Complex system modeling of population growth and public transportation in canton of Genève

Technical Report

Atta Samiullah
Department of Computer Science (DIKU)
University of Copenhagen
rwq249@alumni.ku.dk

Elisabetta Bosello
Department of Mathematical Sciences
University of Milan
elisabetta.bosello@studenti.unimi.it

April 17, 2024

Abstract

Urban expansion has an important impact on the transportation system that have to be improved to face an increasing demand. We develop an agent-based model, following the DPSIR framework, to represent the consequence of the population growth on the public transportation in the canton of Geneve. Doing systematic simulations in GAMA platform, we observe the role of certain parameters such as the speed and the capacity of the trams. Results show that to resolve traffic congestion policies should look towards these two aspects. Our research brings an analysis for how aspects such as overcrowdedness can affect means of transport.

Keywords: Agent-based model, urban expansion ,GAMA, DPSIR, population growth, tram

Introduction

This project is part of BMST2024 (Bachelor in Mathematics Student Task) with the University of Geneva as part of the 4EU+ alliance. The goal is to simulate the tram line in Geneva and to study its relationship with population growth and urban expansion. We present our agent-based model developed in "GAMA Platform" (2024) for analysing different parameters related to the tram line 18 in Geneva. The model is based on the DPSIR framework by Smeets and Weterings (1999) and preprocessed data is from TPG/ "SITG" (n.d.).

The purpose of our project is to analyse the public transportation in the canton of Geneva (specifically the tram line 18). We aim to find how adjusting different parameters such as the capacity, speed and number of trams, can effect the commuters experience in terms of waiting time as well as travel time (quality of life).

Use case

We concentrate on the public transportation in the canton of Geneva. We analyze the tram lines giving more attention to the line 18 that connects the city center to Meyrin CERN, that is near to the international airport of Geneva: it is a crucial line into the transport system in Geneva.

Figure 1 illustrates the area joined by line 18 with its stops.

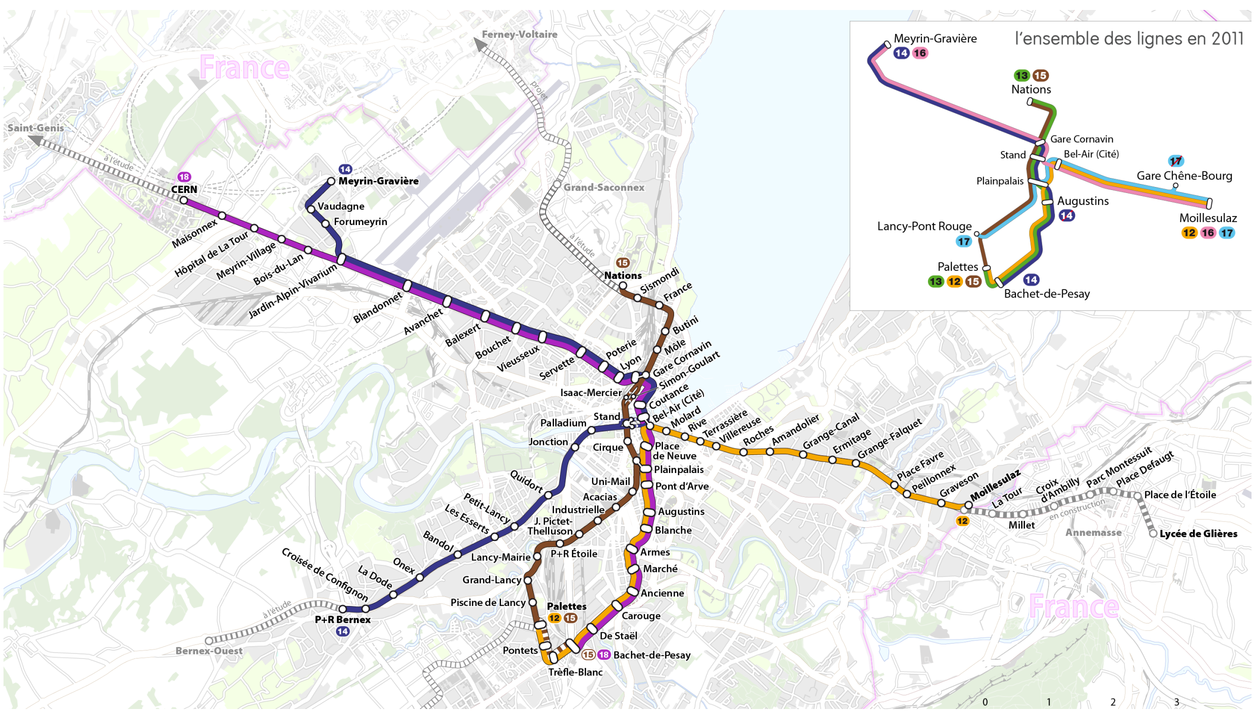

Figure 1: Geneva public transportation system "map", n.d.

The research question is about how human activity has an impact on the land. We do several simulations changing some aspects: the speed, the capacity and the number of the trams, the number of people taking means of transport. We measure the average time spent waiting the tram and travelling on it, then we study its increase or decrease linked to the variables mentioned before.

DPSIR graph

The DPSIR framework consists of different components that can briefly be explained as following. The driving forces lead to environmental pressures. The state of the environment will be modified, having impact on the urban land. Social responses will be given in order to answer to these changes.

We examine how the population growth and the expansion of the city, that can be induced by business growth, has an impact on the mobility. The public transportation offer needs to be enhanced in order to face an increasing demand. We study how, changing the capacity and the speed of trams, traffic congestion can be solved. To sum up, we analyze how the mobility can be improved increasing the serving frequency and establishing new line. We focus on the tram line 18, an eco-friendly means of transport TPG, n.d. Reducing ticket costs, people will be more encouraged to use this means: carbon emissions will be reduced. In addition, decreasing waiting time for trams, life of citizens can improve: they save their time.

