Metropolis Website Tutorial

Last version: December 14, 2018

Contents

1 Overview

Metropolis is a *mesoscopic* and *dynamic* traffic simulator that can handle large networks. The model was originally presented in de Palma, Marchal, and Nesterov (1997).

A website was developed in 2018 to make Metropolis accessible to more people. The website allows anyone to create a simulation, run it and view the results.

This document provides a summary of the theoretical foundations of Metropolis and a guide on how to use the website. A glossary is available at the end of the document.

2 Theoretical Foundations

Metropolis needs two main inputs: the network and the demand. The network is a description of all the nodes and links in the simulated area, together with the congestion model. The demand characterizes the behavior of the travelers and their trip (O-D matrix).

Metropolis is an iterative model, and each iteration corresponds to a simulated day.

2.1 Traveler Choices

The travelers choices are described at the individual level. Each commuter is characterized by a set of parameters which describes his/her behavior. The travelers are partitioned into segments. In each segment, the travelers have parameter values drawn from the same distribution. Travelers having different origins and destinations can belong to the same demand segment.

For instance, travelers can be classified in three segments: high incomes, average incomes and low incomes. If we assume that income is correlated with the value of time (VOT), then the VOT is specific to the segment. The VOT of travelers in the same segment can be different but they are drawn from the same distribution. These VOT are allocated for all travelers at the beginning of the simulations and do not change.

Each day, the travelers make three choices:

1. modal choice (car or public transit),
2. departure time choice (from the origin),
3. route choice (before the departure and en-route choice).

2.1.1 Modal Choice

To compute modal choice, METROPOLIS needs, for each O-D pair, the travel time in public transportation. The modal choice can be enabled only for some demand segments. When enabled, the modal choice is described by a discrete choice model.

The generalized cost associated to public transportation, for a given O-D pair, is

$$C_{PT}(O, D) = VOT_{PT} \cdot tt_{PT}(O, D) + P_{PT}$$

where

- VOT_{PT} is the value of time spent in public transportation (euros per hour),[1]
- $tt_{PT}(O, D)$ is the generalized travel time in public transportation from O to D (hour),
- P_{PT} is a fixed penalty associated to public transportation (euros).

The parameters VOT_{PT} and P_{PT} are specific to each traveler, while the travel time $tt_{PT}(O, D)$ is specified by the planner and is identical for every traveler.

The generalized cost associated to the car can take two forms according to whether it is a short term or a long term choice. In the case of a short term choice, the model computes, given an O-D pair, the hour $t(O, D)$ at which the user would leave if he chooses to travel by car. Then, the model computes the generalized cost $C(t(O, D))$ associated with that departure time. The traveler commutes by car if $C(t(O, D)) < C_{PT}(O, D)$.

In the case of a long term choice, the traveler is unaware of his/her potential departure time. Then, the generalized cost associated to the car is aggregated in time and equal to the accessibility A, defined in the following section.

[1] For simplicity, we assume that the value of time and the penalties are measured in euros but it is possible to choose any monetary unit. If the input of METROPOLIS are measured in euros, the results will also be expressed in euros. Similarly, we assume that the metric system is used to measure length and speed but other units of measurement can be used.

2.1.2 Departure Time Choice

Travelers who commute by car have to select their departure time. For a given traveler commuting from O to D, let

- $tt(t)$ be the travel time from O to D for a departure at time t (hour),
- t^* be the desired arrival time at destination D (hour),
- δ be the flexible time period without penalty (hour),[2]
- α be the value of time in a car (euros per hour),
- β be the unit penalty associated to early arrival (euros per hour),
- γ be the unit penalty associated to late arrival (euros per hour).

The generalized cost function, for a departure time t, is composed of three terms:

$$C(t) = C_1(t) + C_2(t) + C_3(t).$$

The first term represents the travel time penalty:

$$C_1(t) = \alpha \cdot tt(t).$$

The second term represents the early arrival penalty (it is null if the traveler arrives late or on time):

$$C_2(t) = \begin{cases} \beta \cdot \big((t^* - \delta/2) - (t + tt(t))\big) & \text{if } t + tt(t) < t^* - \delta/2 \\ 0 & \text{if } t + tt(t) \geq t^* - \delta/2 \end{cases}$$

The third term represents the late arrival penalty (it is null if the traveler arrives early or on time):

$$C_3(t) = \begin{cases} \gamma \cdot \big((t + tt(t)) - (t^* + \delta/2)\big) & \text{if } t + tt(t) > t^* + \delta/2 \\ 0 & \text{if } t + tt(t) < t^* + \delta/2 \end{cases}$$

In METROPOLIS, the departure time choice model is stochastic because travelers do not necessarily choose the departure time with the lower expected cost.[3] The probability $P(t)\mathrm{d}t$ of choosing the departure interval $[t, t + \mathrm{d}t]$ is given by a continuous logit model:

$$P(t)\mathrm{d}t = \frac{\exp(-C(t)/\mu_d)}{\int_{T_0}^{T_1} \exp(-C(u)/\mu_d)\mathrm{d}u}\mathrm{d}t$$

where the parameter μ_d measures the heterogeneity of the departure time choice in the population. The integration bounds are defined so as to cover the entire simulation period.

The probability of departure is a function that can have several maxima. Each maximum corresponds to a departure time that is locally most probable. For example, one can adopt several arrival strategies to be at destination on time with large travel times or early with smaller travel times.

The logsum of this logit model provides a definition of accessibility that is independent of time but that aggregates within-day travel time fluctuations:

$$A = \mu_d \ln \int_{T_0}^{T_1} \exp(-C(u)/\mu_d)\mathrm{d}u.$$

[2] If the traveler arrives at destination during the period $[t^* - \delta/2, t^* + \delta/2]$, he/she does not face any penalty.

[3] Theoretically, this can be explained by unobserved heterogeneity between the travelers. If the choice model was deterministic, METROPOLIS might not converge to a stable equilibrium.

2.1.3 Route Choice

Metropolis uses a model of route choice based on point-to-point dynamic travel times. Two types of information are available to the travelers: historical travel time and instantaneous (or simulated) travel time. Historical travel times are the result of a learning process whereas current travel times reflect the situation of the current day.

The travelers make two types of choices:

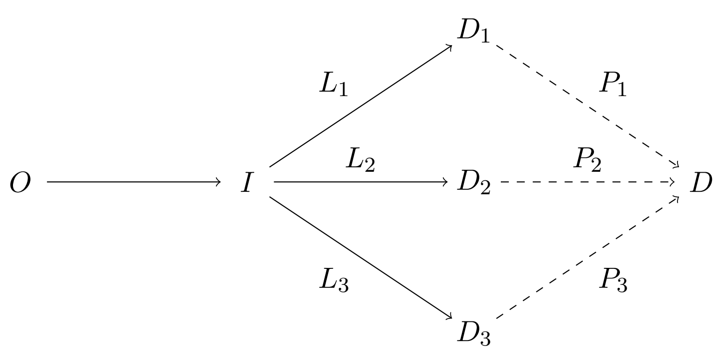
- pre-trip decisions: a joint departure time and route decision at the origin of their trip,
- en-route decisions, a direction choice at each intersection during their trip.

Pre-trip decisions The choice of an itinerary at the origin is only based on the minimization of historical travel times. The travelers observe the travel times of each link during the whole period, for each previous day. They simply select the shortest path from the origin to the destination.

En-route decisions It is assumed that, at each intersection, the travelers observe the travel time on the next link to take. For example, when reaching an intersection I with 3 downstream links (L_1, L_2 and L_3), the travelers choose the link that minimizes the remaining travel time to destination. When choosing the link L_j, the travel time from I to D is the sum of the current travel time on the link L_j plus the historical travel times of the path P_j from D_j to D:

$$tt_j(t) = tt^S_{L_j}(t) + tt^H_{P_j}(t + tt^S_{L_j}(t))$$

where the indices H correspond to the historical travel times and the indices S correspond to the simulated (i.e. current) travel times.



