they are neither removed in the conversion nor detected by the schedule validator available in the MATSim GUI class. If these are present in the file when fed to the simulator, an error will occur, so the solution is to remove it, either through code or by replacing it using a text editor. For the OpenOporto scenario, a text editor was used. Additionally, when using the Schedule Validator, there is a text saying that the network is optional, but the program produces an error if it is not given.

## 3.3 Plans

Generating the plans file is the most complex and important task. The literature gives various examples, such as OpenBerlin[3] and Zurich Schenario[4], which uses a traffic survey that can be almost directly translated into MATSim files, the Barcelona Scenario[8] uses mobile data from phones to reproduce the population displacement. Some of these are directly referenced in the book and have examples in the code-examples repository. However, these examples rely on very specific data from a format and information perspective, and none matched the information available for Porto.

INE, the Portuguese National Institute of Statistics, has a variety of data available; however, there is not much about inner city movements. Most of the data for Portugal is provided at the city level, which is not useful for OpenOporto. There are many different and complex methods for activity generation from census data, such as [15]. For Porto, a simple method was used due to the available data and time constraints. The census offers data for the outward commutes from the neighborhoods of Porto, giving for each neighborhood how many people work in the same neighborhood, in another neighborhood in Porto, or other cities.

The population file is created from this information by generating an initial number of agents randomly distributed in each neighborhood proportional to the census data. Then, a work location is generated proportional to the number of people working in the same neighborhood or another in a random location in either option. Creating a population that follows the census reduced to 10% of the total for simulation performance. Since there is no information about which other neighborhood is the work destination, this is also randomly selected. Based on the census, the agents do a simple commute from their place of origin to their workplace, with the work time selected to start at a random hour between 6 and 12 and end between 16 and 20.

The information about the shapes of the neighborhoods to generate the random points inside it is extracted from OSM through OpenStreetMap Polygons[16]. This process is very manual and should be automated in the future. Porto has only seven neighborhoods, but this would not scale properly for larger places or places with more subdivisions. OSM Polygons use WSG84 and must be converted to EPSG:3857, which can be done with a simple code available in the OpenOporto GitHub repository. Not adjusting the coordinates will cause the simulation not to work properly, but it could possibly not generate errors, just agents that never reach the network, so do its movement by walking.

The plans generation through processing data is done in Python even though MATSim is in Java, and the book recommends all scripts to be done using it. Python offered a faster and simpler way to generate the initial files that could then be fed to Java MATSim code, especially considering time constraints.

## 3.4 Putting everything together

Merging the created files into just the necessary ones is the final step to having a complete model. This can be done in various ways, and the best would be using Java and the MATSim project classes to do so, for OpenOporto text editors were used alongside regular expressions to do most of the work. This more manual approach could be more time-consuming but helps the user become familiar with the MATSim XML file formats and is a good step before automating the work.

For OpenOporto, three network files were merged into the final one: the roads, bus, and metro lines. The first step is to guarantee that the IDs don't repeat over the networks to be merged; the simpler solution is to use a tool for finding and replacing regular expressions to add something like '_metro' and '_bus' after the IDs. An example that works in Visual Studio Code is shown below:

```
FIND:    id="pt_([0-9]*)"
REPLACE: id="pt_$1_metro"
```

It is important to be careful when replacing these IDs since they need to be replaced in different locations, and the necessity of replacing each name can vary depending on whether the original source data had names for the nodes and links or if the converters sequentially number them. Reading and understanding the files before doing these operations is simple and very helpful:

- Node IDs in Netwok.xml
- Link IDs in Network.xml
- Node Reference (from and to) in the Links in Network.xml
- Link Reference (LinkRefID) in the TransitStops in TransitSchedule.xml
- Link Reference (RefID) in the Route Links in TransitSchedule.xml

Another detail to pay attention to is the 'modes' attribute in the links; this needs to be different for each network being merged if using this method because not doing so would cause an incomplete network that breaks the simulation. This is not a problem for the metro, but it has the downside of creating separate bus and car networks, which is unrealistic for the simulation. To achieve this, a replacement can be done on the original file of each network, for example, from 'modes="car,bus"' to 'modes="bus"'.