Figure 2 shows the DPSIR graph

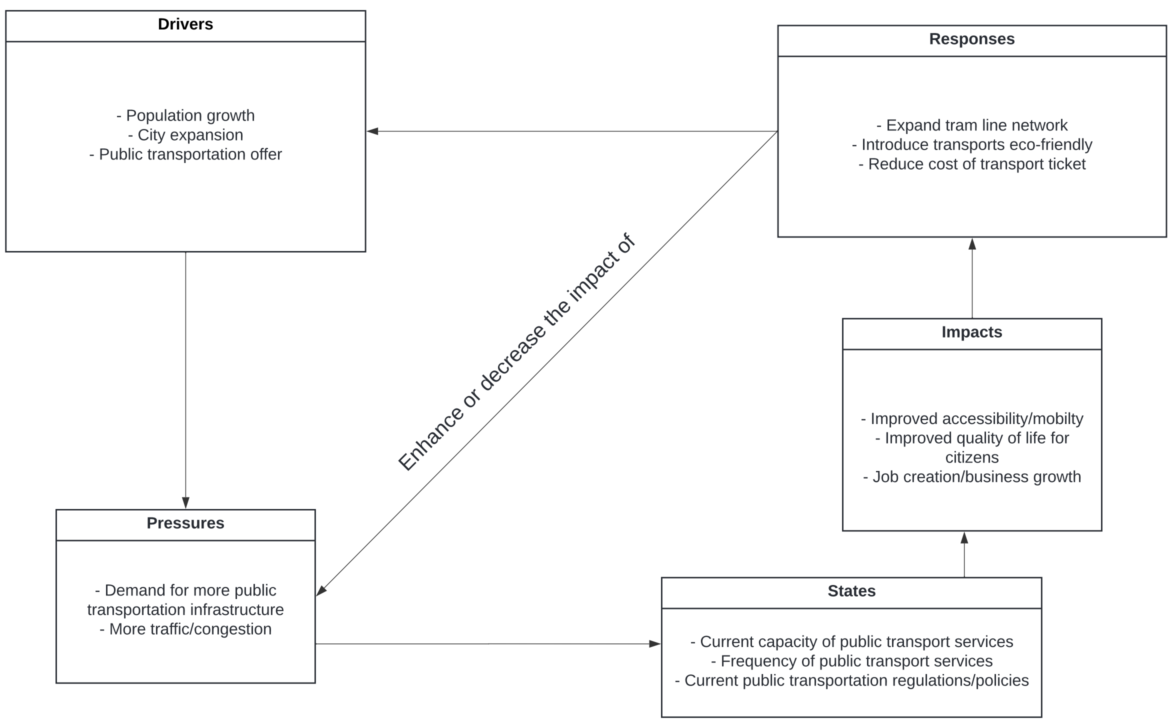

Figure 2: DPSIR graph

Agent-based model

We implement our agent-based model within the GAMA platform. GAMA platform is a modeling and simulation environment which is capable of modeling complex and dynamic systems such as public transportation systems.
We take data about the area of Geneva from the SITG (*Système d'Information du Territoire à Genève*) and we develop vector data in the form of shapefiles. The data are about tram lines (spatial length) and tram stops (position, stop name, direction); but we select only the data associated to the tram 18. Tram lines, tram stops and commuters are defined as agent species with a set of actions. Each species has attributes, behavior, aspects: stop is displayed as black square, commuter as a blue circle, tram as a circle, tram lines are orange. The tram is visualized yellow if it is crowded, if not red.

Trams move along the line *Grand-Lancy, Palettes* - CERN and reverse at the final stop. At the beginning of the simulation at each commuter is associated a random start and destination position among the different stops. They can get on the tram if its current capacity is strictly less than its maximum capacity, otherwise they wait until they find a not-crowded tram. The duration of a simulation step is 10 seconds.

In each simulation we change the number, the speed, the capacity of the trams and the number of commuters, then we measure the time spent waiting the tram and the travel time. Through a Python code we calculate the average of these two types of time. To do this, these sentences are displayed at the GAMA terminal for each commuter: "*Passenger waited . . . for the tram*", "*passenger spent . . . unit time travelling on tram to get to destination*". We copy these lines into a file *txt*; then a Python code calculates the average time doing the division between the total time and the number of passengers. Later we draw some graphs on MATLAB illustrating how the simulations change modifying variables. The GAMA model can be found on this github link:
github.com/AS0013/BMST2024_Geneva

Figure 3 and 4 illustrate the view produced by a simulation in GAMA.

