

In-Depth Behavior Modeling of Transportation Networks: Description and Preliminary Results of a Subway Network Model

Gonzalo Barbeito

Universität der Bundeswehr München, Munich, Germany

gonzalo.barbeito@unibw.de

Abstract

This paper describes the conceptual ideas behind a computer-aided microsimulation model combining agent-based modeling and discrete event simulation in order to reproduce the complex behavior of a fictitious subway system. Such a model allows passengers to be both active and passive agents behaving according to the model rules, and also affecting them in return, for more realistic results. Decision support in this network can be approached from both the passenger and the network operator perspective, by correctly predicting ridership and system delays. Preliminary results are presented, together with some of the challenges faced throughout the development process.

1. Introduction

The occurrence of delays and other schedule deviations on transportation networks mean not only an inconvenience for passengers but also a significant challenge for transit authorities. In highly connected networks, the effect of a single delay rarely presents itself as an isolated event, commonly cascading into more severe schedule deviations or missing connections affecting individual users, the whole network and even other connected networks [1]. All over the world, and due to their operations complexity, major cities deal with and are significantly affected by this problem. Several examples of this problem and its consequences are found in [2, 3, 4].

A variable common to most delay incidents is *overcrowding*, with particularly strong effects during peak hours. Intuitively, the effect of overcrowding can be described as both cause and consequence of delays: a bus running late, for example, allows for more passengers to arrive at the station, which simultaneously may create a positive deviation from the average stop time of said bus. This variable, however, seems merely the immediate and visible cause affecting the performance of a network, instead of the root cause

of the problem. Several underlying variables such as weather [5, 6] and deficient infrastructure and maintenance policies [7], among others, have been shown to negatively affect a network's ability to preserve schedules. As an illustrative example, the situation of the New York City subway network presents a well-documented case on this topic. This city's subway network is one of the world's oldest and most used subway systems, with the highest number of stations in the world¹, and located in the most densely populated city in the United States. In such a complex system, a multitude of delay-inducing variables comes to play [8]. According to the Metropolitan Transportation Authority (MTA), a state-run agency responsible for public transportation in the state of New York, the delay incidence in this city rose steadily since 2014 mainly due to system overcrowding [4]. Even though explanations for delays abound, and while the blame has originally been assigned to a constantly increasing ridership, more recently deficient maintenance and investment policies were suggested as the main culprits [9, 10, 7]. The example of New York not only is interesting as a case study, it also has two main implications: on the one hand, and according to intuition, there seems to be a correlation between the underlying variables creating delays and the number of passengers in a system. On the other hand, it is easier for passengers and decision makers to assign all blame on overcrowding, given the easily observable nature of this feature. For these reasons, when such an undesired behavior is observed, a more in-depth analysis is required to point towards its real cause(s). This paper introduces the conceptual ideas behind an ongoing research aiming to address both implications, creating a bridge between underlying variables and the overcrowding effect. Specifically, this document will introduce a multi-method simulation model of a generic subway network, as well as several challenges in its development and preliminary results obtained using a simplified version of this model.

¹<http://web.mta.info/nyct/facts/ridership/index.htm>
(Last Accessed: June 1st, 2018)

