

Fast Agent-Based Solution to Evaluate the Matching of Public Transport Offer vs Citizen Mobility Demand

Marco Fanfani, Alberto Giovannoni, Luciano Alessandro Ipsaro Palesi, Ammarah Irum, and Paolo Nesi(✉)

Department of Information Engineering, Distributed Systems and Internet Technology Lab (DISIT), University of Florence, Via Santa Marta, 3, 50139 Florence, Italy
paolo.nesi@unfi.it
https://www.disit.org/, https://www.snap4city.org

Abstract. Public transportation is fundamental to support the mobility demand of citizens and drive the development of sustainable and green mobility in smart city scenarios. However, planning an efficient public transport system is a challenging task since it requires on the one hand to deploy a widespread, fast, and reliable system able to fully respond to the mobility demand. On the other hand, financial sustainability must be guaranteed by avoiding excessive and unfruitful expenses that could in the long run hamper the whole transportation system. Therefore, to find such a balance, there is the need for tools able to analyze and evaluate the match between the transportation offer and the mobility demand to allow city officers and decision makers to plan optimal services. To respond to this necessity, in this paper we propose an agent-based solution, that, differently from available software, is easily configurable, requires minimal input, and is able to carry out simulation in a timely manner yet providing useful and accurate insights. Moreover, the proposed solution can be easily adopted for what-if analysis frameworks, thus realizing an extremely valuable tool for smart city transportation planning.

Keywords: Public transport offer vs demand · Simulation · Agent-based · Smart mobility

1 Introduction

Due to an ever-increasing urban population, public administrations must face novel and complex challenges spanning several domains to guarantee citizens a high quality of life. Among them, public mobility is one of the significant issues and it is critically significant to minimize the gap between the Transport, Public or Private, which is the offer and the mobility demand while maintaining optimal operational cost at transit provider's end [1]. Moreover, United Nation, UN, Sustainable Development Goals, SDGs, related to sustainable cities and communities indicate that nearly half of the global urban population lack feasible and equitable access to public transport [2]. Thus, to calibrate the public transport offer, transportation agencies make significant assessments for planning their offer through specific simulations aiming at optimizing resources such as time and cost.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2025, LNCS 15898, pp. 255–272, 2026.
https://doi.org/10.1007/978-3-031-97657-5_16

In the context of smart cities, the availability of diverse static, historic and real-time data, obtained from multiple sources such as transportation data, mobile cellular data, sensors data, vehicular and persons mobility data [3], can enable the development of new solutions fundamental in shaping the future of urban transportation infrastructure and helping the policy makers to make data-driven decisions. The data-driven insights play a pivotal role in evaluating the efficacy of demand-responsive transportation frameworks by laying a fact-based foundation for decision-making processes, thus ensuring an optimal, cost-effective and socially viable design of public transport infrastructure. To reinforce smart mobility in an urban setting Smart City Digital Twins (SCDTs) platforms [4] are fundamental decision support systems for suggesting and assessing the effects of infrastructure and service changes through simulated analysis and automatic optimization before introducing such changes in the real environment [5], the what-if analysis. Thus, in transportation offer modeling, the need to have a flexible, user-friendly and fast simulation tool capable to address the challenges of complex scenarios is evident.

Over the years, Agent Based Modeling, ABM, has emerged as a leading technique to simulate complexity in diverse social systems, including mobility [6]. ABMs, by representing individual agents as autonomous entities interacting with each other and the environment, have been extensively employed at micro-simulation scale accounting for details such as pedestrians' and drivers' behaviors [7]. Agents act as autonomous entities in the simulated environment where they are capable to achieve their specific objectives [8]. Such autonomous behavior of agents makes ABMs particularly appropriate for the representation of agents in transportation offer vs citizen mobility demand. However, despite the effectiveness of various simulation software, building transportation demand modelling simulation scenarios is a complex and time-consuming process that requires high technical experience or compromises [9]. Thus, a few ABM systems have been proposed to address the complexity of transportation systems. *SimMobility* is a multi-scale, comprehensive agent-based simulation platform that integrates demand modeling and urban mobility analysis to support decision-making in transport planning [10]. However, *SimMobility* significantly demands computational resources making large-scale or long-term simulations particularly difficult. *AnyLogic* is a multimethod, agent-based simulation platform that integrates system dynamics modelling, enabling complex system analysis and data-driven decision-making across numerous domains, including transportation and logistics [11], while the high licensing cost limits accessibility for smaller organizations and academic institutions. Additionally, large-scale agent-based simulations require relevant computational resources, which can affect performance in highly complex scenarios. *NetLogo* provides multi-agent modelling environment focused on adaptive interactions and decentralized decision-making across various scientific domains [12]. *NetLogo* has been designed for models of moderate size and becomes inefficient as the number of agents increases significantly and programming knowledge is required to configure agents. *GIS and Agent-based Modelling Architecture, GAMA,* is an agent-based platform that couples heterogeneous data and modeling paradigms to simulate transportation scenarios [13]. It requires high computational resources, especially for large-scale scenarios. *Simulation of Urban Mobility, SUMO,* is multimodal, open-source traffic simulation software that is capable to address demand modelling alongside other transport related issues [14]. SUMO has some computational

limits when dealing with huge scenarios (in terms of number of agents, since each person is also an agent) since it requires careful setups, and it is time-consuming to run. *Multi-Agent Transport Simulation, MATSim,* and *TRANSIMS* have been designed to provide information on traffic impacts to transportation advisors [15, 16]. However, together with *SimMobility*, *MATSim* and *TRANSIMS* simulators focus on specific problems (e.g., traffic assignment generation) making them more purpose-oriented, less interactive and flexible than other simulators [17]. For instance, *MATSim* is an activity-based, large-scale multi-agent transport simulator which is not ideal solution for multi-day simulation as it takes long time to reach an equilibrium state with longer period of simulation due to the increase in the number of possible activities per agent to be tested [18]. Additionally, *MATSim* representation of public transport is limited, it is less detailed than specialized systems and may be approximate in handling schedules and vehicle capacities. The *Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks, Aimsun,* can be deployed to analyze exigency handling for various transportation problems by traffic engineers and policy makers. Multiple tools are provided within the software from simulating dedicated bus journeys to analyzing large-scale regions using microscopic, mesoscopic and hybrid simulation methods [19]. *PTV Vissim* for micro area modelling [20], is one of the comprehensive tools to perform simulations for public transport, urban planning, traffic engineering, etc. However, it requires expensive subscriptions and considerable effort in simulation setup.