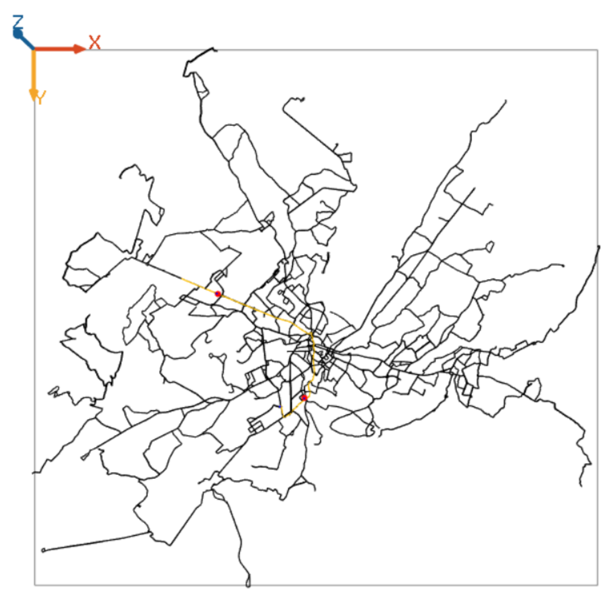

Figure 3: View of a GAMA simulation

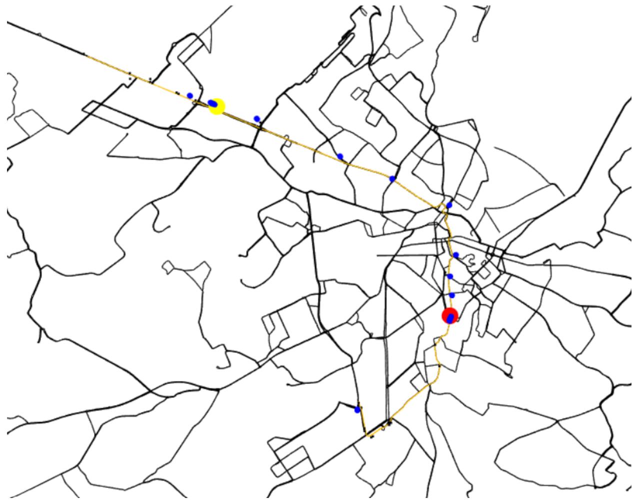

Figure 4: Picture of a GAMA simulation

Agents:

Table 1: Explanation of different agents

Agent	Explanation
Passenger	The passenger agent represents the commuter that uses the tram to travel from one stop to another stop.
Tram	The tram agent represents the tram that the passengers use to travel on.

Passenger:

A passenger represents a commuter that uses the tram. A passenger in our model works such that it randomly choses a start position and stop (destination) and can then decide if it wants to board a tram based on if the tram is going in the direction of the destination. The logic of the passenger can be seen by the pseudo-code below:

```
stop star_stop
stop destination_stop
bool reached_destination <- false
bool is_on_tram <- false
float boarding_time
float start_time
stop direction
tram_18 tram

reflex move{  // logic for boarding and alighting tram
    if not reached_destination{
        if not is_on_tram{
            loop Tram over: tram_18{
            -> check if the tram is moving towards passenger destination
            -> if tram is moving towards passengers desination,
                assign that tram to the passenger and passenger boards the tram.
            -> increment the tram number of passengers by 1
            }

        }
```

```
        else{ // is on tram

             -> check if tram has reached destination
             -> if reached, then decrement the trams number of passenger by 1
             -> passenger alights the tram (do die)

             -> if not reached yet, keep moving with the tram
        }
    }
}
```

Tram:

The tram is the public means of transportation that the passenger uses to get to the desired destination, and the tram 18 in Geneva is simulated. The tram moves back and forth between Palletes and CERN. The logic of the tram can be seen by the pseudo-code below:

```
point target //target is the stop the tram is going towards
int target_index // the index location of the stop in the list of stops
int passenger_capacity <- Tram_capacity // Tram_capacity is at initialisation
int current_passengers <- 0
stop Terminal_1
stop Terminal_2
stop Direction
bool moving_towards_cern

reflex update_target{
    if (moving towards cern){
        Direction <- Terminal_2
        target_index <- target_index + 1  // moving to the next stop
        if  (reached CERN){
            moving_towards_cern <- false // now moving to palletes
            decrement target index by 1 // going reverse order in the list of stops
        }
    }
    else { // moving towards palletes
        Direction <- Terminal_1
        target_index <- target_index -1  //  move one stop back towords palletes
        if (reached palletes)
            moving_towads_cern <- true // now move back towards cern
            increment target index by 1

    }

    -> move towards next stop
}

reflex full_capacity {
    if number of passengers aboard is equal to tram capacity
    -> change color from yellow to red
}
```

Experiments and Results

We perform several experiments in which we fix some parameters and vary others. Below is an explanation of the parameters.

Table 2: Explanation of parameters experimented upon

Parameters	**Explanation**
Number of trams	The number of trams in use during simulation
Speed	The speed of the tram(s)
Capacity of tram	Number of passengers a tram can facilitate
Number of passengers	Total number of commuters for a given simulation

You can see below the summary of the different experiments we perform and which parameters are fixed aswell as which ones are varied.

Parameters	Experiment	Experiment 2	Experiment 3	Experiment 4
Speed	Fixed	Varied	Fixed	Fixed
Capacity	Fixed	Fixed	Varied	Fixed
Number of passengers	Varied	Fixed	Fixed	Varied

Table 3: Overview of the different experiments and their parameters

Experiment 1

Fixing the speed of two trams and their capacity, we analyze how the waiting time for a stop changes increasing the number of commuters. Until the number of passengers is less than the capacity, even if the number of people increases the time measured remains the same because there are enough seats in the trams. When the number of people is more than the capacity we observe that, increasing the number of commuters, the waiting time increases. This phenomenon is linked to the insufficient public transportation system. We decide to simulate another state with the same variables but with only one tram. We observe that, similarly to the previous situation, the waiting time increases when the number of passengers, more than the capacity, increases. Making a comparation between the two situations, it is evident that the waiting time measured with one tram is more than the respective with two trams. This is linked to the following fact. If we have two stops (A, B) and a person wants to go from B to A, if there is only one tram (that goes from A to B) that person has to wait that the tram goes to the final stop and turn back; while if there are two trams, one tram goes from B to A, so the time is less.

Figure 5 shows this situation with the parameters from Table 3. The green line represents when the number of people is equal to the capacity.