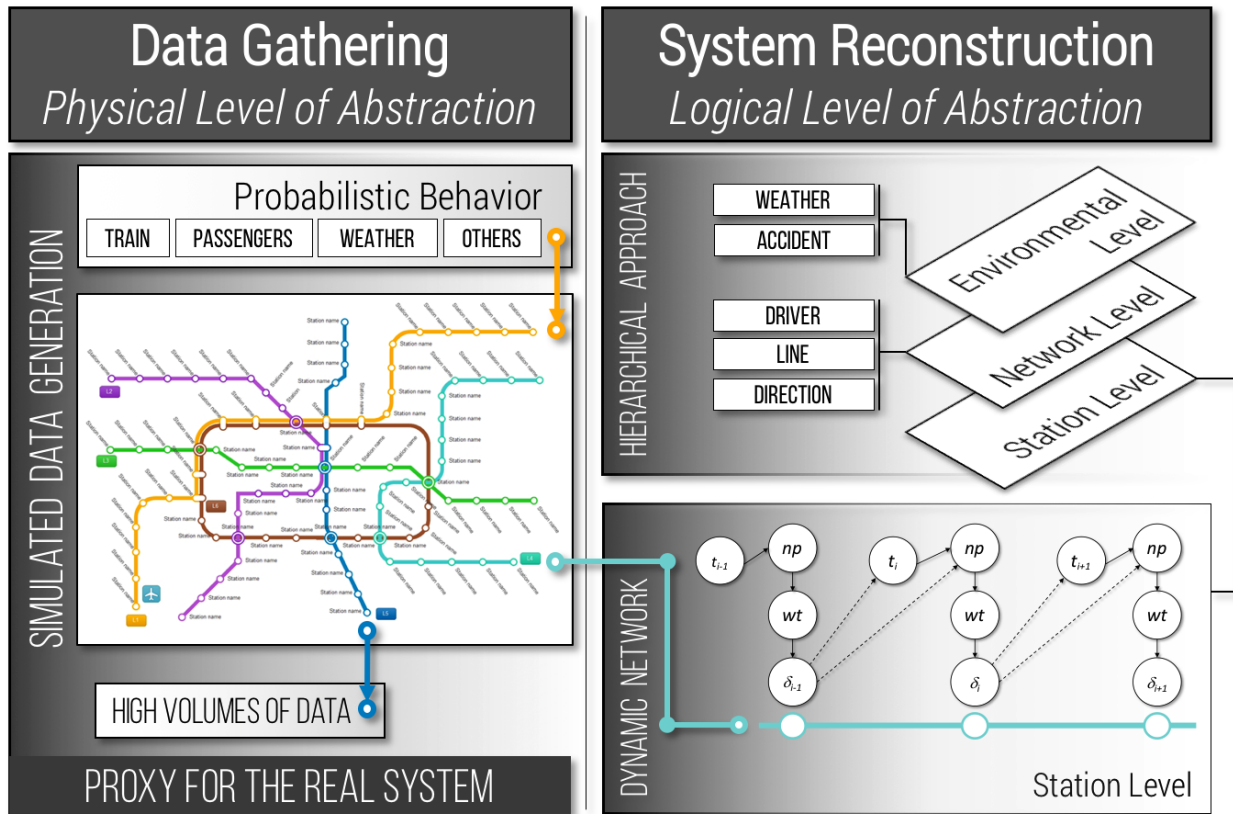


Figure 1. Conceptual Description of the Project's Proposed Methodology

1.1. Background

The field of Public Transportation modeling and simulation, as a general research area, is broad and exceeds the scope of this research. For this reason, the focus here is on a smaller subset of Simulation, called Microsimulation, which can be described as a simulation-based methodology utilizing micro-units of analysis to create emergent macro behavior [11]. In the context of general transit networks, commercial microsimulation tools have traditionally placed a stronger focus on vehicle traffic, relegating public transportation modeling to a secondary position [12], partially due to the difficulties of analyzing pedestrian flows. This shortcoming affects negatively the analysis of the network, given the significant impact of pedestrians and particularly overcrowding in the network's behavior. In the last years, however, some of the leading commercial software simulators started to explore improvements in this regard by incorporating pedestrian behavior in their modeling solutions [13]. Regarding academic research, the review presented in [14] describes some academic projects (as well as other commercial solutions) in this field and, more

recently, [12, 13] presented a platform capable of querying multiple simulation models and presenting joint results, combining simulation of road transit with passenger flow analysis. Specifically for subway networks, it is also worth mentioning the work of [15] and [16], addressing the lack of passenger flow analyses in subway networks, although from a micro perspective, without considering the whole network. These approaches, however, are not without drawbacks: these types of models take a significant amount of time to develop, and given the high specificity of each model, direct reusability is near to impossible. Furthermore, validation is still a challenging task when applying this approach, not only for disrupting scenarios but also for assessing realistic behavior in normal scenarios [12].

