Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 855–860

Procedia Computer Science

www.elsevier.com/locate/procedia

The 6th International Workshop on Agent-based Mobility, Traffic and Transportation Models, Methodologies and Applications (ABMTrans 2017)

Integrating public transport into mobiTopp

Lars Briem[a,*], H. Sebastian Buck[a], Nicolai Mallig[a], Peter Vortisch[a], Ben Strasser[b], Dorothea Wagner[b], Tobias Zündorf[b]

[a] *Institute for Transport Studies, Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany*
[b] *Institute of Theoretical Informatics, Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany*

Abstract

Demand and supply are both relevant for travel time in public transport. While it is obvious that the supply side in form of the timetable corresponds directly to the travel time, the demand side influences the travel time only partially, but in critical moments. During peak hours, when the demand reaches the capacity of the vehicles, the interaction between demand and supply becomes important. Overcrowded vehicles, hindering passengers to catch their chosen route, lead to longer travel times. Therefore, it is important to integrate the supply side of public transport into a travel demand model.

The supply side of public transport has been integrated into the agent-based travel demand model mobiTopp. A timetable has been implemented, which is used for two purposes. First, it serves as input for the Connection Scan Algorithm, which is used by the agents to find the routes with earliest arrival time at their destinations. Second, it is used for the movement of the public transport vehicles. The model also contains capacity constraints for vehicles, which, when activated, result in a noticeable increase in travel time.

Keywords: public transport; travel demand model; agent based; public transport assignment; route search; mobiTopp; Connection Scan Algorithm

1. Introduction

Travel demand models are essential for the assessment of transport policy measures. They typically model the demand side of travel quite well. However, modelling only the demand side is not sufficient, as several of the networks' quality of service attributes, which enter as attributes into the demand model, like travel time, reliability, or comfort, depend on the interaction of the supply side and the demand side. The supply side is typically only modelled for the motorized traffic, since for motorized traffic an increased demand results directly in an increased travel time. For public transport, travel time is at first independent from the demand and only depends on the supply side, i. e. the timetable. However, during peak hours, when capacity is reached, the situation changes. An increasing demand firstly affects the comfort attribute, when there are no longer enough seats for all passengers. As the vehicle gets more and

* Corresponding author. Tel.: +49-721-608-47772 ; fax: +49-721-608-46777.
E-mail address: lars.briem@kit.edu

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.05.401

more crowded, the available space per passenger shrinks, creating more and more discomfort. With an increasing number of passengers standing in the aisles, entering and leaving the vehicle slows down, possibly creating additional delays of the vehicle and as such reducing reliability. In case of an overcrowded vehicle, it is even no longer possible for some public transport users to board. They have to wait for the next vehicle to arrive or even choose a new connection. So in this case, the demand affects the travel time for public transport.

Agent-based models offer a convenient way to integrate the supply side of public transport. We describe here, how the supply-side of public transport has been integrated into mobiTopp. The results show, how taking the capacity of public transport vehicles into account affects the resulting travel times.

2. Related work

In recent years, the supply side of public transport has been implemented in some agent-based models. One of the first implementations was done for MATSim in 2010. Rieser implemented a basic model covering the timetable of one day with several lines, stops and vehicles[1]. A similar work was done for SUMO also in 2010, showing that it is possible to integrate intermodality into SUMO by implementing public transport[2]. POLARIS currently does not have an implementation of the supply side of public transport. But it is planned to model public transport for emergency scenarios, where it is necessary to guide drivers and optimize the routes to save as much lives as possible[3].

Long before agent-based models were as common as today, public transport has also been modelled in macroscopic models. One of those models is VISUM from PTV AG. It is one of the larger public transport models, covering a lot of corner cases. As it is used by public transport companies to plan their operations, it also includes fare rules and costs for vehicles to calculate the overall costs for all operators[4]. This model and the model defined by the Association of German Transport Companies (VDV)[5] can be seen as industry standards.

Some of the macroscopic models, like VISUM, also use vehicle capacities during transit assignment. Simulation experiments by Nuzzolo et al. show that capacity constraints have a great influence on transit assignment. In these experiments a large number of passengers change their departure time based on experience with overcrowded vehicles at earlier days[6]. As Nuzzolo et al. use travel demand aggregated over a period of 5 minutes for the analysis, an agent-based approach may provide a more detailed demand and therefore may provide a more detailed analysis base.