Table 4: parameters

parameter	**fixed/varies**
Speed	Fixed
Capacity	Fixed
Number of commuters	Varies

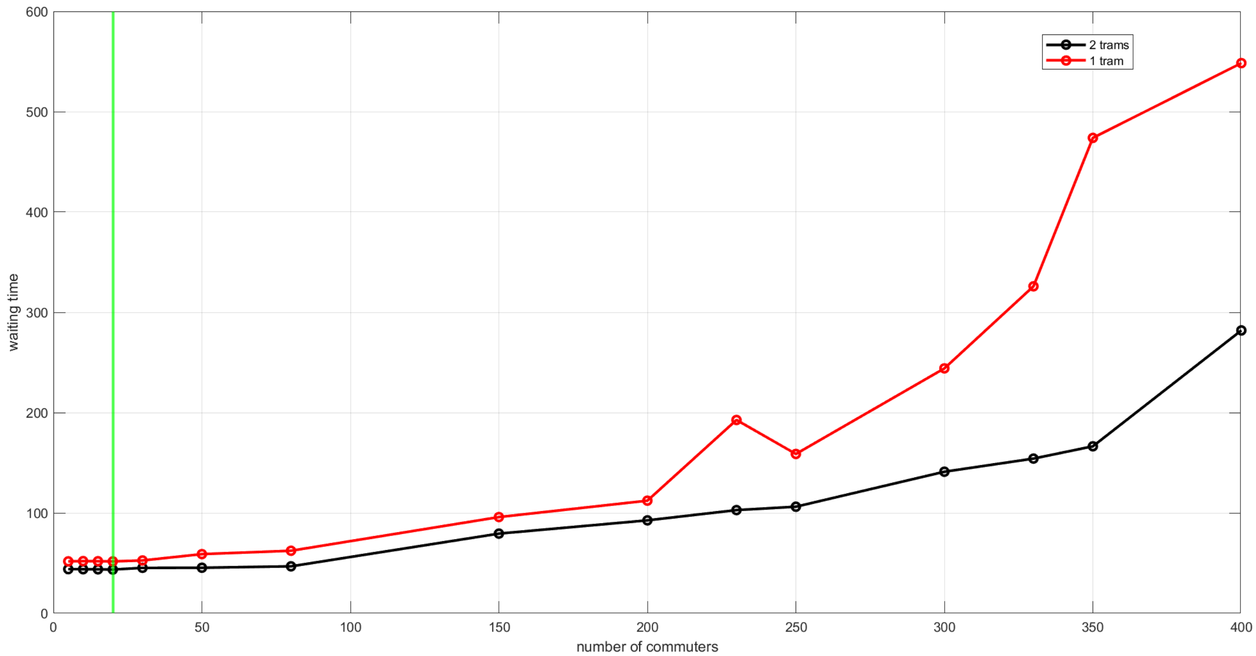

Figure 5: Simulation where the number of commuters changes

Experiment 2

Fixing the capacity of two trams and the number of commuters we analyze how time spent waiting a tram at a stop and the travel length change. Increasing the speed of the trams, both the time decrease. To be specific, there are some values between the waiting time that are slightly larger than the previous: this is related to the random start position of passengers. We find also through the values of the travel time some data bigger than the previous: this is again the consequence of the random start and destination position. When we simulate only one tram, we deduce the same observations related to the increasing of the time. As in the previous situation the waiting time with one tram is a bit more than the one measured with two trams. This doesn't happen for one value of the speed: this is linked to the random start position. Otherwise, time travel is quite the same, as the intuition suggests; different values are, once again, related to the random component of the model.

Figure 6 is graph of the situation with the given parameters in table 4, and shows how waiting time is impacted (Circles evidence "misplaced" values).

Table 5: parameters

parameter	**fixed/varies**
Number of commuters	Fixed
Capacity	Fixed
Speed	Varies

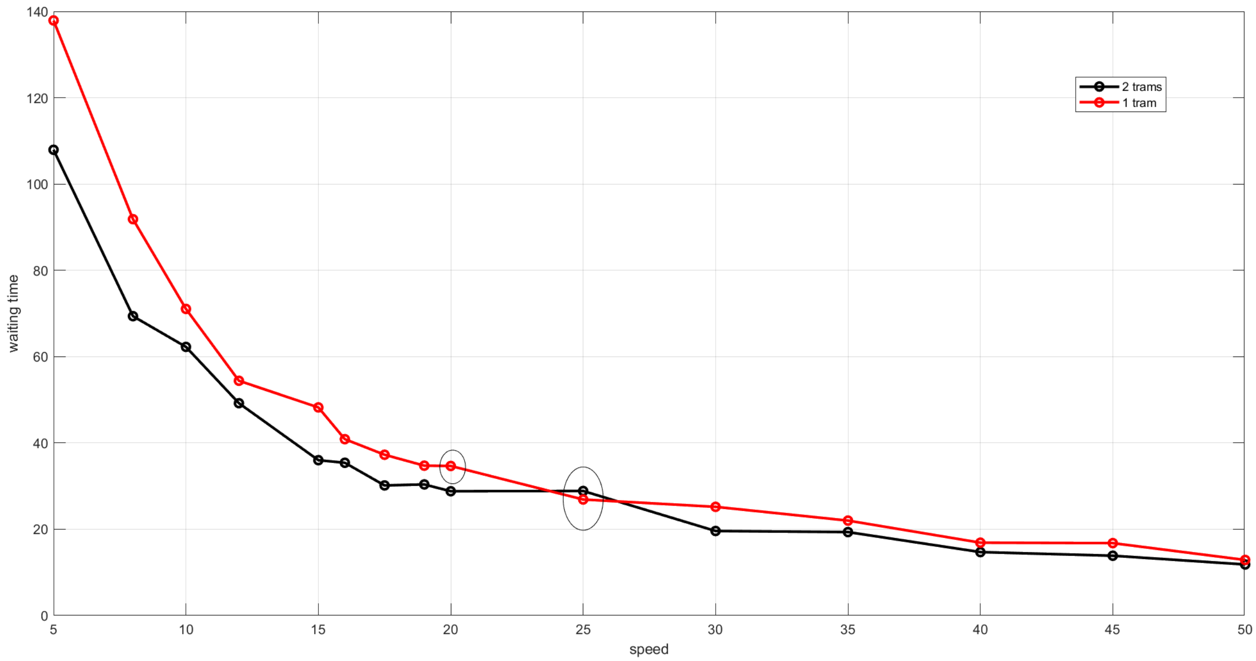

Figure 6: Simulation where speed changes - waiting time

Figure 7 similarily is a graph of the situation with the given parameters in table 4, but shows how travel time is impacted instead.(Circles evidence “misplaced” value).