It is also possible to use stochastic direction choice. In this case the choice probabilities are given by a logit type function specified by its own heterogeneity parameter:

$$P(i) = \frac{\exp(-tt_i(t)/\mu_r)}{\sum_{j=1}^{J} \exp(-tt_j(t)/\mu_r)},$$

where J is the number of downstream links.

2.2 Congestion Model

The congestion model of Metropolis is mesoscopic in the sense that it uses individual vehicles, like a microscopic model, and aggregated congestion laws, like a macroscopic model. Each link is described by a congestion law. This congestion law relates the instantaneous occupancy of the link, the incoming flow and the travel time delay to cross it. This relation, also called volume-delay function, is evaluated each time a vehicle enters or leaves a link of the network.

The travel time on a given link at time t is given by

$$tt(t) = f[\text{dynVol}(t), \text{dynFlo}(t); p_1, p_2, \dots]$$

where

- f is the function which describes the congestion law,
- dynVol(t) represents the number of vehicles on the link at instant t,
- dynFlo(t) represents the incoming flow at the same moment,
- p_1, p_2, ... are the parameters defining the link (length, capacity, etc.).

2.3 Learning Model

In Metropolis, it is assumed that travel times experienced one day by the travelers affect their decisions the next day. For this reason, Metropolis uses an iterative process where each iteration corresponds to one day. This process is a learning process where travelers acquire knowledge about the congestion of the network and adapt their choices accordingly. This process operates as follows:

- On the first day, travelers are naive and assume that there is no congestion in the network. Consequently, they leave relatively late, at the same time, and select the shortest route. The congestion caused by this concentration phenomenon is very high.
- On the second day, they leave much earlier or later in order to avoid congestion. They also select longer (less congested) routes. Consequently, the congestion is reduced. Yet, they do not all arrive at their desired arrival time t^*.
- The process continues until it reaches a stable state. At each stage, the travelers acquire information represented by historical travel times.

The historical information, $X^H(\omega + 1)$, acquired on day $\omega + 1$ is

$$X^H(\omega + 1) = f(X^H(\omega), X^S(\omega))$$

where $X^S(\omega)$ represents the traffic conditions effectively simulated/incurred on day ω. Several functions f can be used to simulate the updating process. Metropolis offers four options.

1. Exponential process:
$$X^H(\omega + 1) = (1 - \lambda)X^H(\omega) + \lambda X^S(\omega).$$
2. Linear process:
$$X^H(\omega + 1) = \frac{\omega}{\omega + 1} X^H(\omega) + \frac{1}{\omega + 1} X^S(\omega).$$
3. Quadratic process:
$$X^H(\omega + 1) = \frac{\sqrt{\omega}}{\sqrt{\omega} + 1} X^H(\omega) + \frac{1}{\sqrt{w} + 1} X^S(\omega).$$
4. Genetic process:
$$X^H(\omega + 1) = \sqrt[\omega+1]{X^H(\omega)^\omega \cdot X^S(\omega)}.$$

3 Step-by-Step Guide

The METROPOLIS website allows you to create a simulation, to edit the network and the demand, to run the simulation and to view the aggregated and disaggregated results.

To use the website you just need a computer with a network connection and a web browser (preferably a recent version of *Mozilla Firefox* or *Google Chrome*).[4]

Here are the 7 steps to follow in order to create and run your own simulation:

1. Create an account and log in.
2. Create a new simulation or copy an existing simulation.
3. Add or modify the network.
   (a) Zones and intersections.
   (b) Congestion functions.
   (c) Links.
   (d) Public transit.
4. Add or modify the behavior of the travelers.
   (a) Parameters.
   (b) Origin-destination matrix.
5. Add or modify the technical parameters of the simulation.
6. Run the simulation and look at the aggregated results.
7. View the disaggregated results.

Step 1: Create an account or log in
Without an account, you can view the public simulations but you cannot create your own simulation. To create an account, click on *create an account* on the top-right corner of the home page and enter your username, your email address[5] and your password (Figure 3.1).

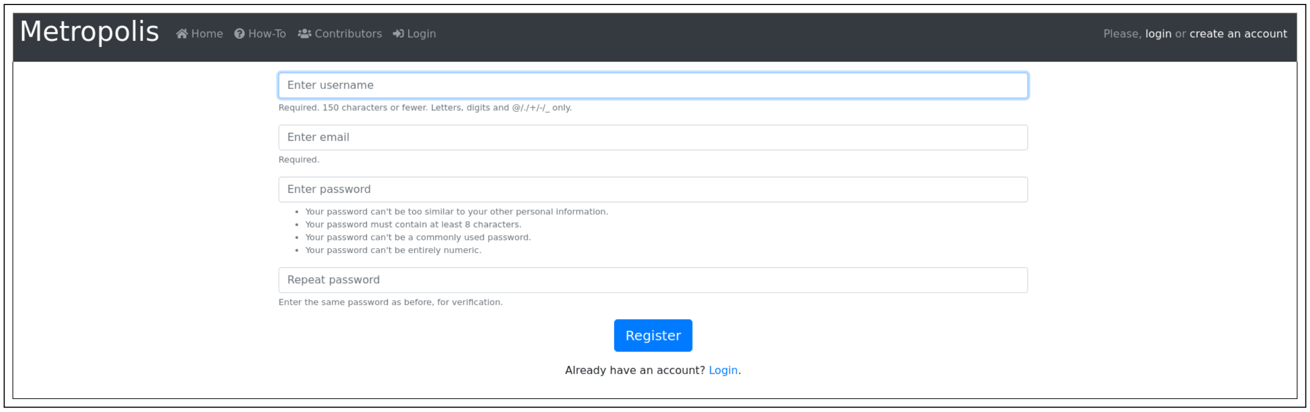


Figure 3.1: Page to create an account

If you already have an account, use the *login* button to connect. After you have entered your username and password, you will be redirected to the home page of METROPOLIS where you can see a list of the public simulations. If you create a simulation, it will appear on this page (Figure 3.2).

[4] *Javascript* must be installed on the web browser, this is usually the case.

[5] For now, the email address is not used; in the future, it could be used to notify you when your simulation run has ended.

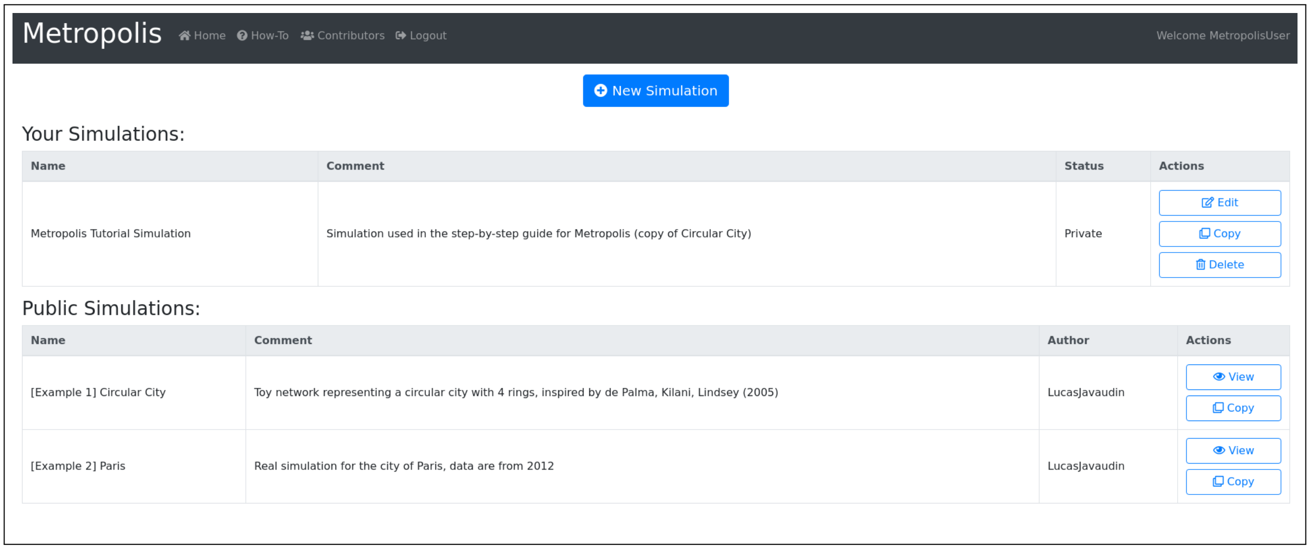


Figure 3.2: Home page of METROPOLIS

Step 2: Create a new simulation or copy an existing simulation

From the home page of the website, you can use the *View* button to view the input and output of any public simulation and you can use the *Copy* button to copy any public simulation. Creating a copy of a simulation allows you to edit and run it without modifying the original simulation.