It's worth noticing that most of the above-mentioned ABM tools demand settings on detailed parameters including road networks, vehicle routes, speed, and flows, traffic light signal configurations, pedestrian behaviors and activities, at any aggregation level [21]. The management of single agent for each pedestrian creates a significant barrier to users limiting their capacity to configure transport scenarios rapidly and effectively and can be a hindrance for those who need flexible and quick access to results, such as policy makers, or urban planners [22]. Moreover, they can be computationally expensive having to model any minimal interaction between the numerous agents and the environment, thus limiting their applicability in huge scenarios and for what-if analysis, that usually requires to instantiate multiple scenarios to assess the impact of possible changes. Finally, requiring complex configurations, is very difficult to set up those tools without human intervention, preventing their exploitation in automatic optimization solutions as the one described in [5].

A different approach was proposed by *DORAM* in [23], where a statistical model was defined to analyze public transport offer against demand achieving fast-computability and producing Key Performance Indicators, KPIs, about the number of pick-ups and drop-offs at each bus stop. This method required as input the total daily number of pick-ups and drop-offs that are distributed among the transportation stops by taking into account the probabilities of each stop to be a start, end, or transit and exploiting some weights (calibrated on the basis of actual measures gathered by PT service providers) to model the commuting probability. However, this method was not able to provide relevant KPIs like the unmet demand or the mean travel time and requires some calibration data not easy to obtain, limiting its applicability.

In Table 1, a summary of the comparative analysis for the discussed microsimulation software is reported considering their functionality, focusing on critical parameters such

Table 1. Comparison of simulators for PT offer vs mobility demand analysis: (✓) indicates that the solution satisfies the requirement, (×) non satisfied. (~) partial fulfillment, and (?) not clear accessible information to be assessed.

Simulator	Open source	Multi-modality	Scale on huge scenario	Fast computation	Easy & fast setup	Produce KPIs
SimMobility	✓	~	×	×	✓	✓
AnyLogic	×	✓	×	×	✓	✓
NetLogo	✓	✓	×	×	×	✓
GAMA	✓	✓	×	×	✓	✓
SUMO	✓	✓	×	×	×	✓
MATSim	✓	~	~	×		✓
TRANSIMS	✓	~	×	×	✓	✓
Aimsun	×	✓	?	?	✓	✓
PTV Vissim	×	✓	✓	?	×	✓
DORAM	✓	×	✓	✓	×	~
Proposed in this paper	✓	✓	✓	✓	✓	✓

as open-source availability, multimodality (i.e., the possibility to model different means of transportation like buses, tram, train, etc.), scalability on huge scenarios, computational efficiency, ease of setup, and the ability to provide KPIs to the decision makers. The findings indicate that free accessibility is a dominant characteristic among most simulators, with significant exceptions that include *AnyLogic*, *Aimsun*, and *PTV Vissim*. Multimodality is widely supported as well as sufficiently relevant KPI production. However, a few solutions seem to be usable on huge scenarios and long timespans. Additionally, computational efficiency remains a challenge for most simulators and the possibility to have easy and fast setups is rarely offered, since often manual customization (which sometimes requires programming skills) is required to create an error-free road network, setting-up several elements, such as vehicle trajectories, OD (origin destination) matrices, activity plans for citizens' mobility, traffic signal temporization, etc.

For these reasons, in this paper, we propose a novel open-source agent-based approach to address the problem of the transportation offer and demand matching by satisfying significant requirements into the system while focusing on strongly reducing computational costs. Thanks to multithreading architecture and a reasoned simplification of the ABM paradigm, the proposed solution is able to carry out simulations with strongly reduced execution time, without requiring complex setups, and it is capable of scaling on huge scenarios including whole cities spanning on multiple days providing sensible KPIs to assess the quality of the transportation offer with respect to the actual demand of mobility. Such improvements allow the system to be effectively exploited in a what-if analysis framework and eventually be enforced into a digital twin platform for

public transport optimization. This work has been carried out in the context of the Spoke 9 of the Italian National Center for Sustainable Mobility (CN MOST) [24], and the CN MOST flagship OPTIFaaS and scalability project SASUAM. The proposed solution has been developed in the Snap4City framework [25, 26], https://www.snap4city.org.

The paper is organized as follows. In Sect. 2, after presenting a series of functional requirements defined to guide the method development, the solution is described. In Sect. 3, a case study is reported to show the effectiveness of the proposed approach illustrating its applications in a what-if analysis. Finally, in Sect. 4 conclusions are drawn.

2 The Proposed Solution

As stated in the previous section, in this paper we propose a novel agent-based simulator, developed exploiting the Snap4City open-source framework [27. 28], to analyze the urban transport and identify possible deficiencies in terms of under or over provisioned services with respect to the citizen mobility demand.

To guide the design of this novel solution the following functional requirements have been identified:

R1. **Selection of the working area**: the system must allow the user to define the area of interest in which the simulation has to work, from city hamlets to whole metropolitan areas.
R2. **Selection of the working timespan**: the system must allow the user to define the timespan on which the simulation runs, from a few hours to days.
R3. **Offer specification format compatibility**: to set the transportation service input the system must be able to accept at least the GTFS format [29] to specify PT services routes, lines/rides, bus stops, timetables, etc. (**R3.a**). Optionally, other standards should be supported, e.g., NeTEx [30] (**R3.b**). Additional information on transportation vehicles' capacity should be supported (**R3.c**).
R4. **Multimodal solutions**: the simulator must be able to consider urban travel solutions including different transportation modalities, like for example buses, trams, and trains, and optionally also the sharing offer.
R5. **Mobility demand specification format**: to know the citizen mobility demand in the form of Origin-Destination Matrix (ODMs) commonly used representation formats must be supported, as for example the Matrix Market Exchange Format, MTX, [31] (**R5.a**). Additionally, geographical data formats like Shapefiles or GeoJSON, WKT, or standard as the MGRS [32], must be supported to define the shapes of the zones used in the ODMs (**R5.b**). In substance, ODM can refer to administrative shapes or regular grids at different scales.
R6. **Modal and Multimodal routing**: to produce trajectories for the city users in the simulation according to the ODMs, a routing service must be exploited (**R6.a**). This additionally requires information about the city road graph. Such routing must be performed in reasonable times to allow a fast simulation (**R6.b**).
R7. **Computational scalability**: the simulation must be able to run efficiently and with acceptable execution time, even for large areas, large timespan, the huge number of vehicles and pedestrians considered.