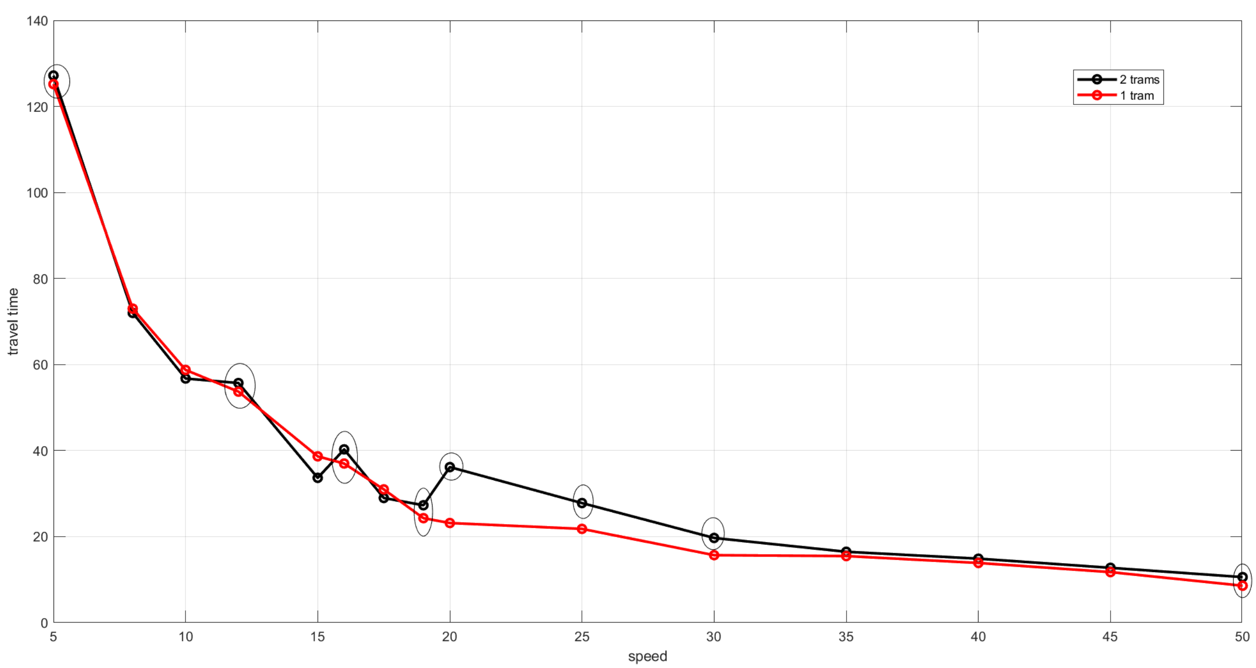

Figure 7: Simulation where speed changes - travel time

Experiment 3

Fixing the speed of two trams and the number of passengers, we see how the capacity influences the time spent waiting for a stop. Until the number of people is more than the capacity, if the capacity increases the waiting time decreases: in these cases it happens that some commuters wait for a not-crowded tram because the transportation offer is deficient. Otherwise, if the number of people is less than the capacity the waiting time remains constant even if the capacity increases: in all these situations the transportation offer is enough. We observe a similar situation having only one tram. As before, waiting time with one tram is more than the respective with two trams.

Figure 8 displays these observations with the parameters from table 5. The green line represents when the number of people is equal to the capacity.