A copied simulation has the same network, travelers and technical parameters as the original simulation at the time of the copy. The results of the runs of the original simulation are not copied. If the input of the original simulation is modified after the copy, it does not modify the copy.

If you do not want to use or modify an existing simulation, you can use the *New Simulation* button at the top of the home page to create an empty simulation.

You must give a name to your simulation and you can, optionally, add a comment. You can also choose your simulation to be public or private. Private simulations are hidden for everyone except for the user who owns the simulation. For public simulations, everyone can view the input and output of the simulation, but only the owner can edit the input and run the simulation.

The name, comment and status (public or private) of the simulation can be changed at any time by using the *Modify* button on the main page of the simulation (Figure 3.3).

Step 3: Add or modify the network

You can create or modify the network from the main page of your simulation. At any time, you can view it with the *Network View* button (Figure 3.4).[6]

Below the *Network View* button, you can find a button for each element that constitutes the network (*zones*, *intersections*, *congestion functions*, *links* and *public transit*). Each button redirects you to a page where you can view and edit the network elements (Figure 3.5).

There are currently two ways to edit the network elements:

1. With the *Edit* button, you can manually add, modify or delete elements of the network. Note that this button is only available if the network is not too large (Figure 3.6).

2. With the *Import* button, you can import the elements of the network from a tab-separated values file (tsv file). The first row of the import file must be the header: a tab-separated list of the attributes. Then, there is one row for each element that you wish to import. The export files have the same

[6] A network must have at least 2 zones, 1 intersection and 1 link. To view the network, the server must generate it the first time and every time it is modified. This operation can take multiple minutes for large networks.

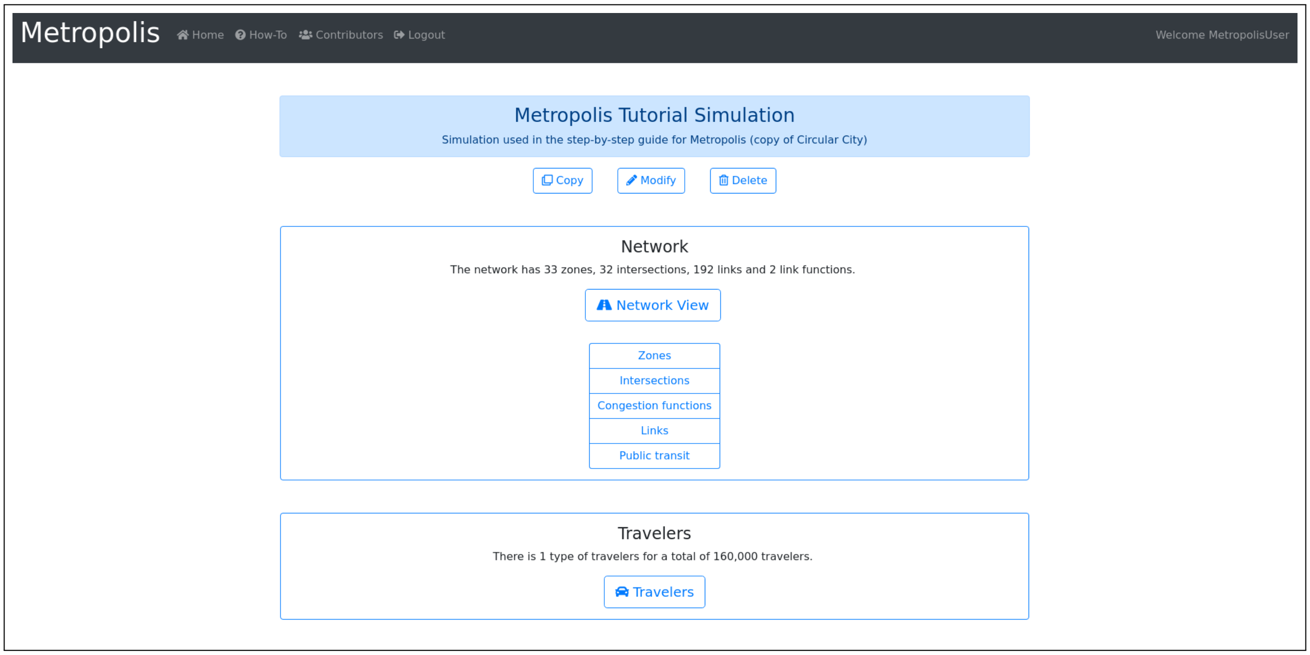


Figure 3.3: Main page of a simulation

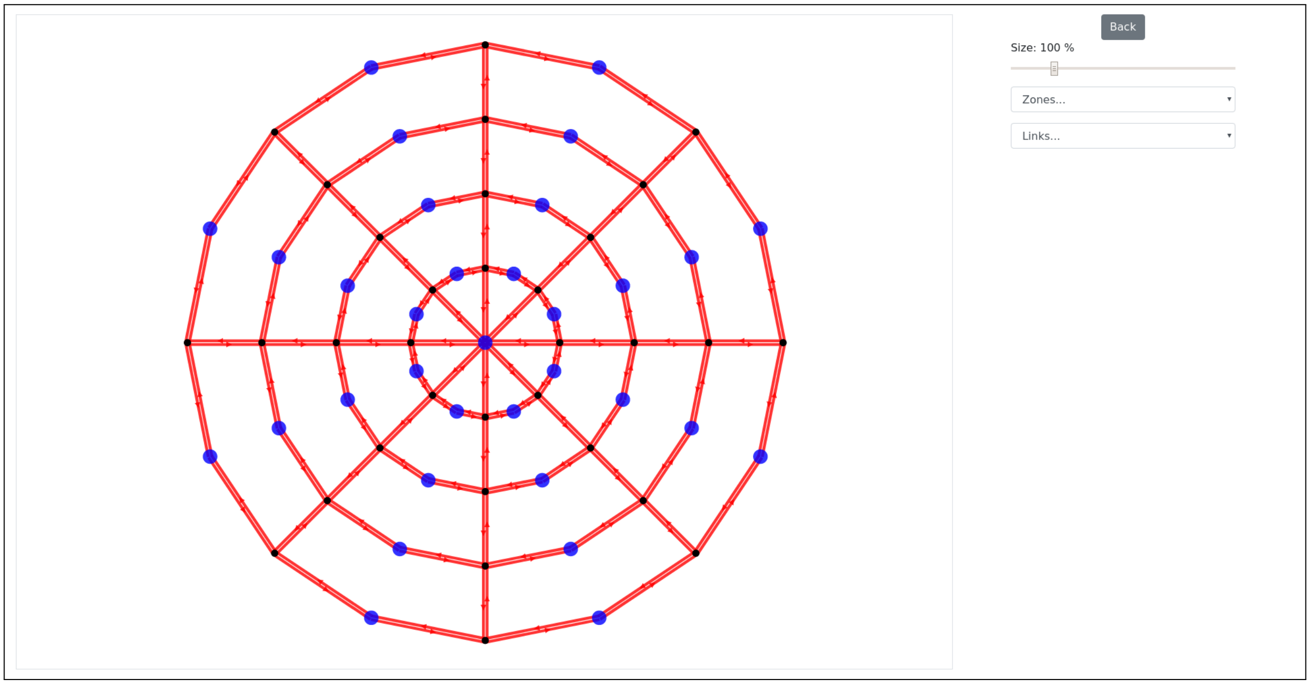


Figure 3.4: Network visualization

format as the import files so you can use the export button on an existing network to see how the import files must look like.