1.2. Motivation and Context

The model described here is framed in the context of an ongoing research aiming to provide accurate predictions of public transport schedules to be used in existing journey planning applications. In order to improve current routing solutions, and building upon some of the ideas introduced in [17], this project intends

to replace fixed schedules with probability distributions, where timetables are only used as a baseline solution for the mean of each distribution. As certain variables, such as weather, accidents, holidays, etc. are observed or forecasted, the baseline distribution for each time slot can be shifted, and its deviation from the mean can be reduced. It is worth noting that the reach of this project only considers the timetables updates with more accurate information, and not the routing procedure, which should be handled by existing tools².

With the intention of compartmentalizing similar processes, the project was divided into two different modeling tasks, as shown in Figure 1, the left side shows the use of an artificial model as proxy for a real system from which data cannot be easily retrieved, and the right side the reconstruction process of said proxy using probabilistic graphical models; In the bottom right, the station level figure describes a dynamic network, where the number of passengers (np) affects the waiting times (wt) causing delay in the line.

Essentially, the project is divided in:

- **Physical Level of Abstraction:** Gathering real data from complex systems can present several operative challenges (e.g. selecting a suitable system, legal concerns, data completeness, among others) and the final result may not be up to standards, due to noisy or incomplete data. In order to overcome these challenges, an alternative approach similar to the one presented in [19], although tailored to this context, was selected: instead of using real data from an existing system, a toy model³ is under development to simulate a wide variety of scenarios, including some that would seldom occur in a real system, and for which sampling would be near to impossible. Once the model is finished, high-dimensional datasets can be retrieved for a multitude of scenarios. Furthermore, given the model's ability to run, theoretically, indefinitely, an arbitrarily large number of samples for each scenario can be retrieved.
- **Logical Level of Abstraction** The second task requires the transportation system model to be rebuilt from the gathered data by creating a *logical* representation of the system using probabilistic graphical models, and under different conditions of

²Even though many routing apps rely on pre-computation according to and requiring fixed-timetable data, some solutions are also capable of handling real-time data [18]

³It is not the goal of this research to predict or analyze a real system, but to demonstrate the suitability of probabilistic models in capturing its complex causal paths. Therefore, it is not required to validate the model using real data.

system knowledge: (i) with no knowledge on the system other than the provided dataset (i.e. automatic model creation) and (ii) with a level of knowledge matching that of an expert (i.e. semi-automatic modeling).

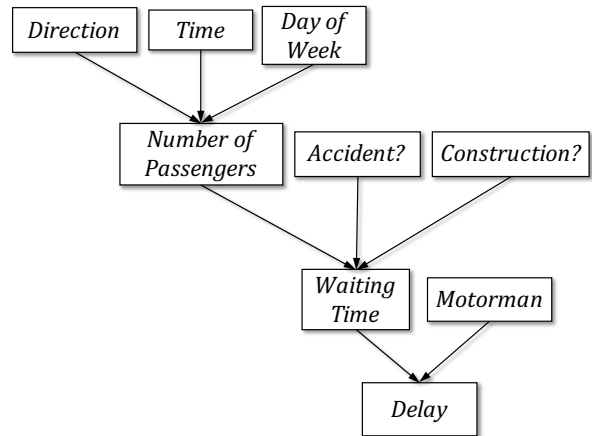


Figure 2. Conceptual example of the logical model.