Table 6: parameters

parameter	fixed/varies
Number of commuters	Fixed
Speed	Fixed
Capacity	Varies

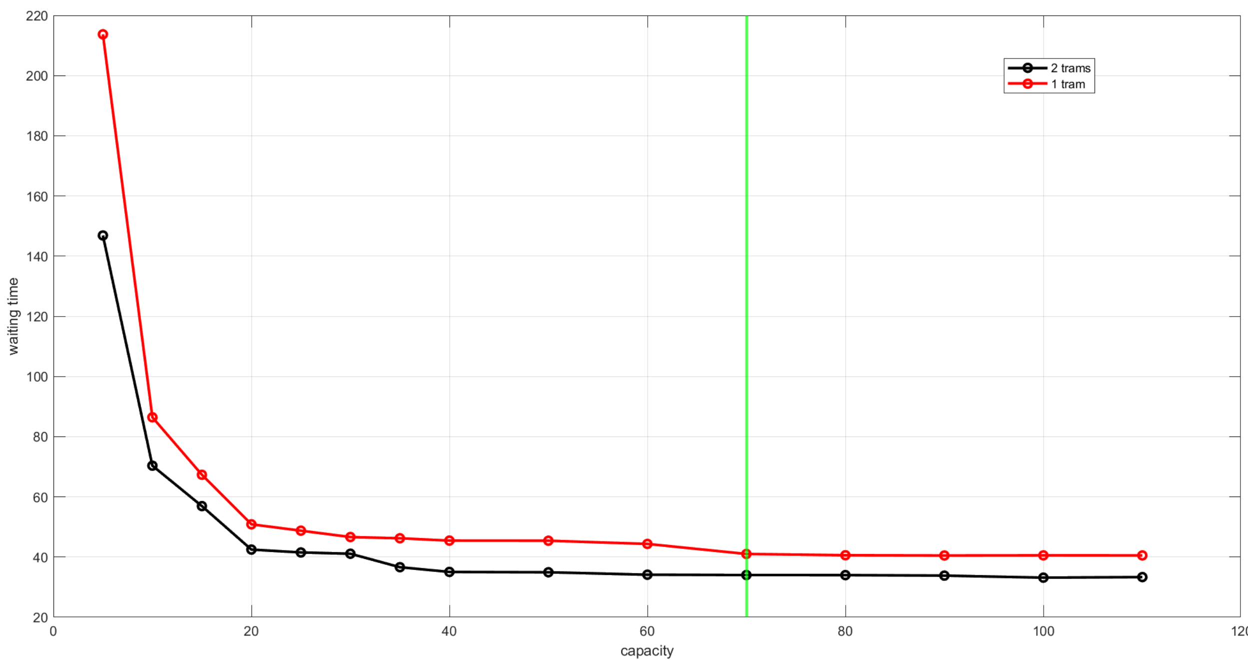

Figure 8: Simulation where capacity changes

Experiment 4

If the simulation stops when the trams arrive at the final stop it can happen that a person does not arrive at his destination; so in each simulation the number of passengers that gets on or off the trams is different. Consequently, we cannot analyze situations when the number of people is fixed: even if we put in the GAMA code the number of passengers, we cannot control that number. We decide to analyze only the situation in which number of people is a variable and the trams stop at the final stop.

We cannot compare, as we did before, the situation with one or two trams because, even if the number of people in the GAMA code is the same, the number of passengers of whom we measure the time is different. In both situations until the number of passengers is less than the capacity, even if the number of people increases, the waiting time remains the same. When the number of commuters is more than the capacity we observe that, increasing that number, the time measured increases. It happens that some values are less than the previous. This is linked to the fact that we manually stop the simulation when the trams arrive at the final stop; so the duration of the simulations can be different for some seconds.

With the given parameters from table 6, Figure 9 shows the situation with two trams, Figure 10 with one tram.The green line represents when the number of people is equal to the capacity (Circles evidence “misplaced” values).

Table 7: parameters

parameter	**fixed/varies**
Capacity	Fixed
Speed	Fixed
Number of commuters	Varies

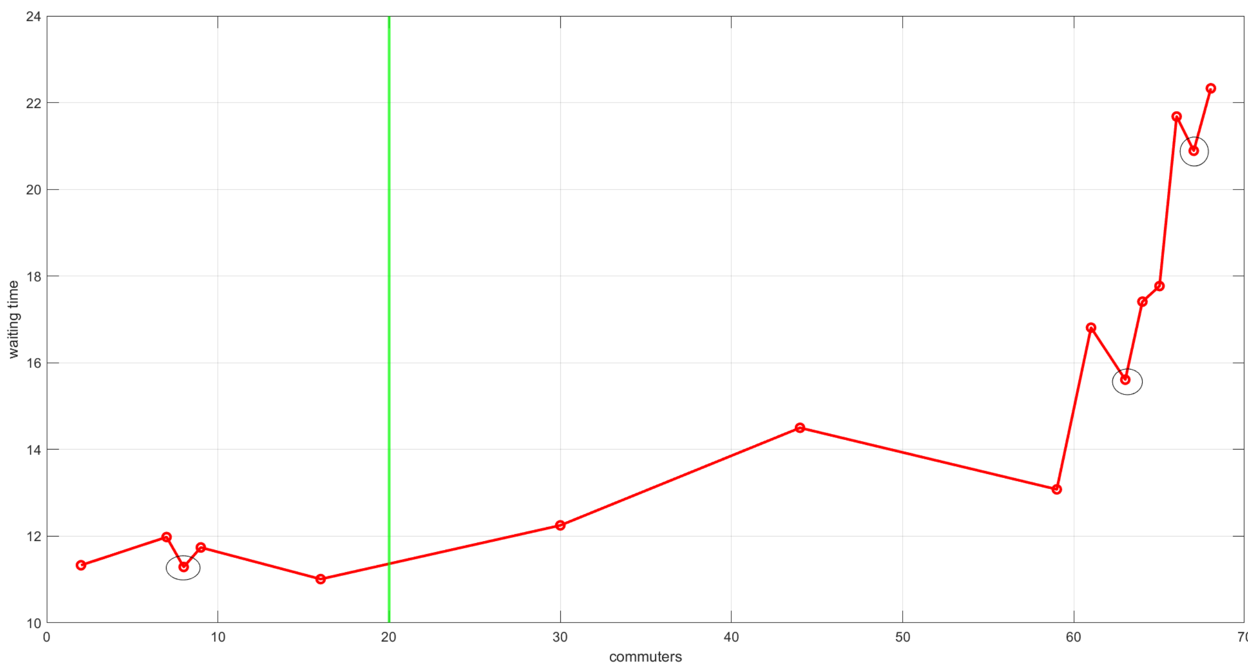

Figure 9: Simulation where the number of commuters changes - two trams stop at the final stop