See Section 4.1.1 for a detailed description of the attributes of the network elements.

Each zone, intersection, congestion function and link has an id. The id of a zone or intersection must be unique among all zones and intersections.[7] The id of a congestion function must be unique among all

[7]To avoid creating one zone with the same id as an intersection, you can start numbering the ids of intersections at 1,000 (or 1,000,000 for large networks) so that all nodes with id lower than 1,000 are zones and all nodes with id greater than 1,000

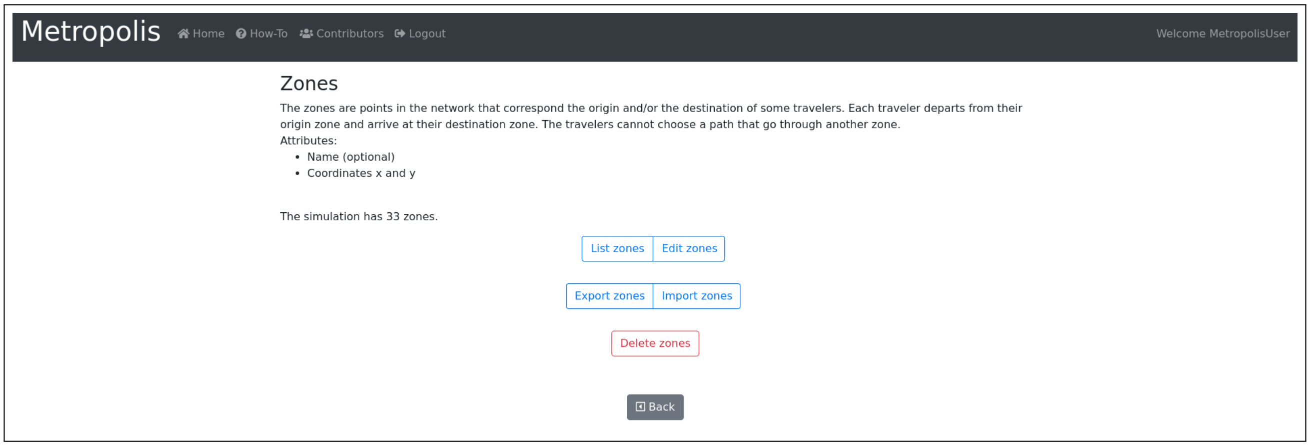


Figure 3.5: Page to view and edit the zones of the network

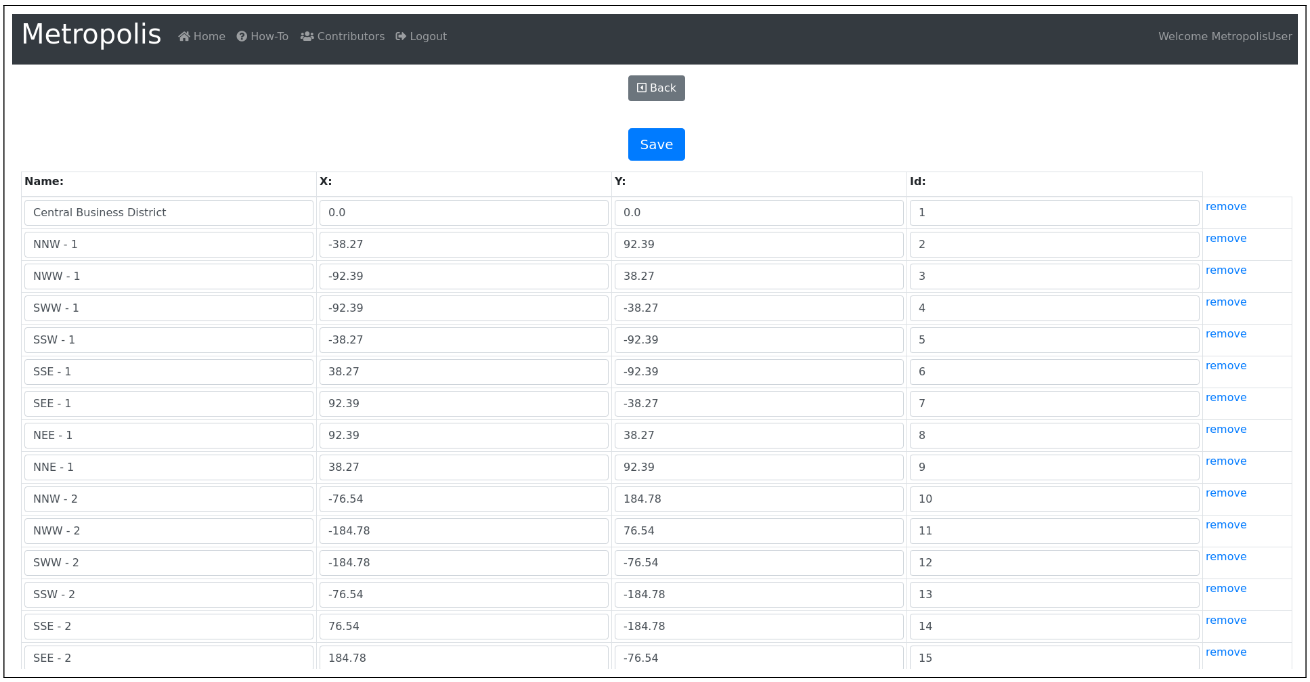


Figure 3.6: Page to manually edit the zones of the network

congestion functions. The id of a link must be unique among all links.

The ids for zones and intersections are used when importing links to specify the origin node and destination node of the links. The ids for congestion functions are used when importing links to specify the congestion function of the links.

When importing a file, the ids are used to determine if a particular element of the network must be created or modified. For example, assume that the current network of the simulation has 2 zones with id 1 and 2 and you import a tsv file containing 2 zones with id 2 and 3. Then the zone with id 1 will not be modified nor deleted, the zone with id 2 will be modified according to the new values in the import file, the zone with id 3 will be added to the network.

Step 3.a: Zones and intersections

List of attributes: *id*, *name*, *x*, *y*.

---

are intersections.

Zones and intersections are nodes in the network. The zones can be the origin or destination node of travelers while the intersections are only used to connect links.[8]

Step 3.b: Congestion functions

List of attributes: *id*, *name*, *expression*.

Congestion functions describe the congestion model of the links. They are used to compute the travel time of a link as a function of the link's attributes and the number of vehicles on the link.

When creating a simulation from scratch, two congestion functions are added by default (you can safely remove them if you do not plan to use them): free flow function and bottleneck function.

The *expression* attribute of the function can use the following operations, functions and parameters:

- the four base operations $+$, $-$, $*$, $/$;
- the exponent ^;
- the conditional tests $<$, $>$, $<=$, $>=$, $==$;
- the trigonometric functions *sin, cos, tan, asin, acos, atan, sinh, cosh, tanh*;
- the logarithmic functions *log, ln, exp*;
- the absolute value function *abs*;
- the link's parameters *length, speed, lanes, capacity*;
- the number of vehicles on the link *dynVol*;
- the flow of vehicles entering the link over a small time period *dynFlo*.