R8. **Track vehicle status**: the simulator must allow the operator to evaluate position and occupancy of each transport vehicle/service every time instant of the simulation.
R9. **Track stop status**: the system must allow the operator to count the number of people waiting at stops, getting in/out, and also their waiting time, and the number of people on each vehicle at each time instant of the simulation.
R10. **Key Performance Indicator (KPI) production**: once a simulation is completed, the system must produce a series of indicators to measure the performance of the transportation offer with respect to the specified demand. Such indicators could include mean travel time, meat waiting time at stops, crowed stops, vehicle occupancy, people or areas not served, number of served city users at each stop over time, number of people getting off, etc.

The proposed solution is able to satisfy all the above requirements as discussed in the following sections and demonstrated in Sect. 3.

2.1 Architectural Overview and Data Flow

Before performing the simulation, a setup of a few parameters is needed, such as to specify: the area of interest, the timespan, and the offer and demand inputs. This allows to define the context of simulation scenario. Wide areas (**R1**) can be specified by using the minimum and maximum latitude and longitude coordinates (bounding box) of the area under study. Datetime is used to set the simulation timespan, which may insist on several days (**R2**). To input the transportation offer, the GTFS files are supported (**R3.a**) and vehicle capacity can be specified (**R3.c**). Multimodal solutions are considered, allowing people to travel using buses, trams, and urban trains (**R4**). Before starting the simulation, the GTFS files (i.e., *agency.txt*, *calendar_dates.txt*, *routes.txt*, *shapes.txt*, *stop_times.txt*, *stops.txt*, *trips.txt*) are automatically analyzed and the data is reorganized as a compact representation into a single JSON where a structure for each single vehicle is defined. Such a structure includes the vehicle IDs (route ID, line name and ID, shape ID), day of operation, and the stop list, that is an array in which each stop is described with an ID, latitude and longitude coordinates, and time. This compact structure simplifies filtering to extract the vehicles that have to be considered in the simulation (having at least two stops in the simulation area and operating in the considered time window) and the agent construction during the simulation.

Demand ODMs can be loaded by using different file formats, including MTX (**R5.a**). At this moment our system supports MGRS areas to describe origins and destinations (**R5.b**). Note that additional geographic formats can be included in the future with minor effort, Snap4City already supports them as models. As for the offer, also the demand is filtered according to the simulation area and the used time span.

To obtain routes (**R6**), the GraphHopper router [33] is exploited providing as input the city road graph extracted from OpenStreeMap [34], and the GTFS files. This enables the possibility of obtaining multimodal travel paths by exploiting both walking and multimodal services. Note that, since ODM are provided according to specific geographic zones, it is possible to approximate the starting and ending positions of people's travels with the centroid of the zones, instead of randomly selecting a different starting/ending point for each person. Such approximation simplifies the computing of people's routing

since multiple people's routes can be obtained with a single request and has a limited impact on the evaluation of the transportation when the zones used are sufficiently small. Moreover, to provide a fast computation of the trajectories we devised a scalable system able to distribute the routing requests on multiple GraphHopper instances to balance the load and speed-up the computation. Additionally, a caching system based on a relational database has been realized to store previously computed routes to be quickly retrieved when executing new simulations on the same scenario with only partial changes in the demand. Thanks to multithreading programming and other measures described in the following, the proposed solution achieves fast execution times even on huge scenarios (**R7**) encompassing wide areas, simulating several days, and considering thousands of vehicles and passengers. Moreover, to reduce memory consumption both the transportation vehicles and the ODM data are dynamically loaded in the simulator: a vehicle is kept in the simulation only for the time required to complete its route, similarly pedestrians are generated into the simulation at their starting stop and removed once reaching their destination.

2.2 Agent Modelling

In the proposed simulator all transportation vehicles (buses, trams, and trains) are modelled as independent agents able to actively interact with the environment. This allows us to track the vehicle status, responding to R8, and indirectly track the stop/station status, fulfilling **R9**. Differently, pedestrians are modelled as passive elements that can be viewed as packages that appear at the stops and wait to take some transportation means to move over the city. Such a solution lightens the computational burden without estimating the status/position of each person at each time instant of the simulation. Indeed, pedestrians are following their precomputed routes. For this reason, we make people appear at the stops, with a delay with respect to their starting time related to the walking distance they have to cover between their origin and the transportation stop/station (this also allows them to estimate their intermediate positions if needed). Note also that the number of people that usually have to be included into a realistic large-scale simulation is much greater than the number of vehicles (usually hundreds of thousands of passengers are served by a few thousand transportation vehicles). Avoiding modeling pedestrians as agents is extremely advantageous to reduce computational times and required hardware resources.

During the simulation each vehicle follows its route according to the GTFS and stops at specific times in all the simulation areas in which one of their stops/stations falls (as for pedestrian starting/ending position, transportation stops are also mapped on the ODM areas). During the simulation, each vehicle collects the pedestrians waiting at the stops considering its max and actual occupancy, i.e., when the vehicle is full, additional people are not allowed to get on board. Then, the vehicle continues its path following the timings specified in the GTFS and changing dynamically its real time occupancy. In other simulators, e.g., SUMO, the movements of the vehicles are computed on the basis of all the variables considered in the simulation, thus increasing dramatically the computational times. Such behavior produces route timings that are very close to those reported in the GTFS, that is indeed created taking into account the real travel times. The proposed solution avoids this complexity to obtain a relevant speed up in the computation

of the simulation. It could be observed that this choice would limit the applicability of our simulator when considering particular days on which private vehicles running on the streets cause unusual traffic jams and consequent delays on the transportation routes (part for the cases in which reserved paths/lanes are dedicated to public transportation). However, these events can be also modelled by computing beforehand the level of traffic congestion, using for example [35], and then altering the travel times of the transportation routes according to the estimated reduced speed.

