

A microscopic simulation approach for optimization of taxi services

Michał Maciejewski^{1,2}, Kai Nagel²

¹Poznan University of Technology

²Berlin Institute of Technology

Abstract

This paper presents a simulation platform along with several on-line dispatching algorithms developed in order to optimize taxi services. First, the issue of simulation-based optimization of modern transport services, especially taxi services, is presented. Next, the proposed approach to microscopically simulate taxi services is explained, followed by a description of the on-line taxi dispatching algorithm framework and three selected dispatching strategies implemented within this framework. The next section presents the simulation scenario of Mielec that the strategies were tested on. Then, the simulation results obtained are analysed and the strategies compared. The paper ends with conclusions on the main properties and other possible applications of the proposed simulation approach, as well as on future plans concerning further improvements of the taxi dispatching algorithms.

Keywords: dynamic taxi dispatching, dynamic vehicle routing, demand-responsive transport, on-line optimization, simulation-based optimization, multi-agent simulation, MATSim, traffic flow simulation

1 Introduction

As a result of the development of Information and Communications Technology (ICT), transport services have become more intelligent, that is more flexible, demand-responsive, safe and energy/cost-efficient. This concerns both substantial improvements in the traditional means of transport, such as regular public transport or taxis, as well as the introduction of novel services, such as demand-responsive transport or personal rapid transit. However, the growing complexity of modern transport systems, besides all the benefits, increases the risk of failure due to the lack of precise design, implementation and testing. One way of dealing with this problem is the use of simulation tools that offer a wide spectrum of possibilities for validating transport service models. Considering the complexity of modern transport

systems, such a simulation tool has to reflect realistic demand dynamism, traffic flow phenomena and fleet management operations.

For over a decade, different simulators combining vehicle routing and traffic simulation have been created, such as a simulation framework for the evaluation of dynamic fleet management systems for truckload carrier operations [Reg98], an AIMSUN extension for optimization of the VRP in cities [Bar], or a system for the simulation-based evaluation of DVRP strategies using real-time information [Lia08].

Concerning the taxi services, the simulation approach has to take into account high variability of both traffic flow and demand. Therefore, the use of microscopic traffic flow simulation combined with microscopic travel demand models (e.g. activity-based) is crucial for accurate evaluation of the service under different circumstances, ranging from low to high load (e.g. large events, bad weather or public transport strikes). These issues, however, remain almost unexplored. To the best knowledge of the authors, traffic flow simulators have been applied only in Singapore; Paramics have been used to simulate taxi dispatching in Singapore's Central Business District [Lee04], and MITSIMLab for a 15 km x 10 km urban area [Seo10]. All these approaches do not include realistic on-line demand generation; one cannot, for instance, model the impact of traffic or transport service availability on customer demand. Moreover, the systems were used only for small-scale problems, where a road network was of limited size and the number of customers was not high.

2 Microscopic Simulation of Taxi Services

As stated in Section 1, in the case of taxi services, a comprehensive approach should allow for running large-scale simulations with microscopically modelled demand, supply and traffic. Out of various simulation platforms considered, MATSim [Bal08] is arguably the one that is closest to meet all the requirements stated; see [Mac12] and references therein for additional justification. MATSim, however, does not provide taxi among the built-in means of transport, and therefore, has been extended and coupled with the DVRP Optimizer [Mac12] that is responsible for dispatching taxis. Whenever a new event occurs (such as a request submission, a vehicle departure/arrival) during the simulation in MATSim, the DVRP Optimizer reacts and adapts taxi drivers' schedules to the current state. Each taxi driver follows his/her schedule that consists of tasks of the following types:

- `DriveTask` – driving along a given route (usually the shortest path between two points in the network).
- `ServeTask` – picking up a passenger at a given location (includes waiting for the passenger).
- `WaitTask` – waiting at a given location for a new request.

A typical sequence of events associated with a single request is presented in Figure 1. A taxi customer calls the taxi service (request i ; event \mathbf{E}_i^0 at time T_i^0) and waits until the taxi

arrives. In response, the taxi dispatcher assigns the new request to one of taxis, according to a predefined algorithm (see Section 3). The selected taxi sets off for the customer (\mathbf{E}_i^1, T_i^1) and arrives at the pick-up location (\mathbf{E}_i^2, T_i^2). Then the customer is picked up and the taxi departs (\mathbf{E}_i^3, T_i^3). Finally, the taxi drops off the passenger at the destination location (\mathbf{E}_i^4, T_i^4). Since taxi demand and traffic are stochastic, times T_i^1, T_i^2, T_i^3 and T_i^4 are subject to change during simulation.