Step 3.c: Links

List of attributes: *id*, *name*, *lanes*, *length*, *speed*, *capacity*, *function*, *origin*, *destination*.

The links represent the roads (or edges) of the network. They connect the nodes (zones and the intersections) of the network. Each link is associated with a congestion function which describes the congestion model of the link.

When importing links from a tsv file, you must specify the function, origin and destination attributes using the id of the corresponding function, zone or intersection.

Step 3.d: Public transit

The public transit system of a network is fully described by the public-transit travel time between all possible origin-destination pairs of the network. The METROPOLIS simulator needs to know the public-transit travel time for the origin-destination pair of all travelers with modal choice enabled. If you do not intend to enable modal choice for any traveler, you can safely skip this step.

You can edit the public transit system by modifying the O-D matrix (only for small networks) or by using an import file with three columns (*origin*, *destination*, *travel time*). Use the id of the zones to specify the origin and destination of the O-D pair in the import file. Public-transit travel times must be specified in minutes.

Step 4: Add or modify the behavior of the travelers

Once you have finished adding or editing the network of your simulation, go back to the main page of the simulation and use the *Travelers* button to see a page with the list of traveler types (Figure 3.7).

Travelers are classified into different types (or segments). You can have as many traveler types as you want. For each traveler type, you need to specify the parameters of the travelers' preferences and the origin-destination matrix (O-D matrix).

[8] Note that travelers can drive from one link to another only if they are connected with an intersection, not a zone.

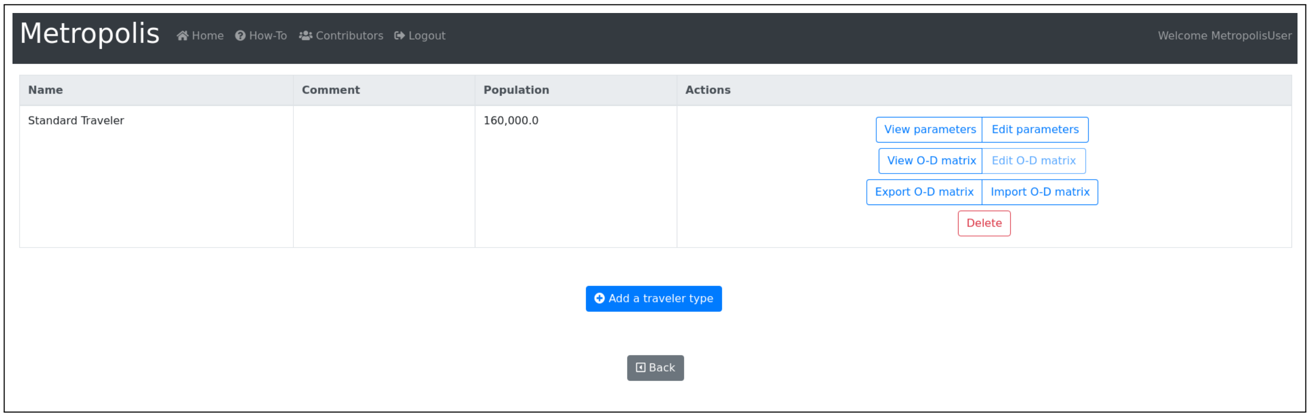


Figure 3.7: Page showing the traveler types of the simulation

Step 4.a: Parameters

Use the *Edit parameters* button of a traveler type to access the page where you can edit the parameters of this type (Figure 3.8). If you create a new type using the *Add a traveler type* button, you will be automatically redirected to the same page.

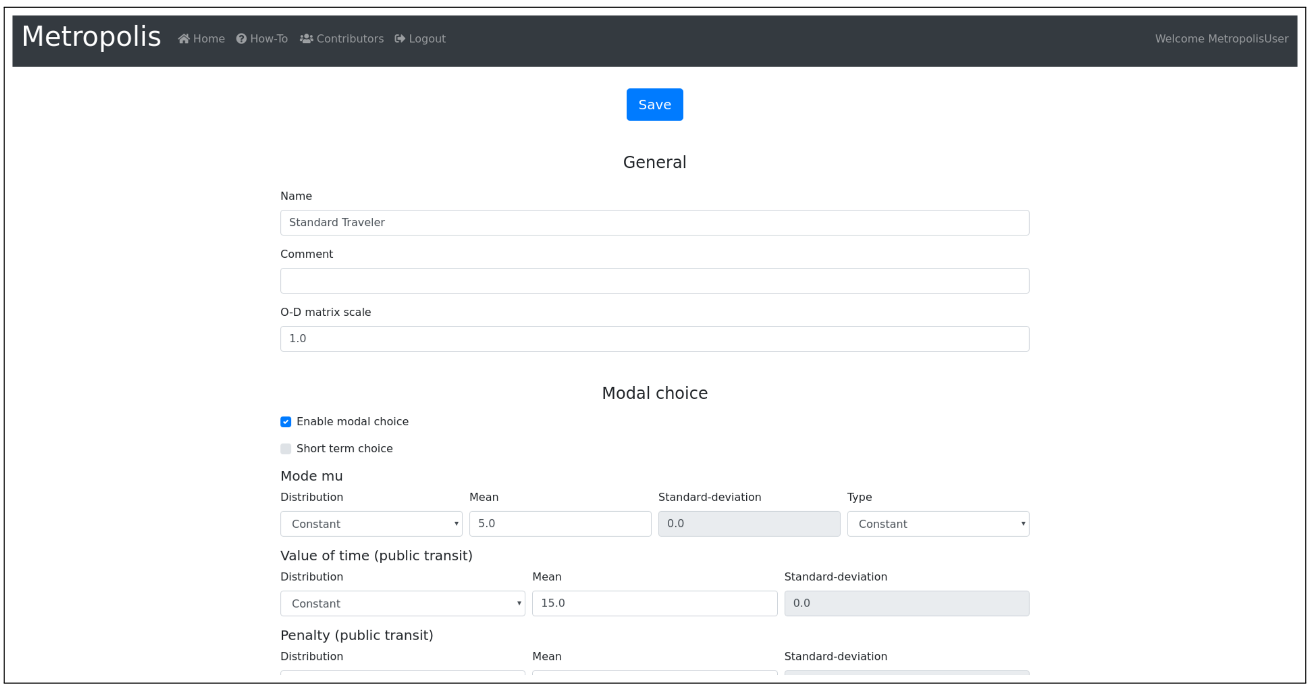


Figure 3.8: Page to edit the parameters of a traveler type

On this page, you can edit general information on the traveler type, parameters of the modal choice (if any), parameters of the departure time choice and parameters of the route choice. See Section 4.1.2 for a detailed description of the different parameters.

For many parameters, you can specify a distribution function from which the individual-specific value of the parameters are drawn for each traveler of this type. Three types of distribution function are available: uniform, normal and log-normal. You can also choose a constant value if you want that all travelers of this type have the same value for that parameter.

When you create a new user type, all parameters are set to recommended values.

For the departure time choice, make sure that the desired arrival time (or desired departure time) is within the recording period of the simulation (see Step 5), and preferably neither too early nor too late.

Step 4.b: Origin-destination matrix

The O-D matrix of a traveler type specifies the number of travelers of this type for each origin-destination pair.

This step is very similar to the public-transit system (Step 3.d). If there are not too many zones in the network, you can edit the O-D matrix directly (Figure 3.9). Otherwise, you have to import it from a tsv file with three columns (*origin*, *destination*, *population*). You need to use the id of the zones to specify the origin and destination of the O-D pair in the import file.