Every time a vehicle reaches a stop, some passengers may get off, in this case, the occupancy counter is decreased leaving space to get new passengers on board. When the person's trip includes a final transportation route, her/his travel ends, with an ending time augmented by the walking time required to reach her/his destination. On the other hand, if multiple (even multimodal) transportation routes are required, the pedestrian is moved to the next stop taking time, where she/he appears at a later time to take into account the walking time. Examples of modelled trips are shown in Fig. 1.

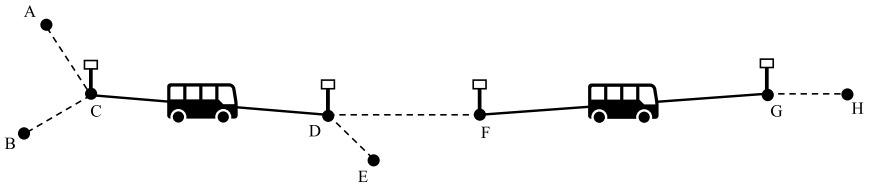


Fig. 1. Example of trips modeled in the proposed simulator. Suppose A and B are starting position for the travels of two persons, Alice and Bob. Both will reach the first stop in C and will catch the bus taking them to D. Then, Alice walks to E, ending her journey. Differently, Bob will walk to E, take the second bus, reach G and then walk to his destination in H. In our simulator movement done by foot are handled considering the required walking times. For example, if Alice starts from A at 9:00 and walking from A to C requires 7 min, she will appear at the first stop in C at 9:07. Similarly, if Bob drop-off in D at 9:30, he disappears and reappears in F at 9:42, considering a walking time of 12 min between D and F. Same considerations apply on the other paths done by foot (e.g., D–E and G–H).

The simulation runs for N time instants $T = \{t_0, t_1, \ldots, t_{N-1}\}$ and all vehicles are updated every $\Delta_V = t_i - t_{i-1}$ by checking their status and executing the pick-ups and drop-offs for those vehicles that at t_i have a scheduled stop in the GTFS. New pedestrians are added every $\Delta_P \gg \Delta_V$, according to the ODM data, and following the procedure described above. If a pedestrian enters the simulation at time t_j it will appear at the first stop indicated by the routing at a time $\hat{t}_j = t_j + \delta_w$, where δ_w is the walking time spent from the origin to the stop. δ_w, that could also be zero, is computed by the GraphHopper according to an average walking speed that can be set by the operator.

2.3 KPI Computation

Once the simulation is completed, the system produces in output a series of KPIs that can be used to summarize the performance of the transport system with respect to the citizen mobility demand (fulfilling **R10**). In our solution, we consider the following KPIs.

Unmet Demand: Total number of people that were not able to get on the transportation vehicle since its maximum capacity was reached. This highlights the main deficiencies of the transportation system and can be used to identify the most critical lines that are under provision.

Mean Travel Time (*MTT*) Ratio: By taking as *expected mean travel time* (MTT_e) the average travel time estimated by the routing algorithm for each person, and the *actual mean travel time* (MTT_a) as the average time measured after the simulation, we define the $MTT = MTT_a/MTT_e \in [1, \infty)$. Whether $MTT = 1$, the transportation works as expected, demonstrating its sufficiency in satisfying all people's demands. Otherwise, if values greater than 1 are achieved, this indicates some deficiencies, since increased MTT_a are due to the impossibility to take the expected transportation vehicle due to overcrowding.

Vehicle Occupancy Reports the number of passengers on the transportation vehicles. It can be represented as an average percentage with respect to the total vehicle capacity, or can be specified for single transportation lines, producing histograms to show how the number of passengers varies along a route or at different times.

Number of Pick-Ups and Drop-Offs: This KPI indicates the number of people getting on and off the transportation vehicles at given stops. This can give insights into the most used stops and provide also the **level of crowding** at given stops, which is useful to identify critical spots since crowded areas can be problematic for both security and sanitary reasons.

Average Waiting Time: This is the mean time spent by the pedestrian at the stop waiting for a non-fully occupied transportation vehicle that can take them to their destination. High waiting times indicate an insufficient transportation vehicle frequency.

Average Walking Time: This indicator counts the average time spent on foot by the pedestrians. If the transportation system is sufficiently widespread around the city such times should be low. Otherwise, there are probably underserved areas that force people to walk longer to reach the transportation stops. It is also possible to provide the number of pedestrians walking longer than a given threshold and potentially highlight the specific problematic areas. Note that at this moment this indicator can be directly computed from the router output. However, in the future pedestrians could be improved with a propensity to walk that could rise the longer they wait for the transportation vehicle.

In the next section an experimental case of study is presented to show the effectiveness of the proposed method.

3 Experimental Case Study

To show the effectiveness of the proposed method, we report in this section a case study carried out on the city of Florence, Italy. We consider an area wider than the Florence municipality, including also suburban areas like Sesto Fiorentino, Scandicci, and Bagno a Ripoli. In Fig. 2 a map of the area is shown. This area has been split into 4624 MGSR zones (i.e., a 64×64 grid), each cell with a dimension of 200×200 m.

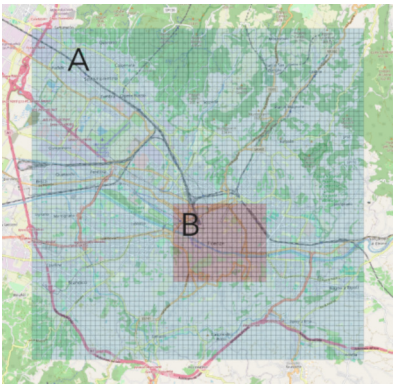


Fig. 2. Map of the area used for the case study (shown in EPSG:32632 for better viewing). The 4624 MGRS cells, with dimension of 200 × 200 meters, used to partition the area are showed over the map. In red the inner area A and in blue the outer area B used to generate the ODMs for the different time slots.