3. The mobiTopp model

mobiTopp[7] models every person, household, and car of the planning area. Persons are represented as agents who make their decisions autonomously, individually, situation-dependent, or based on interactions with other agents[8]. Their decisions are based on activity programs, which they process over a period of one week. During the execution of such an activity program, an agent chooses the locations for his activities and the mode used to reach each destination. Both types of decision are based on Discrete Choice models.

As there is no traffic assignment currently implemented in mobiTopp, cars do not interact with each other. Only agents can interact with cars of their household and use them for their trips. Cars currently in use are not available to others for the mode *car as driver*. However, the car passenger extension allows to travel together in the same car using the mode *car as passenger* as long as there are free seats and the source and destination of both agents match[9].

mobiTopp consists of two stages: initialisation and simulation. During initialisation long term aspects of the whole system and long-term decisions of the persons and households are modelled. These are population synthesis, assignment of fixed activity locations like work and home locations, assignment of private cars and travel cards.

Based on these long term decisions, the simulation covers the short term behaviour of each agent. During this stage all agents are simultaneously simulated for one week. While processing the activity program of each agent, the agent traverses states. Most of the time an agent switches between the states *execute activity* and *make trip*. Both states take some time and a change to another state is triggered at the end of an activity or a trip. Each state change is associated with additional behaviour. For example, after a trip has been finished, the driver returns the car.

States as described here have different types. The states *execute activity* and *make trip* are so-called duration states. An agent remains inside these states for a longer time. All other states are instantaneous states. In these states, the agent executes a specified behaviour and switches to the next state without consuming time. All states are described in Figure 1.

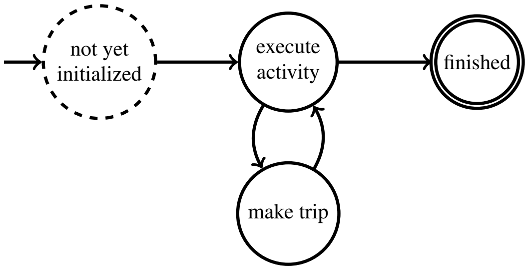

Fig. 1: State diagram of an agent in mobiTopp.[9] Duration states are marked with solid lines and instantaneous states are marked with dashed lines. During the simulation, the agent alternates between the states *execute activity* and *make trip*.

4. Public transport in mobiTopp

The behaviour of persons in public transport is quite different from travelling in other modes like car or bike. In public transport networks, persons can only board or exit a vehicle at specific stations. Therefore, they need the possibility to search a route from their current location to their destination, maybe using several vehicles on fixed routes instead of only one vehicle. In the past several routing algorithms have been developed to solve this problem.[10,11,12] Instead of developing a new algorithm, we decided to choose one that fits our needs. As the routing algorithm will be used heavily during simulation, the speed of a single route search requests is quite important. The Connection Scan Algorithm is one of the fastest algorithms currently available[13] and was hence selected.

4.1. Connection Scan Algorithm

The Connection Scan Algorithm (CSA) is based on two parts. First, an initialisation takes place where data structures are setup and connections are sorted by their departure and arrival time. The second part is a linear sweep over all connections, updating arrival times at stops. An arrival time is updated if a connection is reachable. Reachable means that the earliest arrival time at the departing stop of the connection is earlier or equal to the departure time, taking necessary transfer times into account. Transfer times are only relevant, if the vehicle (trip) of the connection has not already been used. If the arrival time is updated, all footpaths of this stop are used to update the arrival time at the other stop of each footpath[12,14]. The algorithm is described in pseudocode in Listing 1.

Algorithm 1 Connection Scan Algorithm, considering footpaths between stops, based on Dibbelt et al.[12].

function CSA(source s_s, target s_t, departure τ_{dep}, connections C, footpaths F)
    $\tau^*(\cdot) \leftarrow \infty$ ▹ Initialise earliest arrival at all stops
    $\tau^*(s_s) \leftarrow \tau_{dep}$ ▹ Initialise arrival at source
    $\tau^*(s_t) \leftarrow \tau_f(s_s, s_t)$ ▹ Initialise arrival at target
    $T \leftarrow \emptyset$ ▹ Initialise used trips
    sort C