The graphical structure shown in Fig. 2, describes both the ideas driving the physical model and the expected logical reconstruction to be achieved from data. In this trivial example, the combination of *Direction* (towards or outwards city center), *Time of the Day* and *Day of the Week* will affect the probability distribution of the *Number of Passengers* waiting at a station. Similarly, the *Number of Passengers* and the occurrence of an *Accident* or *Construction* may affect the *Waiting Time* before departure at each station. Finally, *Waiting Time* and the experience of the *Motorman* (also known as train operator) will define a probability distribution for the delay observed at each station. This example is an oversimplification of the intended behavior, the final project will try to depict a more realistic behavior in both the physical and logical models.

According to the progress achieved so far, the focus of this paper is on the first task, describing the physical model of a subway network developed for the ongoing research.

2. Model Description

As previously stated, the selected system for this analysis is a subway network where delays are modeled as probabilistic variables depending on several other random variables such as weather, holidays and accidents, among others.

The modeling technique selected for this task currently combines Discrete Event Simulation (DES) with Agent-Based Modeling (ABM) (although the use of System Dynamics (SD) is also contemplated for passenger behavior) using the modeling software Anylogic. A conceptual description of this idea is shown on the left pane of Figure 1. Several probabilistic behaviors, such as the impact that weather, busy schedules or accidents may present in vehicle and passengers' flow were considered for inclusion in this model.

The model is being developed pursuing the following goals, all framed within the research previously described:

- Observe and analyze the interplay between multiple variables, and push the system to out-of-control behavior.
- Measure, with a high level of detail, the impact that several endogenous and exogenous variables have on timetable schedule deviations.
- Provide a graphical interface allowing to follow train and passengers' behavior while on each station. This requires the modeling of each station as separate entities.

The current version of the model can already deal with the first and third goal, although the implementation of a detailed schedule and the impact of weather in the number of passengers are still under development. In its present state, the model is divided into four main hierarchical levels, as described next:

2.1. Network Level

On the network level, the railway structure is graphically declared, including the location of stations within the network and the real-time tracking of each subway traveling in the system as shown in Fig. 3. The logic in this level is simple: the model generates three types of trains (e.g. lines A, B and C) according to a schedule and places them in the tracks on a special node called "warehouse". The warehouse only acts as a platform for adding or removing trains without risk of collision between them, the model logic automatically decides if the next train in the schedule is allowed to enter the railway. Each train created formally corresponds to an agent modeled according to the "train" template and is indexed and added to the trains population.

This level is also used to aggregate control variables over the whole network, such as the total number of passengers in the network and per station, number of

trains and several others.

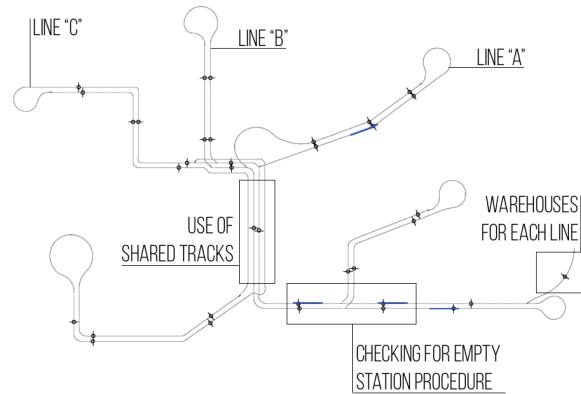


Figure 3. Subway Network Structure

Figure 4 shows the network structure used in this model as well as certain interesting aspects of it: The trains are stored in warehouses, serving as both source and sink of trains in the network. The central station in the network shares tracks between two lines and allows a train from one line to wait for another one going in the same direction, although this behavior is not yet included in this model. In Fig. 4, the subsystem in charge of subway generation is shown. All three sources (*sourceLine A, B and C*) create agents that are later sent to the train level, with its own logic. The basic (and still flawed) schedule used in this version depends on the injection of subways in the network every 10 minutes until the maximum number of trains in the network is reached. The variables *StationsLineA, B and C* contain the order of stations followed by each line, and some variables keep track of aggregated metrics in the network. The variable *OccupiedStations* is a boolean vector keeping track of the station in which each train is, and is used for avoiding train collisions.