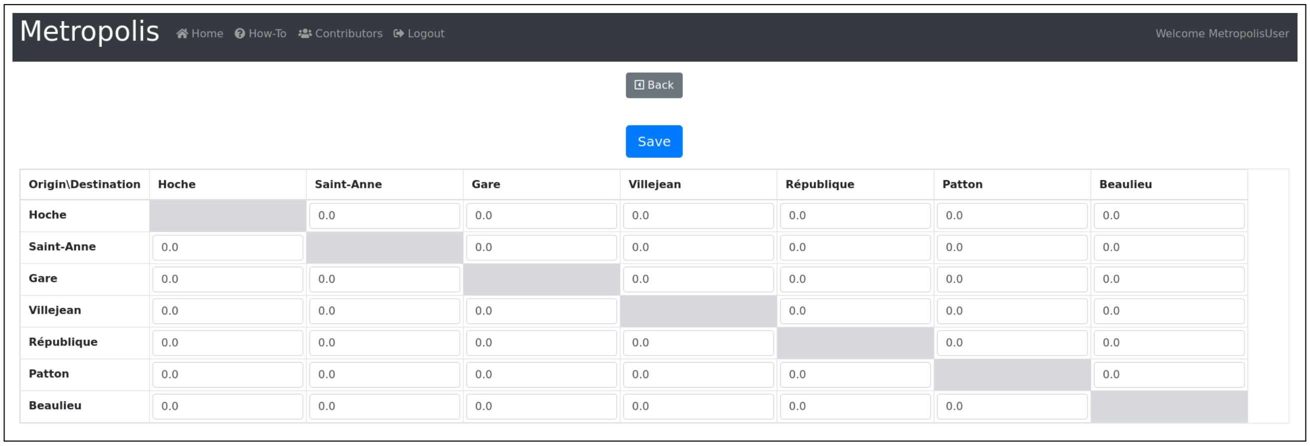


Figure 3.9: Page to edit the O-D matrix of a traveler type

Step 5: Add or modify the technical parameters of the simulation

Go to the bottom of the main page of the simulation to edit the technical parameters (Figure 3.10). See Section 4.1.3 for a detailed description of what is the meaning of each technical parameter. Do not forget to click on the save button after modifying the parameters.

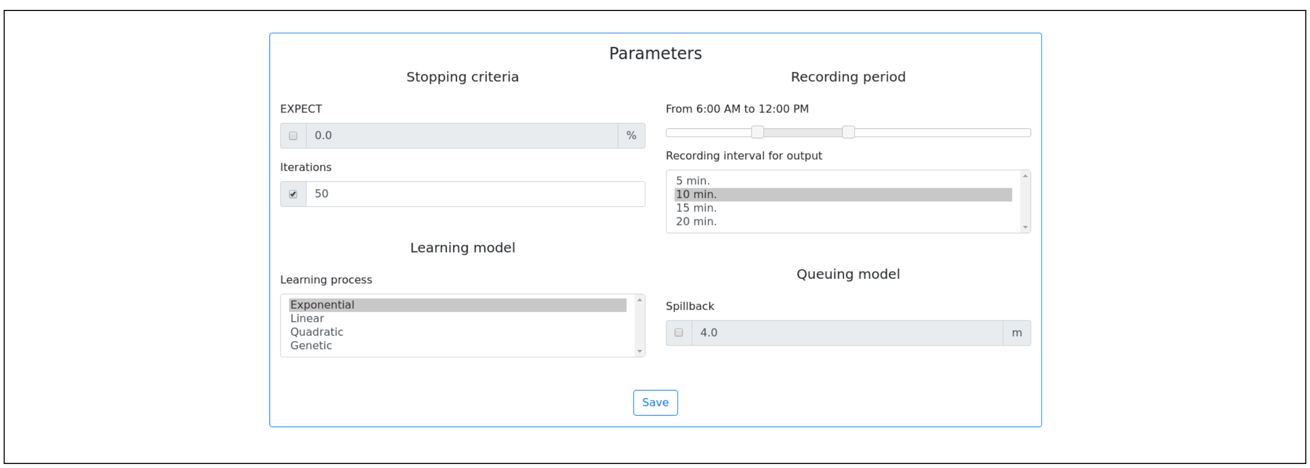
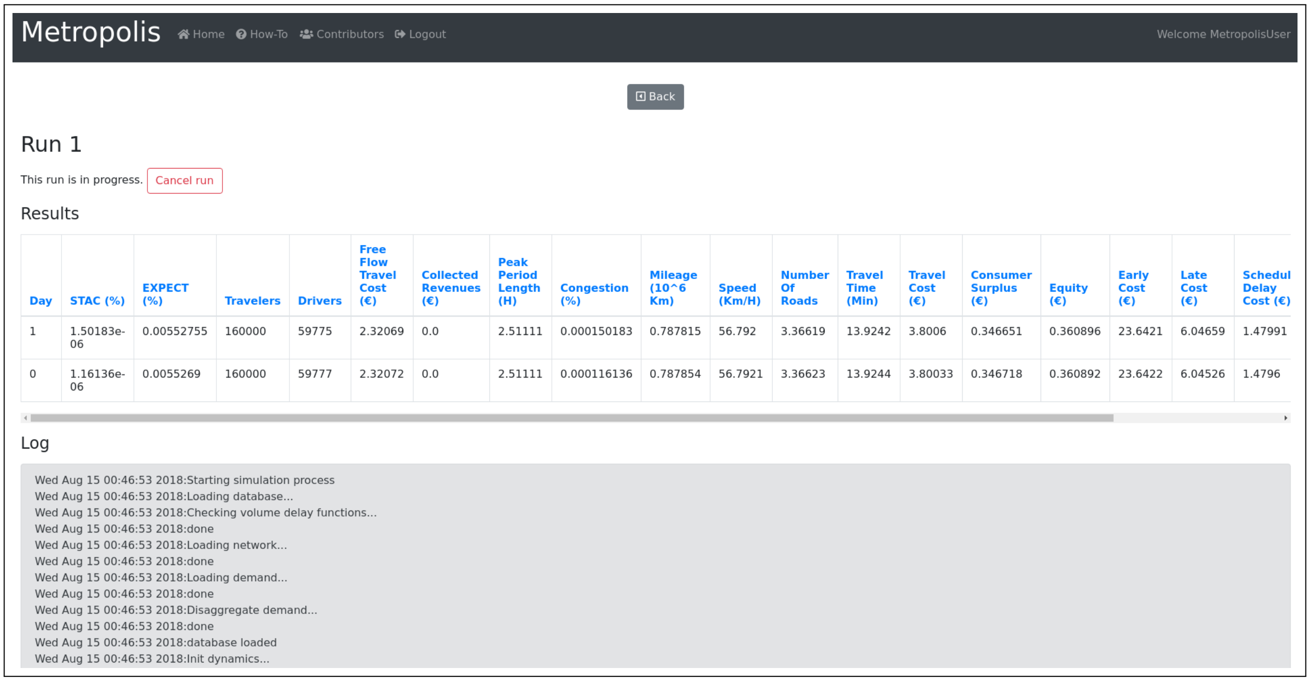


Figure 3.10: Page to edit the technical parameters of the simulation

Step 6: Run the simulation and look at the aggregated results

You can run the simulation from the main page, just use the *Run* button. You will be redirected to a page


Metropolis Home How-To Contributors Logout Welcome MetropolisUser

Back

Run 1

This run is in progress. Cancel run

Results

Day	STAC (%)	EXPECT (%)	Travelers	Drivers	Free Flow Travel Cost (€)	Collected Revenues (€)	Peak Period Length (H)	Congestion (%)	Mileage (10^6 Km)	Speed (Km/H)	Number Of Roads	Travel Time (Min)	Travel Cost (€)	Consumer Surplus (€)	Equity (€)	Early Cost (€)	Late Cost (€)	Schedul Delay Cost (€)
1	1.50183e-06	0.00552755	160000	59775	2.32069	0.0	2.51111	0.000150183	0.787815	56.792	3.36619	13.9242	3.8006	0.346651	0.360896	23.6421	6.04659	1.47991
0	1.16136e-06	0.0055269	160000	59777	2.32072	0.0	2.51111	0.000116136	0.787854	56.7921	3.36623	13.9244	3.80033	0.346718	0.360892	23.6422	6.04526	1.4796

Log

Wed Aug 15 00:46:53 2018:Starting simulation process
Wed Aug 15 00:46:53 2018:Loading database...
Wed Aug 15 00:46:53 2018:Checking volume delay functions...
Wed Aug 15 00:46:53 2018:done
Wed Aug 15 00:46:53 2018:Loading network...
Wed Aug 15 00:46:53 2018:done
Wed Aug 15 00:46:53 2018:Loading demand...
Wed Aug 15 00:46:53 2018:done
Wed Aug 15 00:46:53 2018:Disaggregate demand...
Wed Aug 15 00:46:53 2018:done
Wed Aug 15 00:46:53 2018:database loaded
Wed Aug 15 00:46:53 2018:Init dynamics...