    for all $c \in C$ **do**
        if $\tau_{dep}(c) \geq \tau^*(s_s(c)) + \tau_{transfer}(s_s(c))$ **or** $c_t \in T$ **then** ▹ If connection is reachable or trip has been used
            if $\tau_{arr}(c) < \tau^*(s_t(c))$ **then** ▹ and arrival time is earlier
                $\tau^*(s_t) \leftarrow \tau_{arr}(s_t(c))$ ▹ update earliest arrival time
                add c_t to T
                for all $(s_t(c), s') \in F$ **do** ▹ Relax all footpaths of current stop
                    if $\tau_{arr}(s_t(c)) + \tau_f(s_t(c), s') < \tau^*(s')$ **then**
                        $\tau^*(s') \leftarrow \tau_{arr}(s_t(c)) + \tau_f(s_t(c), s')$ ▹ arrival time + footpath
                    end if
                end for
            end if
        end if
    end for
    return $(s_s, s_t, \tau_{dep}, \tau_{arr}(s_t))$
end function

Starting with the connection scan, the algorithm needs some input data to work on. The integration of public transport into the simulation requires a timetable, which provides every information needed by the route search algorithm. Based on this timetable, vehicles can be scheduled for the whole simulation period.

4.2. Timetable

The definition of the data structures for the timetable starts with a minimal model, which is extended as necessary. The simplest model to find routes in a network consists of *stops* and *connections*. A stop is a location where agents can wait for, board, or exit a vehicle. A connection is a direct link between two stops with a given departure and arrival time. When transferring this into graph theory, stops correspond to nodes and connections to edges. Based on this, every graph based algorithm can be used for routing.

Using only connections and stops is not enough to find routes. Situations exist where agents have to transfer from one stop to another. Therefore, transfers are modelled as footpaths in the neighbourhood of each stop. A neighbourhood contains all stops that are reachable from a stop within a given period of time, e.g. 15 minutes. Neighbourhoods are explicitly not modelled as stations, since there can be transfers between different stations. Stations on the other side are also modelled, basically for analysing aggregates of stops.

To take transfer times into account during route search, the algorithm also needs information when transfer times have to be considered. Therefore, all connections served by the same vehicle are grouped into a journey. Here, we combine the modelling of a vehicle and a journey and add the capacity of the vehicle directly to the journey.

Additional to the components needed for routing, we specify several others for easier conversion from other models and for visualisation. These are time profiles, transport systems and route points. Time profiles describe departure and arrival times at stops. Times are given relative to the first stop. The departure time at the first stop is always zero. Combining a time profile with an absolute departure time, including a date, results in a journey and the corresponding connections. Route points represent the geographic route of a journey. They are currently only needed for visualisation. A complete overview of the model is given in Figure 2.

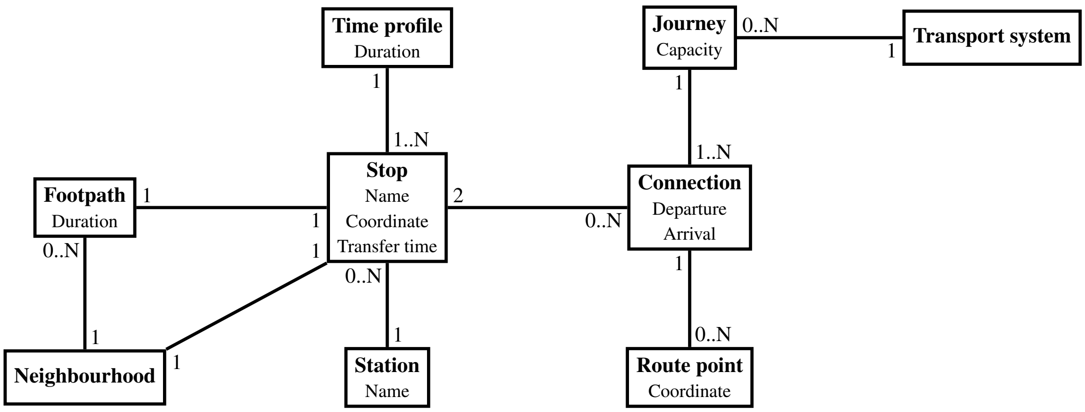

Fig. 2: Complete overview of the timetable in mobiTopp

4.3. Behaviour

The original behaviour of agents in mobiTopp, as shown in Figure 1, is no longer sufficient for the detailed modelling of public transport. We replace the *make trip* state with a more detailed behaviour, in case the agent uses public transport. Instead of *make trip*, an agent switches to a system of states after an activity, which starts with the state *use public transport* and finishes with the state *leave public transport*.