The simulation has been run on a timespan of five working days, from the 00:00 of November 11th, 2024 (Monday) to the 23:59 of November 15th, 2024 (Friday). Offer has been modeled according to GTFS data provided by the PT service providers, i.e. Autolinee Toscane (buses), Gest (trams), and Trenitalia (trains). A total of 8570 vehicles per day has been considered (7254 buses, 882 trams, and 434 trains).

Since we do not have access to real data to specify the citizen demand, we created simulated ODMs: considering **150000** passenger per day, we distributed the starting travel times of the pedestrians following to the traffic curve reported in [35] where a strong rise of trips is registered between after 07:00 with a peak at 09:00 in the morning. Then the number of trips slightly reduces and stays rather constat between 10:00 and 16:00, reaching a second peak at 17:00, followed by a progressive drop from 18:00 to 23:00. A histogram illustrating the number of passengers starting their journey in each hour is shown in Fig. 3. We use this temporal distribution for all five working days.

Differently, origins and destinations have been randomly selected, with independent samples for each day, using different origin-destination probabilities for zones in the outer (A) or inner (B) areas, as shown in Fig. 2. The inner area (B) includes the central part of Florence where most of the offices and attractions are located. Differently the outer area (A) includes the outskirts of the city with many residential districts. Therefore, we modelled ODMs such that between 05:00 to 10:00 we have a probability $P^{OUT}_{Orig} = 0.8$ to select one the outer areas as origin, and $P^{OUT}_{Dest} = 0.2$ to select those areas as destination. Complementarily, for the cells in the inner area we set probability of $P^{IN}_{Orig} = 0.2$ and $P^{IN}_{Dest} = 0.8$, respectively to select one of those cells as origin or destination. Then in the successive time slot, from 11:00 to 16:00, the probabilities have been changed as $P^{OUT}_{Orig} = 0.6$, $P^{OUT}_{Dest} = 0.4$, $P^{IN}_{Orig} = 0.4$ and $P^{IN}_{Dest} = 0.6$. Finally in the evening from 17:00 to 23:00, we set $P^{OUT}_{Orig} = 0.2$, $P^{OUT}_{Dest} = 0.8$, $P^{IN}_{Orig} = 0.8$ and $P^{IN}_{Dest} = 0.2$. Such probabilities have been defined to broadly model the people's flow that in the morning and during the afternoon tend to move from the outer area to the city center to go to work,

and in the evening perform the opposite trip to go back home. We are aware that this kind of data generation is a simplification of the transportation demand patterns in a city, thus producing ODMs that could not fully agree with real people's movements along the day. However, with these tests we want to demonstrate the capabilities of the proposed solution and not perform an actual analysis of the transport services in Florence, thus such an approximation is sufficient for the scope of this work.

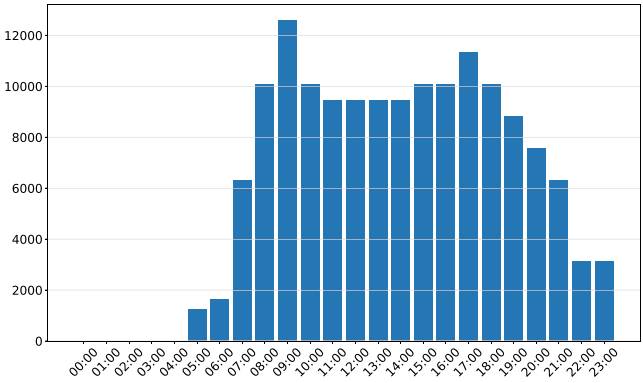


Fig. 3. Temporal distribution of starting time of people's trip used in the case study to generate ODMs.

The simulation has been executed on a workstation equipped with a CPU with 4 cores at 2.30 GHz and 24 GB RAM. In addition, four instances of the GraphHopper router have been exploited on different virtual machines, each with 29 cores at 2.30 GHz and 64 GB RAM, using a RAM-drive configuration. Queries toward GraphHopper instances have been performed using REST APIs over HTTP from the machine running the simulator. Query results were also saved on a relational database realizing the cache system described in Sect. 2.1. Note that two kinds of tests have been performed to assess the impact of the caching systems: in the first test, the database cache was empty and progressively filled. Then we repeated the test using the preloaded cache to observe the achievable speed-up. In Table 2 the computational times required to preprocess the GTFS and execute one day of simulation have been reported for the two tests. As can be seen, GTFS preprocessing requires around 30 min, of which most of the time is used to compute the compact JSON representation described in Sect. 2.1. Note that the JSON construction from the GTFS data is carried out only when a new GTFS is released, usually once every two or three months. Filtering the relevant vehicles from the processed GTFS and the pedestrians from the ODMs is instead a very fast operation, in the order of a few seconds. Once the preprocessing is performed, one day of simulation is typically completed in 2765 s, if the routing cache is empty and the GraphHopper services must be queried to obtain the routings. Differently, when the cache is available, simulation times are strongly reduced to 271 s on average, with a speed-up of 10 times. This can be extremely useful when the operator needs to repeat the simulation with minimal changes, as usually happens when carrying out what-if analysis looking for an improvement by changing the transportation offer with the same mobility demand.

Table 2. Computational times.

Step	Times (s)
GTFS conversion	1802
GTFS and ODM filtering	10
Simulation of 1 day (empty cache)	2765
Simulation of 1 day (using cache)	271

After completing the simulation, KPIs described in Sect. 2.3 are computed and made accessible for the operator. Hereafter some examples have been reported.

Table 3. Measured values for the following KPIs: unmet demand, MTT ratio, average waiting time, and average walking time (with in bracket the number of travelers having to walk for more than 20 min).