2.2. Station Level

One tier down, each station has been individually considered and parameterized, with incoming and outgoing trains and passengers. The design at this level is based on a slightly modified version of an example model developed by and included with the software Anylogic and allows to register a heat map according to the most frequented locations of the passenger on each station. The logic in this part of the model creates a new track segment and copies the train from the network level as soon as it arrives at a station, holding it locally until all passengers are off and on the train with probabilistic time according to the passengers' quantity.

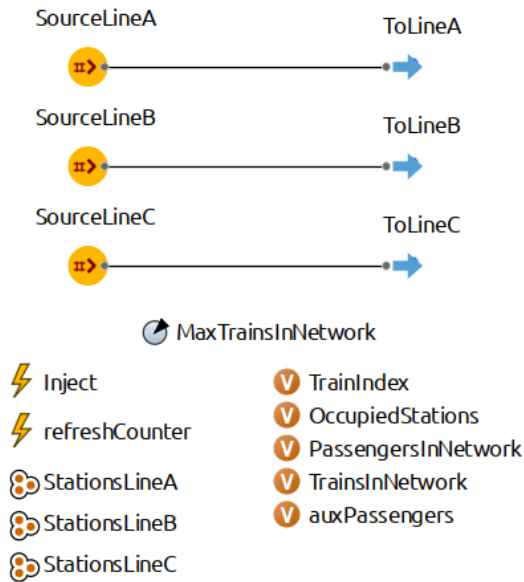


Figure 4. Network Level Subsystem

The idea driving the behavior of this section works differently than traditional models: instead of defining a probabilistic window of time for the train to wait at the station, it is the passengers' flow (within certain limits) the one defining the waiting time on each station, copying the real behavior of a subway. In order to achieve a higher level of realism, granular passenger behavior is also defined here, where agents randomly select from 32 doors on the subway wagons to go in, and during the unloading process, they choose among several station's exits. Currently, the model has no seat assignment or passengers' placement model within the wagon, although it is contemplated for a future iteration.

There are two variables in this level that define the level of service on each station according to a centrality value: the key idea here is that the closer a station is to the city center, the more passengers it will attract. This means that the first variable corrects the loading values (a station where the train is going in the direction of the city center will have higher load values, than those where the train is directed towards the suburbs) and vice-versa for unloading values.

An important subsystem included in this level is the interaction train-passenger when the train arrives at the platform. Fig. 5 shows the structure of this model, it can be seen how the train arrives in the *enter* block and a logical waiting period is assigned for the time it takes for the graphical representation of the subway to stop. In *PassengerUnload*, a certain amount of passengers leaves the train through the *passengerExit*

block, and according to the centrality values described before. *passengerEnter* is used to take passengers from the station into the train, this behavior lasts a stochastic amount of time which is repeated until no more passengers try to enter the train or until the train is at full capacity. Finally, the train leaves the station, and another logical waiting period is used to match the graphical accelerating subway representation.

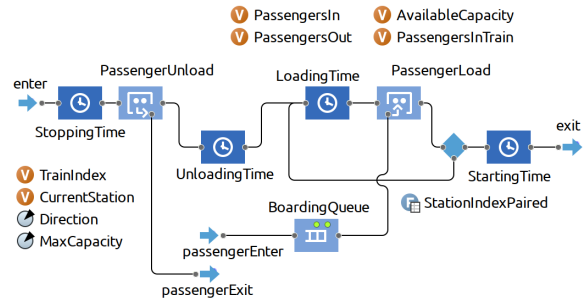


Figure 5. Subway and Passengers Interaction at Each Station

2.3. Train Level

The train subsystem is in charge of each individual subway agent (i.e. each train in the network), particularly the movement between stations and their behavior to avoid collisions between two or more trains traveling on the same railway. There are only three types of train types in this model, corresponding to the fictitious lines A,B and C. All three of them possess a different routing logic (in the form of a sequence of stations), which is read from the *PhysicalStationIndex* variable. All other variables in this part of the model referring to stations are used to decompose paired stations into individual tracks (i.e. towards or outwards the city center), each one of them with its own parameters.