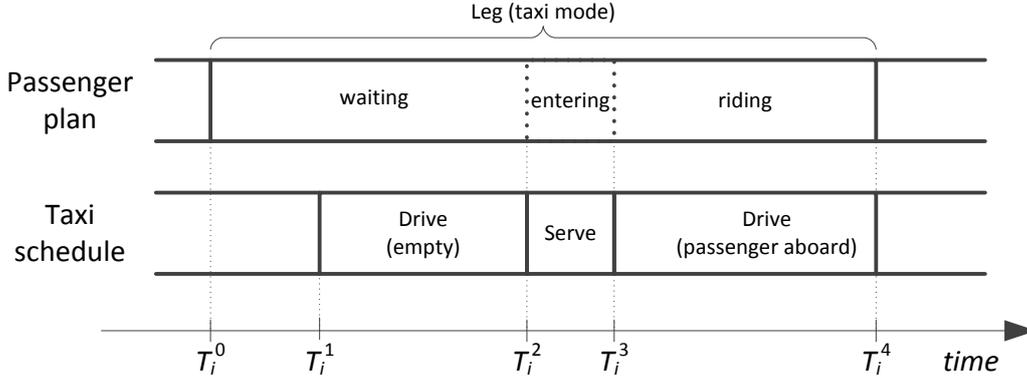


Figure 1: A planned taxi leg and the corresponding sequence of taxi tasks.

The traffic flow simulation in MATSim is based on queue approach, where each link is a FIFO queue and constitutes the basic network element. Vehicles enter and leave the network at the end of links. As they traverse the network, they move from the current link to the next one along their paths. Therefore, the accuracy of the vehicle's position is constrained to a single link. The DVRP Optimizer operates on a directed graph where arcs are the shortest paths between a given pair of locations (i.e. links). As link travel times change over day, arcs are time dependent, i.e. their travel times and shortest paths (i.e. sequences of links) depend on departure time. By default, it is assumed that both link and arc travel times are calculated with the accuracy of 15 minutes, which is enough to reliably model the dynamics of traffic flow, and at the same time, limits the amount of shortest path searches (arc travel times and paths are cached for each time bin).

3 On-line Taxi Dispatching

Some studies propose the use of multi-agent approach to model (partial) independence of taxi drivers [Che11; Seo10; Als09], while other assume a fully centralised management [Lee04; Wan09]. All on-line taxi dispatching algorithms currently available in the DVRP Optimizer implement a certain pattern of collaboration between customers, taxi drivers and the dispatcher. Some algorithms are customizable and offer a choice between several patterns. A collaboration pattern is defined by the following set of properties:

- *destination knowledge* – the destination is known a priori if a customer informs the dispatcher about his/her destination.

-
- *request and taxi monitoring* – the dispatcher may monitor taxis and constantly update the timing of their schedules. Otherwise, taxi drivers notify the dispatcher only about switching between the *busy* and *idle* states.
 - *requests reassignment* – already assigned requests can be dynamically reassigned between drivers. Request swapping is expected to be beneficial for both customers and drivers, and is usually coordinated by the dispatcher.

Each option implies some cooperation between interested parties. The first one involves additional customer-to-dispatcher communication, the second one imposes extra driver-to-dispatcher communication, while the last one requires real-time collaboration between drivers and the dispatcher.

Taxi fleets operate in a very dynamic environment, where demand, supply and traffic are stochastic and to some extent unknown. On-line taxi dispatching algorithms have to react to events representing changes in the system. In the simplest approach, commonly used by taxi companies, the dispatching procedure are performed only in response to submissions and completions of requests (events E_i^0 and E_i^4 , respectively). More sophisticated algorithms may be triggered also when taxis set off for, arrive at and depart from pick-up locations (events E_i^1 , E_i^2 and E_i^3 , respectively). Additionally, all taxicabs can be monitored on-line and the dispatching procedure may be executed when an ongoing *DriveTask* is expected to end ahead of or behind schedule.

In this paper, three dispatching strategies are analysed: *no-scheduling* that mimics the simplest approach, *re-scheduling* that monitors vehicles and responds to all possible events, and a modification of the latter that minimizes pick-up trip times instead of waiting times. All three strategies are based on the following assumptions:

- Dispatching procedures are fast enough to operate on a static snapshot of the current system state.
- Customers perform only immediate taxi calls and then wait for a taxi to come. To assure fairness, all taxi requests are scheduled based on the *first-come, first-served* policy.
- By default, *nearest taxi* means the nearest one in *time*. However, the first and third strategies may also use any measures of closeness in *space*.
- Although the second and third strategies may take advantage of the a priori destination knowledge [Mac13b], this aspect is beyond the scope of this paper – it is assumed that the destination remains unknown to the dispatcher until time T_i^3 .

No-scheduling strategy (NOS) This strategy responds to E_i^0 and E_i^4 events in the following way:

- E_i^0 – the nearest vehicle among the idle ones is dispatched to this request; if no vehicle is available at that time, the request is queued in the FIFO queue.