Day	Unmet Demand	MTT Ratio	Average Waiting Time (s)	Average Walking Time (s)
Monday	34885	1.217	472	362 (24428)
Tuesday	34813	1.217	475	363 (24301)
Wednesday	35077	1.223	477	361 (24133)
Thursday	34693	1.216	478	362 (24468)
Friday	34584	1.207	481	361 (24229)

In Table 3, the unmet demand, MTT ratio, average waiting time, and average walking time (with in bracket the number of travelers having to walk for more than 20 min) have been included. As expected, very few differences are appreciable among the days since the transportation vehicles schedule is stable for the working days and the demand can be produced by ODM or generated with the same random procedure. In assessing an actual transportation offer, such numbers would highlight that more than 23% of the demand is unable to reach the destination and among those able to arrive at the destination an average of 20% increased travel time is measured. Waiting times at the stops are in order of 8 min (for those that are able to get on the vehicles) while on average 6 min' walk are required to complete a trip (with only 16% of people forced to walk for more than 20 min).

To show the pick-ups and drop-offs, Figure 4 reports the histograms illustrating the number of people boarding and descending from buses at two representative bus stops. The first one (plots in Fig. 4, a, c, e, g, and i) is a stop in the inner area (San Marco square), while the second (see Fig. 4, b, d, f, h, and j) is a stop located in the outer area (Via Baracca, il Barco). As can be observed, at the inner area stop, most of the drop-offs are concentrated in the first hours of the days and decrease during the afternoon and evening. Conversely, at the outer area bus stop drop-offs are concentrated in the late hours. This

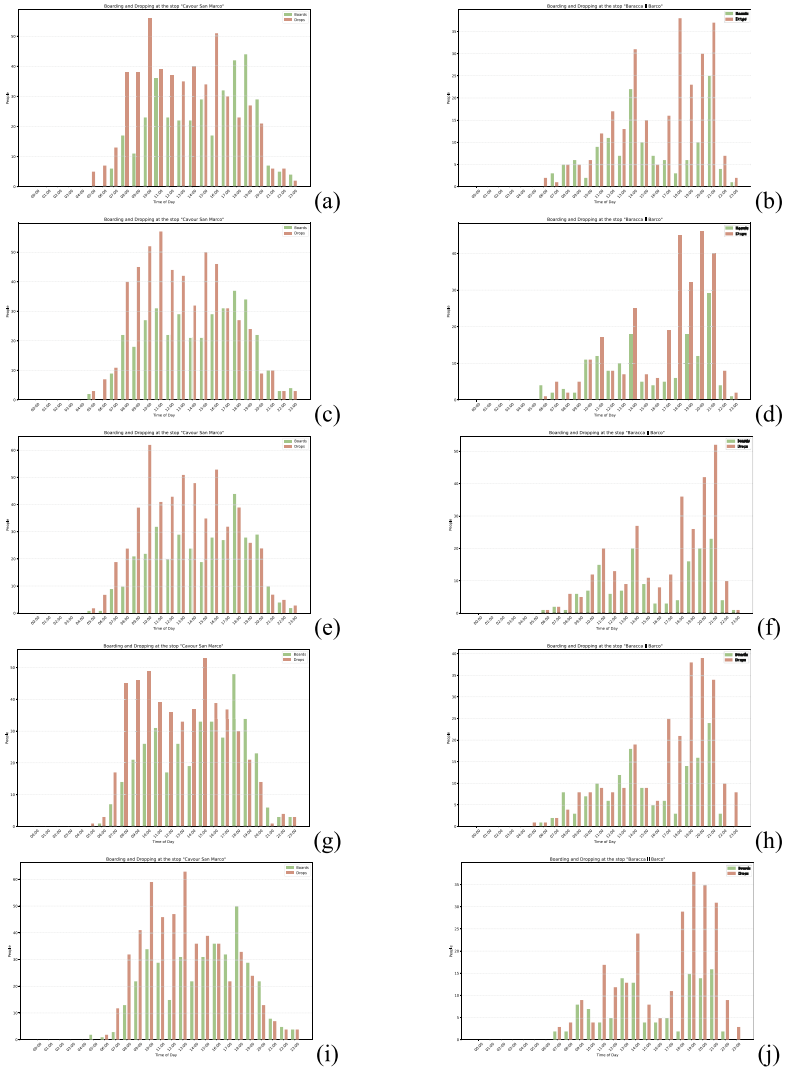


Fig. 4. Pick-ups (green) and drop-off (red) for two different stops. In the first column a stop in the inner area, in the right column a stop in the outer area. From top to bottom the different days of the week (from Monday to Friday).

behavior is aligned with the demand generated where the inner area is the most probable destination during the morning, while the outer area is a probable destination during the evening. Pick-ups for the inner area stop following a specular behavior concentrating in the second part of the days. Differently in the outer stop, pick-ups are not particularly concentrated in the morning. Indeed, the inner stop is a bus terminal for some of the lines stopping by, thus it is used both for trips going into the inner area and for those in the opposite direction. Differently, the outer stop is a transit stop from which the

buses pass then move further away toward external city areas, thus it is mostly used to travel from the inner to the outer areas in the evening. As illustrated in Fig. 5, it is also possible to show aggregated pick-ups and drop-offs for a given stop, thus obtaining the crowding level at the stop, and providing details on the contribution of the single lines (for example, in Figure 5, four bus lines stop at the same location).

Finally, the bus occupancy level can be analyzed. Figure 6 presents the occupancy level of the bus line/ride sketched on the map (see Fig. 6a) in two moments of the day: a morning trip, from 09:26 to 10:40 (see Fig. 6b), and an evening trip, from 19:25 to 20:32 (see Fig. 6c). Observing the distribution over time, in which each bin corresponds to the people on-board at each stop (stops in outer areas are represented in blue, in the inner areas in red), it is possible to note that in the morning trip the bus progressively fills up by picking up people in outer areas, then starts to drop-offs passengers when reaching the inner areas. Conversely, an opposite behavior can be observed in the evening ride with passengers boarding at stops in the inner area and dropping off the bus in outer areas. Such results further align with the demand specified in input, therefore confirming the capability of the proposed solution to produce correct estimates on the transportation usage.