Once the IDs are treated, merging the networks is as simple as copying and pasting all the nodes from two other files into the '<nodes>' XML element of the third one. And repeating the same process with the links in the ´<links>' element. It is possible for visual inspections in Via to have more than one 'nodes' or 'links' element in a way that would make merging the files a bit easier, but this does not work as input for the MATSim simulation.

To merge the transit schedule and vehicles, it is necessary to do some similar replacements with the same remarks.

- VehicleType ID in TransitVehicles.xml
- network mode name of NetworkMode of VehicleType in TransitVehicles.xml
- vehicle id in the vehicles of TransitVehicles.xml
- vehicle reference id (vehicleRefId) in departures in TransitSchedule.xml

With the IDs fixed, the transit vehicles are straightforward to merge, just putting all the information together in the same file inside the 'vehicleDefinitions' element. While the transit schedule is a bit more complex, first, the stops from both schedules need to be merged inside the '<transitStops>' element, then following it, the '<transitLine>' elements can come one followed by the other with no necessary order.

Finally, everything needs to be aggregated in the config file; the default config file for the MATSim 'equil' scenarios is a good starting point. The OpenOporto configuration can also be used as a reference and is available on GitHub. The main elements to pay attention to are the transit element, the addition of the 'mainMode' parameter to the ´qsim' element, the addition of the 'changeMode' module, and the use of this module in the 'strategy' module.

The OpenOporto scenario used a 30% chance of changing the mode of transportation and a 10% chance of rerouting for the strategy. A total of 10 iterations, since the main focus of the work was on the creation of the model, and further simulation improvements and analysis are left for future work. And a flow capacity factor of 10% since the population created was 10% of the census population.

Finally, to test the simulation, the MATSIM GUI available at the matsim-example-project was used, in which the config file can be input, and it will run the simulation accordingly. Via was used to inspect the results, alongside some minor visual analysis of the generated output files as shown in Figure 3.

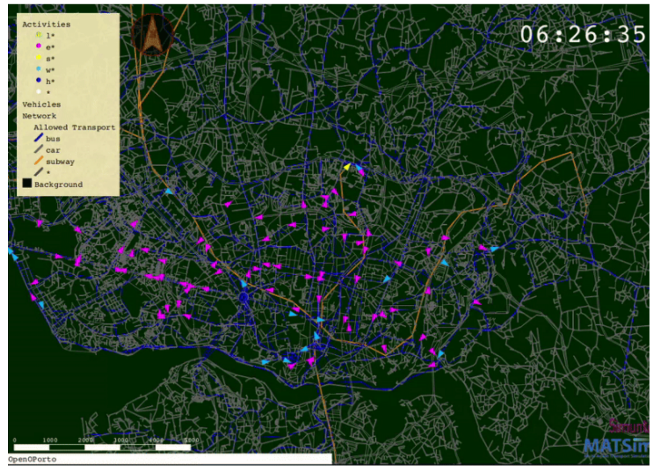


**Figure 3: Screenshot of Final results in Via**

## 4 FUTURE WORK

The OpenOporto work has some limitations due to time constraints, simplifications, and a focus on the modeling that could be improved in future works. A method for cropping the network according to the shape of the studied area could help improve performance and be useful in some scenarios. Additionally, a pipeline for the process of cropping, merging, and treating the information from the different inputs could be very useful and reusable in future scenarios in other cities. This pipeline would ideally fit into MATSim philosophy by being developed as a Java module for the simulation.

A method for merging the bus and car networks could be developed, possibly using Google Maps information, to cross the information from the GTFS to the street data. More complex population generation methods should be studied, adding more layers, some of which are available from the INE census, such as the percentage of use of each transportation method and social indicators like wealth and gender. Other synthetic methods can be used to extrapolate and generate data not directly available from the census.

Finally, the studied area could be expanded to include Porto's Metropolitan area instead of just the municipality, including information about places of work, shopping districts, and study places that can be found online. A complete visualization of the simulation using OTFVis could be useful in future iterations of the work. As well as a deep analysis of the results.