Figure 3.11: Page where you can see data on the current run of the simulation

where you can see the current status, the aggregated results and the log of the run (Figure 3.11). When the simulation is running, this page will be refreshed regularly with new results and new log entries.

Before clicking on the *Run* button, check that you correctly specified the input of the simulation (all origin-destination pairs are connected by links, the public-transit system is complete, the desired arrival time is compatible with the recording period, . . .).

At each day (or iteration) in the run, aggregated results are computed. See Section 4.2 for a detailed description of what is the meaning of each column in the result table.

If you run a simulation multiple times, you can still look at the results of the previous runs by using the *Run history* button on the main page of the simulation.

Step 7: View the disaggregated results

Link-specific results

Once the simulation has successfully run, you can view the disaggregated results on the network using the *Results on network* button (on the page with the aggregated results or on the *Run history* page). You will be redirected to a page similar to the page to view the network, but new options are available (Figure 3.12). You can view incoming flows, outgoing flows and travel times for all the links. Use the slider if you wish to view disaggregated results at different point in times of the recording period.

You can toggle between historical and simulated values for the disaggregated results. Historical values are the values used by the travelers at the beginning of the last iteration to make their choices. Simulated values are the values effectively observed during the last iteration.

The *Download link-specific results* button (on the page with the aggregated results or on the *Run history* page) allows you to download a tsv file with the detailed values of incoming flows, outgoing flows and travel times, for each link and for each period.[9]

Note that only historical incoming flows are available for simulation with a very large network.

[9]See paragraph Recording period, section 4.1.3 to understand how the time of the periods is computed.

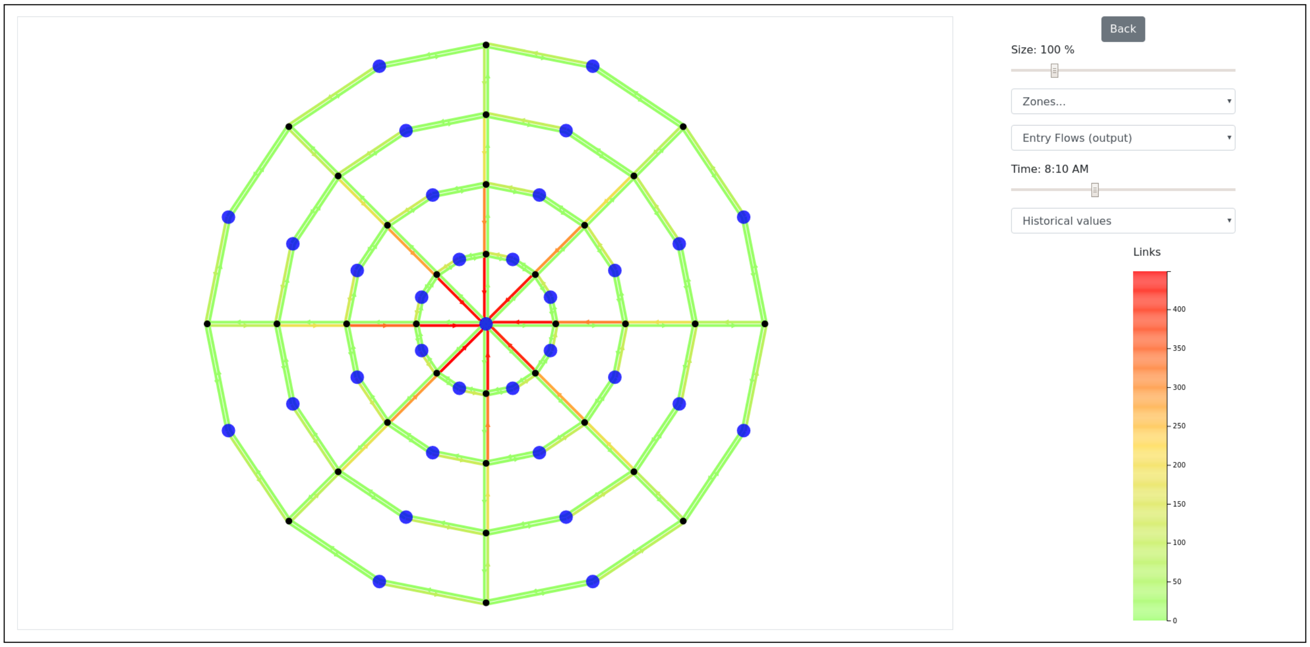


Figure 3.12: Page where you can view the disaggregated results of a simulation run

Traveler-specific results

You can also download disaggregated results using the *Download traveler-specific results* button. The results are stored in a tab-separated file with the following columns:

- origin: identifier of the origin zone;
- destination: identifier of the destination zone;
- segment: identifier of the traveler-type;
- driveCar: 1 if the traveler took the car, 0 otherwise;
- alphaTI: value of time spent in the car (in euros per hour);
- beta: penalty associated to early arrival (in euros per hour);
- gamma: penalty associated to late arrival (in euros per hour);
- alphaPT: value of time spent in other modes (in euros per hour);
- ptPenalty: penalty associated to other modes (in euros);
- td: departure time (in seconds after the start of the recording period);
- ta: arrival time (in seconds after the start of the recording period);
- ltstar: desired arrival time, minimum (in seconds after the start of the recording period);
- htstar: desired arrival time, maximum (in seconds after the start of the recording period);
- fee: fee paid by the traveler (in euros);
- surplus: consume surplus (in euros).

4 Input and Output Description

4.1 Input

4.1.1 Network

Zones and intersections
Zones and intersections share the same following attributes:

- id: must be unique, used to modify existing nodes with import files and to specify the origin and destination when importing links or O-D matrices;
- name: optional, only used to identify nodes on the network view;
- x: coordinate of the node on the x-axis, only used to draw the network;
- y: coordinate of the node on the y-axis, only used to draw the network.

Congestion functions
Congestion functions have the following attributes:

- id: must be unique, used to modify existing functions with import files and to specify the congestion function when importing links;
- name: optional;
- expression: expression used by METROPOLIS to compute the travel times of the links, see the functions of existing simulations on the website and see Step 3.b. of Section 3 (page 10) to learn the syntax to use.

Links
Links have the following attributes:[10]

- id: must be unique, used to modify existing links with import files;
- name: optional, only used to identify links of the network view;
- lanes: number of lanes for cars on the link;
- length: length of the link, in kilometers;
- speed: free flow speed of the link, in kilometers per hour;
- capacity: capacity per lane of the link, in vehicles per hour;
- function: congestion function which describes the congestion model of the link;
- origin: origin node (zone or intersection) of the link;
- destination: destination node (zone or intersection) of the link.

Public transit
Each pair in the public-transit system has the following attributes:

- origin: origin zone of the pair;
- destination: destination zone of the pair;
- travel time: public-transit travel time from origin to destination.