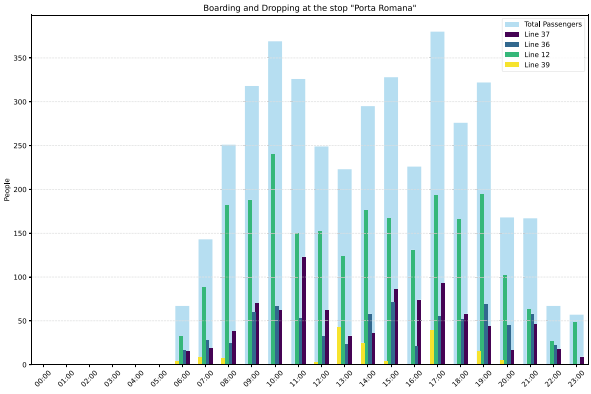
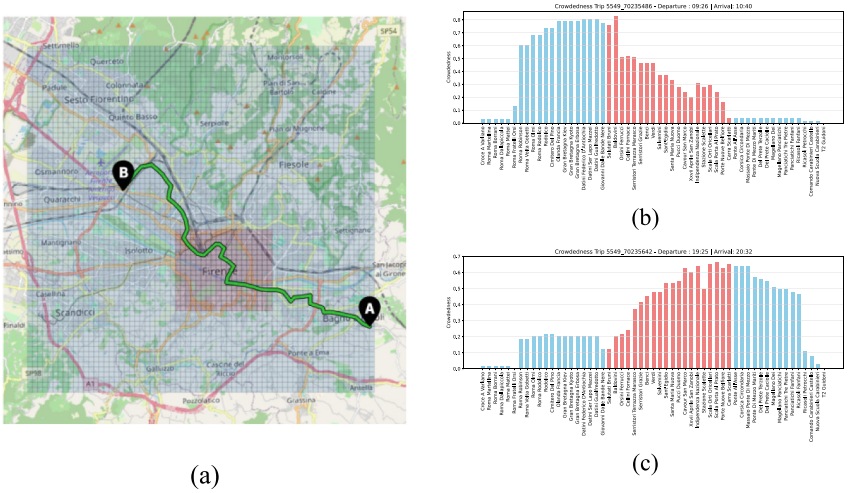


Fig. 5. Aggregated pick-ups and drop-offs at a bus stop with details on single lines.

3.1 What-if Analysis

To further show the effectiveness of our solution, hereafter an example of what-if analysis is presented. After running the simulation, we noticed that at a few stops, at specific time slots presented a huge number of people waiting for a bus and not being able to board since the passing vehicles were full. For example, on Monday at 17:39 the simulator listed 655 people waiting at Porta Romana bus stop to board bus 12 (on the ride running from 17:19 to 17:47).

We tried to add new rides on the same line to observe how these changes would impact the transportation service. First, we added a new ride operating in the same time window from 17:19 to 17:47. This in practice doubled the number of possible passengers



(a) (b) (c)

Fig. 6. Occupancy level for two rides on the same line. In (a) the bus line that travels from A to B has been reported over the map. In (b) the occupancy level for the morning ride (traveling from 09:26 to 10:40). In (c) the occupancy level for the evening route, from 19:25 to 20:32. Blue histogram bins indicate stops in the outer area, while reb bins are related to stops in in inner area.

Table 4. What-If experiments. New rides have been added in the simulation to observe the impact of the changes in reducing the number of people waiting at the inspected stop.

Experiment	Number of added rides	Added ride operating times	People waiting at the stop
Original	0		655
What-If #1	1	17:19–17:47	579
What-If #2	2	17:02–17:47	503
What-If #3	3	16:46–17:47	426
What-If #4	4	16:29–17:47	357

for the critical ride and produced a reduction in the number of people waiting passing from 655 to 579. Then we tried to introduce additional rides in previous windows. Results are shown in Table 4. As can be seen, by progressively adding new buses running on the same line the people waiting to be served progressively reduced passing from 655 to 579 with an additional ride, to 503 with two new rides, to 426 with three, to 357 adding four new rides. Note that a reduction of approximately 75 people for each new ride is observed, equal to the capacity set for the buses. This confirms that our solution is able to adapt properly to the input changes and can be effectively used to carry out what-if analysis in a very short time (since, as described above, by avoiding recomputing the routing the simulations terminate in less than 5 min). The solution resulted in being much faster than SUMO simulator in both setup and execution.

4 Conclusions

In this paper we have presented a novel solution to assess the match between transportation offer and city users mobility demand using agent-based simulations. Different from available state-of-the-art solutions, our method fulfills all the defined requirements, needs minimal input (i.e. GTFS and ODM describing respectively the offer and the demand, plus the city road graph), avoiding the need for any manual setup, and is able to provide quick full-day simulations that can run in a few minutes for huge area. Once the simulation is completed, a series of KPIs is produced to evaluate unmet demand, mean travel times, average waiting and walking times, number of pick-ups and drop-offs at specific stops, and the level of occupancy of any ride while traveling.

As shown in the experimental section, the proposed solution produces results that permit the actual simulation and react correctly to changes in the offer and demand. In addition, the solution has been capable of identifying the critical conditions, thus confirming its validity. Moreover, it can be easily used to perform what-if analysis to assess the impact of possible changes in the transportation service. Finally, since no manual input is required to configure a simulation and fast computation times are achieved, our solution can be effectively enforced into an automatic optimization framework exploiting for example reinforced learning to produce the most relevant solutions for solving transportation deficiencies strongly reducing the human effort, thus helping decision-makers in planning activities. The solution resulted in being much faster than SUMO simulator in both setup and execution.

Acknowledgements. This study has been carried out in the context of the Spoke 9 of the Italian National Center for Sustainable Mobility (CN MOST), and the CN MOST flagship project OPTIFaaS and scalability project SASUAM.