## 5 CONCLUSION

This work proposed OpenOporto, a model of the city of Porto, Portugal, using MATSim, and a methodology based on the available literature while updating any outdated information on it. The OpenOporto model uses only open and free resources for anyone to reproduce the process, including a GitHub page with the code used[17].

The biggest challenge to developing a methodology for model creation is population generation since the data varies broadly for each region to be studied, especially the census data format. For Porto, a simplistic commute activity plan was generated based on the available 2021 Census. The debugging process is also a challenge due to some lack of information in the literature.

The final process manages to generate a working model following a step-by-step process that contains population movement and public transit networks and can be a good starting point for the creation of models for most cities worldwide as well as the beginning of the construction of a deeper model to study the city of Porto and its surroundings.

## REFERENCES


[1] Andreas Horni, Kai Nagel, and Kay W. Axhausen, editors. *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press, London, 2016. License: CC-BY 4.0.

[2] Andreas Horni, Kai Nagel, and Kay W. Axhausen, editors. *The Multi-Agent Transport Simulation MATSim*. 2023. Extract of selected chapters to serve as user guide. — This version: July 13, 2023.

[3] Dominik Ziemke, Ihab Kaddoura, and Kai Nagel. The matsim open berlin scenario: A multimodal agent-based transport simulation scenario based on synthetic demand modeling and open data. *Procedia Computer Science*, 151:870–877, 2019. The 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019) / The 2nd International Conference on Emerging Data and Industry 4.0 (EDI40 2019) / Affiliated Workshops.

[4] Andreas Horni and Basil J. Vitins. The zurich scenario: A technical overview 1.0. *ETH Zurich*, 2011.

[5] Sk. Md. Mashrur, Brenden Lavoie, Kaili Wang, and Khandker Nurul Habib. A regional multimodal network microsimulation (gtasim) for a comprehensive utility maximizing system of travel options modelling (custom) in the greater toronto and hamilton area. *Procedia Computer Science*, 220:110–118, 2023. The 14th International Conference on Ambient Systems, Networks and Technologies Networks (ANT) and The 6th International Conference on Emerging Data and Industry 4.0 (EDI40).

[6] Benjamin Kickhöfer, Daniel Wesemeyer, Kai Martins-Turner, and Alejandro Tirachini. Creating an open matsim scenario from open data: The case of santiago de chile. 04 2016.

[7] Moez Kilani, Ngagne Diop, and Daniel De Wolf. A multimodal transport model to evaluate transport policies in the north of france. *Sustainability*, 14(3), 2022.

[8] Aleix Bassolas, José J. Ramasco, Ricardo Herranz, and Oliva G. Cantú-Ros. Mobile phone records to feed activity-based travel demand models: Matsim for studying a cordon toll policy in barcelona. *Transportation Research Part A: Policy and Practice*, 121:56–74, 2019.

[9] Ine census 2021. Available online at https://censos.ine.pt/ (Accessed January 2024).

[10] Open data porto digital. Available online at https://opendata.porto.digital/ (Accessed January 2024).

[11] Simunto via. Available online at https://www.simunto.com/via/ (Accessed January 2024).

[12] Open street maps. Available online at https://www.openstreetmap.org/about (Accessed: January 2024).

[13] Osmium tool. Available online at https://osmcode.org/osmium-tool/ (Accessed January 2024).

[14] Josm. Available online at https://josm.openstreetmap.de/ (Accessed January 2024).

[15] Samuel Felbermair, Florian Lammer, Eva Trausinger-Binder, and Cornelia Hebenstreit. Generating synthetic population with activity chains as agent-based model input using statistical raster census data. *Procedia Computer Science*, 170:273–280, 2020. The 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.

[16] Open street maps polygons. Available online at http://polygons.openstreetmap.fr/ (Accessed January 2024).

[17] Iohan Sardinha. Openoporto github repository. Available online at https://github.com/IohanSardinha/OpenOPorto.