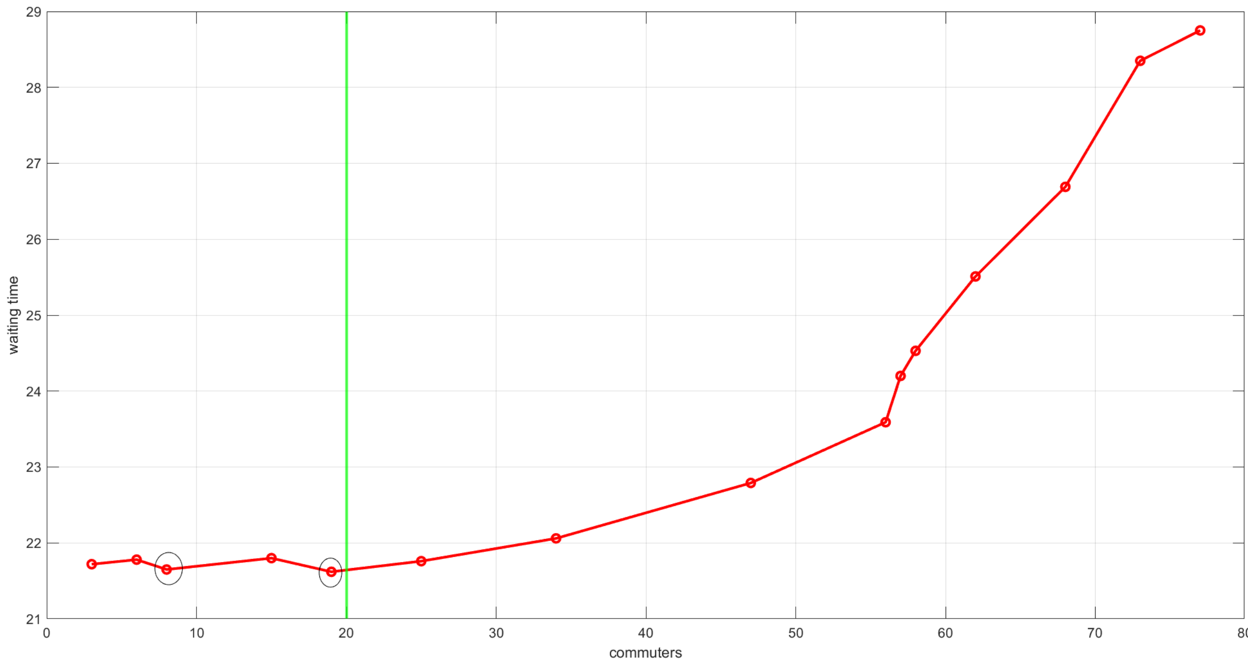

Figure 10: Simulation where the number of commuters changes - one tram stops at the final stop

Regarding the validation, our results agree with those presented in the research paper by Chambers et al. (2023). We both see that journey length increases when public transportation offer is decreased, which is correlated with more overcrowded trams.

Discussion: Shortcomings and possible improvements

Observations on the graphs are focused on the trend of the data, not on their specific value. We simulate a situation with number of people and trams much lower than reality. We can improve our model increasing the number of trams along the axis studied: the GAMA simulation will show more realistic events. In this way, we will see more precisely how the time measured changes, modifying the variable considered.

We evidence a problem in the last part: stopping manually the time, we cannot be too accurate for the duration of the simulation. To resolve this, we can write a code when the tram stops as soon as it reaches the final stop.

The random component of our model linked to the start and destination position affects the data. To reduce this effect, we can do more simulations for each situation and calculate the average of the average time.

Due to certain system limitations aswell as GAMA's own sensitivity to how code is written. It was quite difficult to stress test the model and see how far we could take it in terms of how much we could increase the values of the different parameters, as GAMA would become incredibly slow and fill the memory very quickly.

Conclusion

We concentrated on the urban expansion and on its impact on the transportation system. We focused on the crucial line 18 in the public transportation system of Geneve.

We provided an agent-base model following the DPSIR framework. We analyzed how the transportation system can be enhanced in order to face an increasing demand induced by the population growth and the expansion of the city. We studied the time people spent waiting for a tram and travelling on it. We did systematic simulations in GAMA searching how the parameters (speed, capacity, number of passengers) has an impact on the mobility.

Through the charts about the waiting and travel time, we noticed some important aspects that affect the public transportation system. We observed that increasing the speed and the capacity of the trams the transportation system is improved; obviously, there is a maximum value for them. We also saw how the number of people has an impact on the transportation system: if the number of passengers is high, the system has to be enhanced.

This research can be useful to take policies to improve public transportation system to face an increasing demand. Looking how much waiting time and travel time changes, modifying speed or capacity, more accurate policies can be taken.

APPENDIX: PYTHON CODE FOR PROCESSING DATA

Listing 1: handling data in python

```
import re

average_waiting_time = 0
average_travel_time = 0

total_waiting_time = 0
total_travel_time = 0


#pattern for "Passenger waited 15000.0 for the tram"  and get the time
pattern_1 = re.compile(r"Passenger␣waited␣(\d+\.\d+)␣for␣the␣tram")
pattern_1_count = 0
# pattern for "passenger spent 185400.0 unit time travelling on tram to get to
    destination" and get the time
pattern_2 = re.compile(r"passenger␣spent␣(\d+\.\d+)␣unit␣time␣travelling␣on␣tram␣
    to␣get␣to␣destination")
pattern_2_count = 0
# read the file and get the waiting time and travel time
with open ('waiting_time7.txt') as file:
    for line in file:
        match = pattern_1.search(line)
        if match:
            total_waiting_time += float(match.group(1))
            pattern_1_count += 1
        match = pattern_2.search(line)
        if match:
            total_travel_time += float(match.group(1))
            pattern_2_count += 1

# calculate the average waiting time and travel time
average_waiting_time = total_waiting_time / 100
average_travel_time = total_travel_time / 100

print("Average␣waiting␣time:␣", average_waiting_time)
print("Average␣travel␣time:␣", average_travel_time)
print("pattern␣1␣count:␣", pattern_1_count)
print("pattern␣2␣count:␣", pattern_2_count)
```