References

1. Kaeoruean, K., Phithakkitnukoon, S., Demissie, M.G., Kattan, L., Ratti, C.: Analysis of demand–supply gaps in public transit systems based on census and GTFS data: a case study of Calgary, Canada. Public Transp. **12**, 483–516 (2020)
2. SDG indicators (no date) United Nations. https://unstats.un.org/sdgs/report/2020/goal-11/. Accessed: 12 Feb 2025
3. Bellini, P., Bilotta, S., Collini, E., Fanfani, M., Nesi, P.: Data sources and models for integrated mobility and transport solutions. Sensors **24**(2), 441 (2024)
4. Adreani, L., Bellini, P., Fanfani, M., Nesi, P., Pantaleo, G.: Smart city digital twin framework for real-time multi-data integration and wide public distribution. IEEE Access **12**, 76277–76303 (2024). https://doi.org/10.1109/ACCESS.2024.3406795
5. Bellini, P., Collini, E., Fanfani, M., Palesi, L.A.I., Nesi, P.: Smart city digital twin platform architecture for mobility and transport decision support systems. In: 2024 IEEE International Conference on Big Data (BigData), pp. 5486–5495. IEEE (2024)
6. Shaharuddin, R.A., Misro, M.Y.: Traffic simulation using agent based modelling. In: AIP Conference Proceedings, vol. 2423, No. 1. AIP Publishing (202)
7. Bernhardt, K.:. Agent-based modeling in transportation. In: Artificial Intelligence in Transportation, 72 (2007)

8. Nguyen, J., Powers, S.T., Urquhart, N., Farrenkopf, T., Guckert, M.: An overview of agent-based traffic simulators. Transport. Res. Interdisc. Perspect. **12**, 100486 (2021)
9. Azfar, T., Weidner, J., Raheem, A., Ke, R., Cheu, R.L.: Efficient procedure of building university campus models for digital twin simulation. IEEE J. Radio Freq. Identif. **6**, 769–773 (2022)
10. Adnan, M., Pereira, F.C., Lima Azevedo, C.M., et al.: SimMobility: a multiscale integrated agent-based simulation platform. In: Transportation Research board 95th annual meeting, pp. 10–14. Washington, United States (2016)
11. Grigoryev, I.: AnyLogic in three days: Modeling and simulation textbook. The AnyLogic Company (2018)
12. Wilensky, U., Rand, W.: Modeling Natural, Social, and Engineered Complex Systems with NetLogo. The MIT Press (2015)
13. Taillandier, P., Gaudou, B., Grignard, A., et al.: Building, composing and experimenting complex spatial models with the GAMA platform. GeoInformatica **23**, 299–322 (2019). https://doi.org/10.1007/s10707-018-00339-6
14. Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D.: SUMO–simulation of urban mobility: an overview. In: Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation. ThinkMind (2011)
15. Horni, A., Nagel, K., Axhausen, K.W.: Introducing matsim. In: Horni, A., Nagel, K., Axhausen, K.W. (eds.) The multi-agent transport simulation MATSim, pp. 3–8. Ubiquity Press (2016). https://doi.org/10.5334/baw.1
16. Smith, L., Beckman, R., Baggerly, K.:TRANSIMS: Transportation analysis and simulation system. Los Alamos (1995)
17. Codeca, L., Erdmann, J., Cahill, V., Haerri, J.: Saga: An activity-based multi-modal mobility scenariogenerator for sumo. In: SUMO Conference Proceedings, vol. 1, pp. 39–58 (2020)
18. Ordóñez Medina, S.A.: Multi-day activity models: an extension of the Multi-Agent Transport Simulation (MATSim). Arbeitsberichte Verkehrs-Und Raumplanung, 1211 (2016)
19. Aimsun. (n.d.). Technical Notes Archives. Retrieved 14 Feb 2025. https://www.aimsun.com/category/technical-notes/
20. Ullah, M.R., Khattak, K.S., Khan, Z.H., Khan, M.A., Minallah, N., Khan, A.N.: Vehicular traffic simulation software: a systematic comparative analysis. Pakistan J. Eng. Technol. **4**(1), 66–78 (2021)
21. Ejercito, P.M., Nebrija, K.G.E., Feria, R.P., Lara-Figueroa, L.L.: Traffic simulation software review. In: 2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA), pp. 1–4. IEEE (2017)
22. Li, S., Azfar, T., Ke, R.: Chatsumo: Large language model for automating traffic scenario generation in simulation of urban mobility. IEEE Trans. Intell. Veh. (2024)
23. Arman, A., Badii, C., Bellini, P., Bilotta, S., Nesi, P., Paolucci, M.: Analyzing demand with respect to offer of mobility. Appl. Sci. **12**(18), 8982 (2022)
24. Alberti, F., et al.: Mobile mapping to support an integrated transport-territory modelling approach. Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci. **48**, 1–7 (2023)
25. Bellini, P., Bilotta, S., Collini, E., Fanfani, M., Nesi, P.: Mobility and transport data for city digital twin modeling and exploitation. In: 2023 IEEE International Smart Cities Conference (ISC2), pp. 1–7. IEEE (2023)
26. Adreani, L., Bellini, P., Fanfani, M., Nesi, P., Pantaleo, G.: Design and develop of a smart city digital twin with 3d representation and user interface for what-if analysis. In: Gervasi, O., et al. (eds.) Computational Science and Its Applications – ICCSA 2023 Workshops: Athens, Greece, July 3–6, 2023, Proceedings, Part VIII, pp. 531–548. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-37126-4_34

27. Bellini, P., Bilotta, S., Palesi, A.L.I., Nesi, P., Pantaleo, G.: Vehicular traffic flow reconstruction analysis to mitigate scenarios with large city changes. IEEE Access **10**, 131061–131075 (2022). https://doi.org/10.1109/ACCESS.2022.3229183
28. Fanfani, M., Palesi, L.A.I., Nesi, P.: Microservices' libraries enabling server-side business logic visual programming for digital twins. SoftwareX **27**, 101805 (2024). https://doi.org/10.1016/j.softx.2024.101805
29. General Transit Feed Specification. https://gtfs.org/. Accessed on 26th Feb 2025
30. Network Timetable Exchange. https://transmodel-cen.eu/index.php/netex/. Accessed on 26 Feb 2025)
31. Matrix Market Exchange Format. https://math.nist.gov/MatrixMarket/. Accessed 26 Feb 2025
32. Military Grid Reference System. https://mgrs-data.org/. Accessed 26 Feb2025
33. GraphHopper: Directions API Route Planning. https://www.graphhopper.com/. Accessed 26 Feb 2025
34. OpenStreetMap. https://www.openstreetmap.org/. Accessed on 26 Feb 2025
35. Bilotta, S., Nesi, P.: Traffic flow reconstruction by solving indeterminacy on traffic distribution at junctions. Futur. Gener. Comput. Syst. **114**, 649–660 (2021)