Fig. 6 shows the building blocks of this logic: The trains created in the network level enter through *enter1* and *enter* to complete a closed movement loop. While on this loop, the trains verify next station's status and wait for clearance before proceeding. If the station is free to proceed, then the subway goes to *trainMoveTo* and drives to the next station, where the *exit* blocks leads it back to the station level.

2.4. Passenger Level

This level so far only includes a basic pedestrian logic included. In [20] a review of common techniques in pedestrian modeling was conducted, some directly

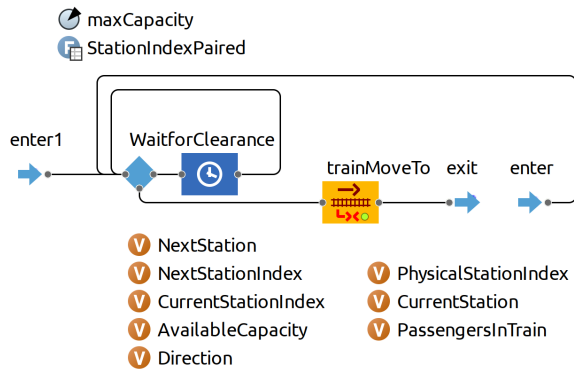


Figure 6. Train Logic Subsystem

applicable to this model for more realistic behavior in a general context. Particularly to this system, the idea is to include, in a future iteration, the passengers' decision making process according to delays or overcrowding, allowing them to select alternative routes or leave the system by replacing the transportation mode.

3. Preliminary Results

The results presented in this section are described from a quali-quantitative perspective since the goal is to show the suitability of this approach to reproduce certain complex behavior observed in subway networks and not to accurately model an existing system. Some noteworthy aspects include passenger behavior within the stations and in the whole network, collision avoidance system between trains and passenger loading and unloading at the station.

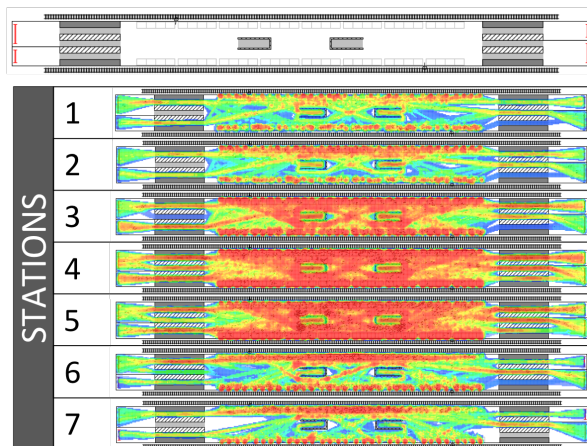


Figure 7. Passengers' behavior at each station

Regarding the passengers' behavior, Figure 7 shows the different load at each station. The load distribution

is evidenced in the heat-map⁴ color coded from green (i.e low passenger density) to red (i.e. high passenger density), at each station (1 to 7, both sides) and the upper figure corresponds to an empty station to understand its structure: it contains four exits (two laterals and two in the middle) and two lateral entries, as well as two tracks for trains arriving in opposing directions. Station 4 corresponds to the central station, and 1 and 7 are peripheral. A more detailed inspection shows also the movement behavior of passengers and points in which they wait for the train to come. The aggregated load of the network is also measured, and Fig. 8 shows a plot over time for a typical simulated day, which is key to assess model stability. The results show that the model increases the number of passengers in a transitory state until it reaches a weak equilibrium averaging around five thousand passengers for the whole network. The randomness found around this average is mostly due to the granular design of the model, where each agent, be it passengers or trains, are assigned physical properties, which upon interaction create a complex behavior.