In *use public transport* the agent searches his next route and walks to the first stop of his route. Currently, the agent always uses the route with the earliest arrival time at his destination. Arriving at his start stop, the agent *waits for the vehicle* he wants to board. If it is available, the agent *tries to board* it. While trying to board, the vehicle accepts or rejects the boarding attempt, based on the available remaining capacity. If the agent is allowed to board the vehicle, he starts to ride the vehicle (*ride vehicle*). Contrarily, if the agent is not allowed to board the vehicle, he has to search a new route and wait for the first vehicle of the new route.

The agent stays inside the vehicle until he arrives at his target stop or the next transfer stop. In both cases the agent leaves the vehicle. In case he reaches his target stop, the agent walks to his destination and *leaves public transport*. If not, the agent searches for the next vehicle or walks to the stop where his next vehicle departs. *Wait for vehicle* and *ride vehicle* are modelled as duration states, like *make trip* in the original model. Figure 3 shows all states of an agent and the corresponding state changes.

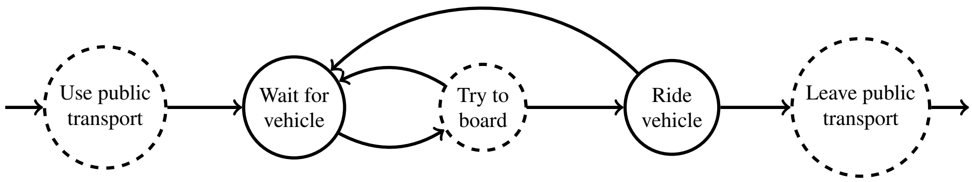

Fig. 3: Visualisation of state changes for a person using public transport. *Wait for vehicle* and *ride vehicle* are duration states (solid border) in which an agent stays for a longer time. The other states are instantaneous (dashed border) and are used to encapsulate specified behaviour.

5. Application of the model

The implementation of public transport in mobiTopp, which has been calibrated based on the data of a recent household travel survey [15], has been successfully applied to the region of Stuttgart. The region contains the city of Stuttgart and the surrounding districts. All 2.7 million inhabitants of the study area are simulated over a period of one week. The input data used to construct the timetable has been taken from a given macroscopic model [16]. The resulting timetable contains 785 118 connections, 48 100 journeys, and 24 different vehicle types, which serve 13 941 stops. The stops are combined to 11 410 stations. Between stops exist 48 236 bidirectional footpaths, which have been read from the macroscopic model and have been extended by a footpath search to find all stops within 15 minutes walk time around each stop.

Using the presented model in mobiTopp results in about 1 million trips made by almost 620 000 agents. One simulation took about 48 hours to simulate a single day of travel demand on an Intel Core i7 3820 (3,6 GHz) using 24 GB of RAM. A mobiTopp run without public transport assignment takes for the same simulation period about 3 hours. The increased runtime is mainly the result of the large number of route search requests.

Travel times are distributed over a wide range, but a huge number of trips take less than one hour and most of the trips are faster than one hour and a half. Figure 4 shows the distribution of travel times in public transport, with and without capacity constraints. The results show that the output of the simulation run with the constrained capacity contain fewer trips with shorter duration and more trips with longer duration than the output of the simulation run with unconstrained capacity.

6. Conclusion and future work

We have shown how the supply side of public transport can be integrated into an agent-based model of travel demand. The resulting implementation provides the potential to account for discomfort in the mode choice model and to integrate delays resulting from prolonged boarding times due to overcrowded vehicles. For these use cases, additional empirical data is needed. The current implementation allows already to analyse the effect of vehicle capacities on the simulation results and hence allows to assess the potential bias of models neglecting vehicle capacities.

The presented approach follows the modular fashion and basic principle of mobiTopp, using simple models where possible, but providing the possibility to replace them with more sophisticated models when needed. In this case, such a simple model is the route choice behaviour. Currently, the route with the earliest arrival is chosen. An improved implementation could present a set of alternatives to the agent, differing for example by arrival time, travel time, and number of transfers. The agent could then choose from this set of alternatives, using a Discrete Choice model. Looking further, we can also integrate the agents experience about empty or crowded vehicles into his decision.