APPENDIX: MATLAB CODE FOR PROCESSING DATA

Listing 2: handling data in MATLAB

```
clearvars, close all

% File used to calculate average waiting and travel time

% Initialize variables
average_waiting_time = 0;
average_travel_time = 0;
total_waiting_time = 0;
total_travel_time = 0;

% Pattern for "Passenger waited 15000.0 for the tram"
pattern_1 = 'Passenger␣waited␣(\d+\.\d+)␣for␣the␣tram';
% Pattern for "passenger spent 185400.0 unit time travelling on tram to get to
    destination"
pattern_2 = 'passenger␣spent␣(\d+\.\d+)␣unit␣time␣travelling␣on␣tram␣to␣get␣to␣
    destination';

% Open the file and read line by line
fileID = fopen('waiting_time1.txt', 'r');
if fileID == -1
    error('Error␣opening␣file');
end

% To keep track of the number of matches
count_waiting = 0;
count_travel=0;

while ~feof(fileID)
    line = fgetl(fileID);
    % Try matching the first pattern
    match = regexp(line, pattern_1, 'tokens');
    if ~isempty(match)
        total_waiting_time = total_waiting_time + str2double(match{1});
       count_waiting = count_waiting + 1;
    else
    % Try matching the second pattern
    match = regexp(line, pattern_2, 'tokens');
    if ~isempty(match)
            total_travel_time = total_travel_time + str2double(match{1});
           count_travel = count_travel + 1;
    end
    end
end

fclose(fileID);

% Calculate the average waiting time and travel time
if ( count_waiting > 0 && count_travel > 0 )
    average_waiting_time = total_waiting_time / count_waiting;
    average_travel_time = total_travel_time / count_travel;
end

fprintf('Average␣waiting␣time:␣%.2f\n', average_waiting_time/60);
fprintf('Count_waiting:␣%d\n', count_waiting);
fprintf('Average␣travel␣time:␣%.2f\n', average_travel_time/60);
fprintf('Count_travel:␣%d\n', count_travel);
```

APPENDIX: MATLAB CODE FOR GRAPHING DATA

Listing 3: handling data in MATLAB

```
clearvars, close all

% number of commuters - waiting time - until end of simulation
x=[5 10 15 20 30 50 80 150 200 230 250 300 330 350 400];
y_2trams=[43.93 43.87 43.76 43.54 45.27 45.35 46.77 79.38 92.61 102.84 106.21
    141.08 154.18 166.44 281.96];
y_1tram=[51.74 51.98 51.85 51.70 52.63 58.94 62.30 95.88 112.24 192.68 158.78
    244.11 325.93 473.90 548.46];
plot(x,y_2trams,'ko-',x,y_1tram,'ro-','LineWidth',2);
hold on
xline(20, 'g','LineWidth',2);
legend('2 trams', '1 tram')
grid on
xlabel('number of commuters'); ylabel('waiting time');

% speed - waiting time - until end of simulation
x=[5 8 10 12 15 16 17.5 19 20 25 30 35 40 45 50];
y_2trams=[107.93 69.37 62.25 49.20 36.00 35.40 30.15 30.38 28.80 28.86 19.59 19.34
     14.70 13.86 11.83];
y_1tram=[137.90 91.85 71.05 54.40 48.23 40.87 37.26 34.73 34.66 26.89 25.19 22.00
    16.88 16.78 12.88];
plot(x,y_2trams,'ko-',x,y_1tram,'ro-','LineWidth',2);
legend('2 trams', '1 tram')
grid on
xlabel('speed'); ylabel('waiting time');

% speed - travel time - until end of simulation
x=[5 8 10 12 15 16 17.5 19 20 25 30 35 40 45 50];
y_2trams=[127.18 71.99 56.75 55.70 33.66 40.28 28.98 27.29 36.16 27.80 19.69 16.48
     14.88 12.75 10.59];
y_1tram=[125.18 72.99 58.75 53.70 38.66 36.99 30.98 24.29 23.16 21.80 15.69 15.48
    13.88 11.75 8.59];
plot(x,y_2trams,'ko-',x,y_1tram,'ro-','LineWidth',2);
legend('2 trams', '1 tram')
grid on
xlabel('speed'); ylabel('travel time');

% capacity - waiting time - until end of simulation
x=[5 10 15 20 25 30 35 40 50 60 70 80 90 100 110];
y_2trams=[146.86 70.35 56.92 42.47 41.52 41.09 36.62 35.04 34.93 34.12 34.01 33.98
     33.82 33.14 33.33];
y_1tram=[213.71 86.43 67.33 50.87 48.75 46.64 46.23 45.45 45.42 44.35 41.02 40.59
    40.50 40.55 40.53];
plot(x,y_2trams,'ko-',x,y_1tram,'ro-','LineWidth',2);
hold on
xline(70, 'g','LineWidth',2);
legend('2 trams', '1 tram')
grid on
xlabel('capacity'); ylabel('waiting time');

% number of commuters - waiting time - final stop - 2 trams
x=[2 7 8 9 16 30 44 59 61 63 64 65 66 67 68];
y_2trams=[11.33 11.98 11.29 11.74 11.01 12.25 14.50 13.08 16.81 15.61 17.41 17.77
    21.68 20.89 22.33];
plot(x,y_2trams,'ro-','LineWidth',2);
hold on
xline(20, 'g','LineWidth',2);
grid on
xlabel('commuters'); ylabel('waiting time');
```

```
% number of commuters - waiting time - final stop - 1 tram
x=[3 6 8 15 19 25 34 47 56 57 58 62 68 73 77];
y_1tram=[21.72 21.78 21.65 21.80 21.62 21.76 22.06 22.79 23.59 24.20 24.53 25.51
    26.69 28.35 28.75];
plot(x,y_1tram,'ro-','LineWidth',2);
hold on
xline(20, 'g','LineWidth',2);
grid on
xlabel('commuters'); ylabel('waiting time');
```

References

Chambers, F., Cruz, C., & Serugendo, G. D. M. (2023). *Agent-based modelling of commuting patterns in geneva, switzerland, towards portraying urban expansion and land cover changes*. https://www.researchsquare.com/article/rs-3452421/v1

Gama platform. (2024). https://gama-platform.org/

Map. (n.d.). https://commons.wikimedia.org/wiki/File:Gen%C3%A8ve_-_Tramway_network_map.png

Sitg. (n.d.). https://ge.ch/sitg/sitg_catalog/sitg_donnees?keyword=TPG&topic=tous&datatype=tous&service=tous&distribution=tous&sort=auto

Smeets, E., & Weterings, R. (1999). Environmental indicators:typology and overview. https://www.eea.europa.eu/publications/TEC25

TPG. (n.d.). https://www.tpg.ch/sites/default/files/tpg_document_list/TPG%20-%20Green%20Bond%20Framework%20August%202022.pdf