[10] Note that links represent only one-way streets. You need two opposite links to represent two-way streets.

4.1.2 Travelers

Parameters

Each traveler type has the following attributes:

- General attributes:
  - name: optional, use names to easily identify traveler types in the list;
  - comment: optional, use comments to describe the particularities of a traveler type;
  - O-D matrix scale: for each O-D pair in the O-D matrix of the traveler type, the number of travelers is multiplied by this scale (default is 1).
- Attributes related to the modal choice (see Section 2.1.1):
  - modal choice: enable or disable modal choice between car and public transit;
  - short term choice: enable or disable short term choice for the computation of the generalized cost associated to the car;
  - mode mu: heterogeneity of the modal choice (parameter of the logit model);
  - type of mode mu: specify how the mu is computed (constant, adaptive to alpha or adaptive to free-flow travel cost);
  - value of time (VOT_{PT}): value of time spent in public transit, in euros per hour;
  - penalty (P_{PT}): penalty associated to public transit, in euros.
- Attributes related to the departure time choice (see Section 2.1.2):
  - commute type: choose to simulate morning commute (target is arrival time) or evening commute (target is departure time);
  - departure mu (μ_d): heterogeneity of the departure time choice (parameter of the logit model);
  - type of departure mu: specify how the mu is computed (constant, adaptive to alpha or adaptive to free-flow travel cost);
  - desired arrival/departure time (t^*): desired arrival time at destination or departure time at origin, in minutes after midnight (0 is midnight, 360 is 6:00 AM, etc.);
  - on-time period length (δ): length of the flexible time period without penalty, centered around t^*, in minutes;[11]
  - value of time (α): value of time spent in the car, in euros per hour;
  - early penalty (β): penalty associated to early arrival or early departure, in euros per hour;
  - late penalty (γ): penalty associated to late arrival or late departure, in euros per hour.
- Attributes related to the route choice (see Section 2.1.3):
  - type of route choice: toggle between deterministic and stochastic route choice;
  - route mu (μ_r): heterogeneity of the route choice (parameter of the logit model);
  - type of route mu: specify how the mu is computed (constant, adaptive to alpha or adaptive to free-flow travel cost);
  - travelers observe congestion on the next link: if enabled, at each intersections, travelers observe travel time on the downstream links and can choose to modify their route choice.

[11] The on-time period is $[t^* - \delta/2; t^* + \delta/2]$.

O-D matrix
Each pair of the O-D matrix of a traveler type has the following attributes:

- origin: origin zone of the pair;
- destination: destination zone of the pair;
- population: number of travelers commuting from origin to destination.

4.1.3 Technical Parameters

Stopping criteria
Metropolis simulates days (or iterations) until one stopping criterion is met. There are two stopping criteria. You can enable only one, or both.

- EXPECT: the simulation stops when the value of EXPECT is below the specified threshold (Section 4.2 shows how the value of EXPECT is computed);
- iterations: the simulation stops if it reaches the specified maximum number of iterations.

Learning model
See Section 2.3 to view the differences between the four types of learning process (exponential, linear, quadratic and genetic).[12]

Recording period
The travelers can begin their trip at the start of the recording period and they must arrive at destination before the end of the recording period. It is recommended to choose a recording period which lasts no more than 6 hours.

The recording interval is the interval of time at which the travel times and flows of the links are stored. For example, if the starting time is 6:00 AM, the ending time is 12:00 PM and the interval is 10 minutes, the simulation will record the disaggregated results at 36 points in time: 6:00 AM, 6:10 AM, 6:20 AM, . . . , 11:40 AM and 11:50 AM. A shorter recording interval implies a longer time to compute the results of the simulation.

Queuing model
If the horizontal queuing model (spillback) is enabled, congestion on a link might spread to the upstream links. If you enable horizontal queuing, you must specify the length of a link used by a single vehicle when the link is congested, in meters.

4.2 Output

Aggregated results
For each iteration, Metropolis computes the following results:

- STAC: indicator based on the relative variations of travel times from one iteration to the next on all the links and for the entire period of the simulation, in percentage;
- EXPECT: criterion used for the stopping rule — the value of EXPECT, in percentage, is

$$\text{EXPECT} = \frac{|\widehat{tt} - tt^*|}{\widehat{tt}}$$

  where $\widehat{tt}$ is the total travel time expected at the beginning of the day and tt^* is the effective total travel time during the day;

[12]The value of λ in the exponential model is set to 0.1

- travelers: total number of travelers in the simulation;
- drivers: total number of travelers who commute by car;
- free flow travel cost: average free flow travel cost for drivers, in euros — it is equal to the value of time α multiplied by the free flow travel time on the path selected;
- collected revenues: sum of all toll revenues, in euros.
- peak period length: length of the peak period, in hours — the peak period starts when 10% of the drivers have reached their destination, and ends when 90% of the drivers have reached their destination;
- congestion: average congestion on the links selected by at least one driver, in percentage — the value of congestion is

$$\text{congestion} = \frac{tt^* - tt^0}{tt^0}$$

  where tt^0 is the free flow travel time and tt^* is the effective travel time;
- mileage: total mileage of all drivers, in millions of kilometers;
- speed: average speed of drivers, in kilometers per hour;
- number of roads: average number of road sections per driver's trip;
- travel time: average travel time for drivers, in minutes;
- travel cost: average travel cost for drivers, including tolls, in euros;
- consumer surplus: average accessibility of travelers, in euros — it corresponds to the variable A in Section 2.1.2;
- equity: standard deviation of the consumer surplus, in euros;
- early cost: average schedule delay cost of drivers who arrive early, in euros;
- late cost: average schedule delay cost of drivers who arrive late, in euros;
- schedule delay cost: average schedule delay cost for drivers, in euros — it is equal to $C_2(t) + C_3(t)$ in Section 2.1.2;
- early ratio: ratio of drivers who arrive early, in percentage;
- on-time ratio: ratio of drivers who arrive on time, in percentage;
- late ratio: ratio of drivers who arrive late, in percentage.

Disaggregated results

For each link of the network, METROPOLIS computes the following results:

- entry flows: number of travelers entering the link over the given interval;
- exit flows: number of travelers leaving the link over the given interval;
- travel times: average travel time on the link over the given interval, take into account congestion.

You can choose between historical and simulated values (see Section 2.3).

Glossary

congestion function Function measuring the travel time of a link given the flow and volume of the link. 6–10, 14, 15

intersection Point of the network used to connect links between each other. 4, 6–10, 13, 15, 16

link Edge of the network representing a one-way road. 1, 4–10, 13–18

network Abstract representation of the city or region where the travelers commute. 1, 4–10, 12–14, 18

node Point of the network, nodes can represent either a zone or an intersection. 1, 8–10, 14, 15

O-D matrix Matrix which specifies the number of trips going from each origin to each destination. 1, 10, 12, 15, 16

run Each time METROPOLIS does a simulation, a run is created. One simulation can have multiple runs and can be modified between two runs. 6, 7, 13, 14

segment See traveler type. 2, 10

simulation Instance of METROPOLIS characterized by a network and a demand (the travelers). 1–3, 6–14, 16, 17

technical parameter Parameter of a simulation not related to the network or the travelers. 6, 7, 12

traveler Individual characterized by preferences and by an origin and destination zone. 1–7, 10–13, 15–18

traveler type Group of travelers characterized by specific preferences and O-D matrix. 10–12, 15, 16

value of time Opportunity cost of time spent while traveling. 2, 3, 15–17

zone Zone of departure or arrival of the travelers, a zone is represented by a single node on the network. 6–10, 12, 13, 15, 16

Bibliography

de Palma, A., F. Marchal, and Y. Nesterov (1997). Metropolis: Modular system for dynamic traffic simulation. *Transportation Research Record: Journal of the Transportation Research Board* (1607), 178–184.