The runtime of the current public transport implementation is a multiple of the runtime without it, mostly as a result of the route search. So it is necessary to increase the performance of the route search significantly. Nevertheless, the

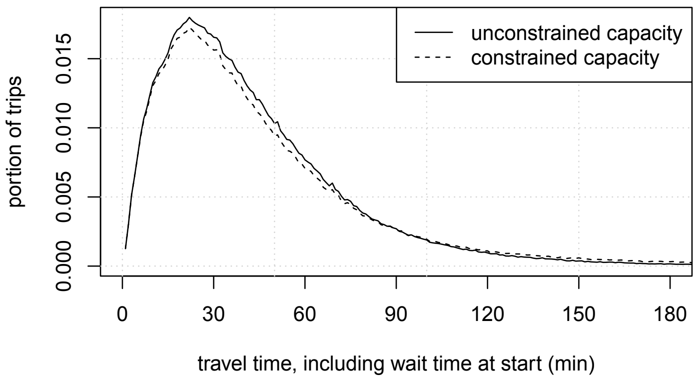

Fig. 4: Distribution of travel times in public transport in mobiTopp, with and without capacity constraints. The travel time is measured from the arrival of an agent at the first stop to the exit at the last stop.

presented work, integrating the supply side of public transport into mobiTopp, including the capacity constraints on single vehicles, provides a solid foundation for further extensions.

Acknowledgement

This work has been supported by grant FOR 2083 from Deutsche Forschungsgemeinschaft (DFG).

References

1. Rieser M. Adding transit to an agent-based transportation simulation: Concepts and implementation. Ph.D. thesis; Technischen Universität Berlin; Berlin; 2010.
2. Behrisch M, Erdmann J, Krajzewicz D. Adding intermodality to the microscopic simulation package sumo. In: The Middle Eastern Simulation and Modelling Conference. 2010,.
3. Sokolov V, Auld J, Ley H, Boltonc M. Coordinated transit response planning and operations support tools for mitigating impacts of all-hazard emergency events. In: World Conference on Transport Research. Shanghai; 2016,.
4. PTV AG . PTV Visum 15 – Manual. PTV AG; 2015.
5. Verband Deutscher Verkehrsunternehmen (VDV) . VDV-Standardschnittstelle Liniennetz/Fahrplan. 2013.
6. Nuzzolo A, Crisalli U, Rosati L. A schedule-based assignment model with explicit capacity constraints for congested transit networks. Transportation Research Part C: Emerging Technologies 2012;20(1):16–33.
7. Mallig N, Kagerbauer M, Vortisch P. mobitopp – a modular agent-based travel demand modelling framework. Procedia Computer Science 2013;19:854–9.
8. Bonabeau E. Agent-based modeling: Methods and techniques for simulating human systems. Proceedings of the National Academy of Sciences (PNAS) 2002;99(suppl. 3):7280–7.
9. Mallig N, Vortisch P. Modeling car passenger trips in mobitopp. Procedia Computer Science 2015;52:938–43.
10. Schulz F, Wagner D, Weihe K. Dijkstra's algorithm on-line: an empirical case study from public railroad transport. Journal of Experimental Algorithmics (JEA) 2000;5:12.
11. Delling D, Pajor T, Werneck RF. Round-based public transit routing. Transportation Science 2015;49(3):591–604.
12. Dibbelt J, Pajor T, Strasser B, Wagner D. Intriguingly simple and fast transit routing. In: International Symposium on Experimental Algorithms. Springer. ISBN 978-3-642-38527-8; 2013, p. 43–54.
13. Bast H, Delling D, Goldberg AV, Müller–Hannemann M, Pajor T, Sanders P, et al. Route Planning in Transportation Networks. In: Kliemann L, Sanders P, editors. Algorithm Engineering - Selected Results and Surveys; vol. 9220 of *Lecture Notes in Computer Science*. Springer; 2016, p. 19–80.
14. Strasser B, Wagner D. Connection Scan Accelerated. In: McGeoch CC, Meyer U, editors. Proceedings of the 16th Meeting on Algorithm Engineering and Experiments (ALENEX'14). SIAM; 2014, p. 125–37.
15. Verband Region Stuttgart . Mobilität und Verkehr in der Region Stuttgart 2009/2010 – Regionale Haushaltsbefragung zum Verkehrsverhalten. 2011.
16. Schlaich J, Heidl U, Pohlner R. Verkehrsmodellierung für die Region Stuttgart – Schlussbericht; 2011.