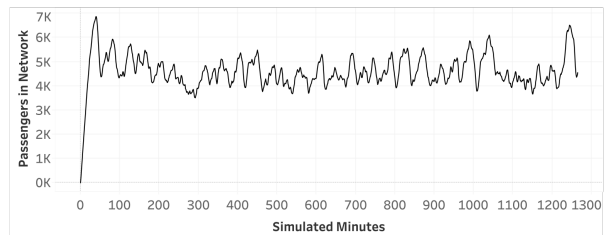


Figure 8. Passenger Load for the Aggregated Network

Finally, the results stemmed from the collision avoidance logic embedded on each train. Figure 3 partially shows this behavior (inside “**checking for empty station procedure**”): the train on the outer station cannot move to the next station until the one occupying it doesn't leave. While this logic works as intended (otherwise the simulation would crash), significant instability is introduced on the schedule. This phenomenon is called headway instability (HI), and is produced by an aggregation of vehicles in certain network areas. In [21] a model is introduced to study the characteristics of HI, and some strategies are proposed to improve system efficiency. Future iterations of the present model will adopt these strategies and measure the impact on a subway network.

⁴The heat-map function as well as the station's graphical distribution and logic were borrowed from the following Model: <https://cloud.anylogic.com/model/ea70770-1328-4c62-889b-13a9d09fb6bf?mode=SETTINGS> (Last Accessed: June 1st, 2018) and are not developments from the author.

4. Challenges in Modeling This System

There are three main challenges that still need to be overcome in this work:

- **Correct Parameterization:** The current model implementation deals with made-up behavior, where physical parameters are not completely realistic. Some examples include pedestrian walking speed above average values, and track and train dimensions not up to scale.
- **Detailed Scheduling:** The scheduling implemented is overly simplistic and needs to be calibrated by the minute. Ideally, the model should be able to run indefinitely, hence requiring several granularity levels in the schedule planning, including minute-level timetables, hourly load corrections and weekend and holidays definitions.
- **Model Complexity and Running Times:** Pursuing a high level of detail on the model meant to define individual agents for several elements in the network, such as passengers, trains and stations. Each agent introduces a great deal of complexity to the model and needs to be handled with care or the simulation cannot be completed. The current implementation struggles mostly with the transitory period in the beginning of the simulation, where a high number of passengers enter the system, although the performance improves as they reach the destination and leave the system.

5. Conclusion

In this paper, a new model of a Subway Network was introduced. This microsimulation model presents the particularity of being driven by a combination of atomic agents' behavior and stochastic macro rules, instead of exclusively leaning towards any extreme of the modeling spectrum. The goal of this project is fundamentally different from other research working with microsimulation. While most models aim to capture real complex behavior, this model attempts to create complex behavior using only sensible rules and assumptions, with validation requirements limited to physical system properties, not specific to any transportation network. Furthermore, while most of the reviewed models focused on endogenous system behavior, this model lays the ground to describe the effect of variables exogenous to the transportation network on the subway system. Thanks to the focus on easily explainable system behavior, a modeler can intuitively play with different variables and scenarios

and find alternative strategies that may extrapolate well to a real system.

Even though the first results are promising, the behavior considered so far is not yet showing the complexity or correctness that the project requires. In its current state, this model serves as a proof of concept of the potential of this approach and a preliminary version upon which the final model will be built. There are two main directions that will be explored next: the first involves variables equally affecting all the network, a typical example is weather, where certain climatological conditions may introduce more passengers than the network's average, up to the point in which extreme conditions may disrupt the system. The second direction is the inclusion of events restricted to a segment of the network, with different impact degrees in other segments; accidents, special events and constructions all fall in this category. Finally, and even though it is not the main goal of this model, a more realist qualitative behavior could also be embedded in the model, by observing the particularities of several subways in major cities.

References

- [1] E. Nagy and C. Csiszár, "Analysis of Delay Causes in Railway Passenger Transportation," *Periodica Polytechnica Transportation Engineering*, vol. 43, no. 2, pp. 73–80, 2015.
- [2] S. Murakami, "Japan's trains always on time? Report highlights frequency of rush-hour delays in Tokyo - The Japan Times," 2018.
- [3] L.-Y. Chiang, R. Crockett, I. Johnson, and A. O'Keefe, "Passenger Flow in the Tube," tech. rep., 2017.
- [4] E. Fitzsimmons, F. Fessenden, and R. K. Lai, "Every New York City Subway Line Is Getting Worse. Here's Why. - The New York Times," 2017.
- [5] S. Tao, J. Corcoran, F. Rowe, and M. Hickman, "To travel or not to travel: 'Weather' is the question. Modelling the effect of local weather conditions on bus ridership," *Transportation Research Part C: Emerging Technologies*, vol. 86, no. April 2017, pp. 147–167, 2018.
- [6] Metropolitan Transit Authority, "MTA Storm Performance Review," tech. rep., 2011.
- [7] S. Maslin and B. M. Rosenthal, "'Overcrowding' Is Not at the Root of Delays, Subway Chief Says - The New York Times,"
- [8] Metropolitan Transit Authority, "MTA Transportation Reinvention Commission," tech. rep., 2014.
- [9] E. G. Fitzsimmons, "Key to Improving Subway Service in New York? Modern Signals - The New York Times," 2017.
- [10] B. M. Rosenthal, E. G. Fitzsimmons, and M. LaForgia, "How Politics and Bad Decisions Starved New York's Subways - The New York Times," 2017.
- [11] C. O'Donoghue, *Handbook of microsimulation modelling*. Emerald Group Publishing, 2014.

- [12] S. Srikukenthiran, “Integrated Microsimulation Modelling of Crowd and Subway Network Dynamics For Disruption Management Support,” 2015.
- [13] S. Srikukenthiran, A. Shalaby, and A. Shalaby, “Enabling large-scale transit microsimulation for disruption response support using the Nexus platform Proof-of-concept case study of the Greater Toronto Area transit network,” *Public Transport*, vol. 9, pp. 411–435, 2017.
- [14] C. E. Cortés, V. Burgos, and R. Fernández, “Modelling passengers, buses and stops in traffic microsimulation: review and extensions,” *Journal of Advanced Transportation*, vol. 44, no. 2, pp. 72–88, 2010.
- [15] M. Ercolani, A. Placido, L. D’Acierno, and B. Montella, “The use of microsimulation models for the planning and management of metro systems,” in *WIT Transactions on The Built Environment*, vol. 135, pp. 509–521, WIT Press, 2014.
- [16] Q. Zhang, B. Han, and D. Li, “Modeling and simulation of passenger alighting and boarding movement in Beijing metro stations,” *Transportation Research Part C: Emerging Technologies*, vol. 16, pp. 635–649, oct 2008.
- [17] K. Bérczi, A. Jüttner, M. Laumanns, and J. Szabó, “Stochastic Route Planning in Public Transport,” in *Transportation Research Procedia*, vol. 27, pp. 1080–1087, 2017.
- [18] H. Rowley and S. Baluja, “Transportation Routing - U.S. Patent No. 7,908,080,” *U.S. Patent No. 7,908,080*, 2011.
- [19] N. Patki, R. Wedge, and K. Veeramachaneni, “The synthetic data vault,” in *Proceedings - 3rd IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016*, pp. 399–410, 2016.
- [20] M. Zsifkovits and T. Pham, “Modelling and parameterizing pedestrian behaviour in public places: A review.,” *International Journal of Simulation Modelling (IJSIMM)*, vol. 16, no. 4, 2017.
- [21] C. Gershenson and L. A. Pineda, “Why does public transport not arrive on time? The pervasiveness of equal headway instability,” *PLoS ONE*, vol. 4, no. 10, 2009.