- E_i^4 – the vehicle that has just completed request i is dispatched to the first request in the FIFO queue; if the queue is empty, the vehicle becomes idle.

Despite its simplicity, this strategy has several advantages over more elaborate ones. It has low demand for computational power. Additionally, it does not require travel times to be known since it does not build schedules; one can even use straight-line distance to find the nearest idle taxi. Among the drawbacks is performance deterioration with the decreasing number of idle taxis — in an overloaded system, the first idle taxi may appear on the opposite side of a city.

Re-scheduling strategy (RES) This strategy extends the existing taxi schedules by appending a new request to the schedule of the nearest (in time) vehicle among all vehicles (both idle or busy), which requires the knowledge of travel times. This strategy monitors execution of requests and movement of vehicles. In response to some delays (or speedups), it updates the timelines of schedules and reassigns requests between taxis if the other one appears to be nearer. The strategy acts in the following way:

- E_i^0 – request i is appended to the schedule of the *nearest* taxi
- $E_i^1 - E_i^4$ – if the vehicle serving request i is ahead of/behind time, full rescheduling is carried out (all planned requests are removed from schedules and scheduling is performed again according to the FCFS rule)
- link-to-link moves – if the vehicle is ahead of/behind time, the timing of its schedule is updated, while the assignments remain unchanged

This strategy considers all vehicles, both idle and busy, which increases the chances of finding better assignments. However, as destinations are unknown, the planning horizon is limited up to one pickup ahead (event E_i^3), and therefore, vehicles with already one planned pickup cannot be considered when scheduling a new request (before arriving at the pick-up location). Frequent reassignments cause higher demand for computational power, compared to NOS.

Re-scheduling strategy for minimizing pick-up trip times (RES-PT) This strategy is a slightly modified RES, where the measure of closeness does not represent passenger's waiting times ($T_i^2 - T_i^0$) but pick-up trip times ($T_i^2 - T_i^1$), or other arbitrary pick-up trip distance measures. This modification shifts the preference from customers to taxi drivers. In an overloaded system, however, the reduction of pick-up trip times increases the system throughput, and hence, reduces the amount of time passengers wait for a taxi.

4 Test Scenario

In order to evaluate different strategies a simulation scenario was created in MATSim. The scenario represented a hypothetical private car traffic in the city of Mielec (south-eastern

Poland, over 61,000 inhabitants) between 6:00 am and 8:00 pm, including both peak hours. The network consisted of 200 nodes and over 600 links and traffic was made up of over 56,000 private car trips. Detailed description of the model can be found in [Mac13a].

First, the simulation of Mielec was carried out for 20 regular iterations without taxis, which was enough for the relaxation process to converge. Next, a given fraction of intracity private car trips were changed into taxi trips. Since such a conversion introduces extra traffic related to pick-up trips, the travel times increased. Additionally, the original private car paths may differ from the derived taxi drop-off paths, so the geographical layout of traffic may change. Therefore, before benchmarking each algorithm, five fully-functional (i.e. with taxis) warm-up iterations were carried out, which helped to obtain correct travel times. Finally, 20 benchmark iterations were executed.

The performance of the proposed optimization strategies was tested against different amounts of demand and supply. The taxi demand was modelled as $n = 406, 636, 840, 1069, 1297, 1506$ and 1719 requests, which corresponds to approximately 1, 1.5, 2, 2.5, 3, 3.5 and 4 per cent of private car trips converted into taxi ones. The spatio-temporal distribution of the taxi demand was identical to the regular traffic, as a result, the rush hours were the most challenging for taxi dispatching. On the supply side, the size of the fleet m was set to 25 and 50, and it did not change during a simulation period.

5 Simulation Results

Many different measures, all described in [Mac13a], were used to assess the performance of the proposed strategies, both from the taxi customers' and taxi company's points of view. In this paper, the following two measures are analysed: average passenger waiting time, $T_W = \sum_{i \in N} (T_i^2 - T_i^0)/n$, and average pick-up trip time, $T_P = \sum_{i \in N} (T_i^2 - T_i^1)/n$, where the former represents the customers' perspective and the latter defines the interest of the company. One may say that NOS and RES minimize T_W whereas RES-PT minimizes T_P . Figures 2 and 3 present the results obtained for the Mielec scenario and different n and m values. Separate curves were plotted for $m = 25$ (n/m between 16.24 and 68.76) and $m = 50$ (n/m between 8.12 and 34.38). All algorithms were run with time as the measure of taxi-to-request closeness. The results are averages over 20 benchmark iterations.

Figure 2 shows that neither of the strategies turned out superior for all demand-to-supply ratios. At low load, up to $n/m \approx 30$, NOS performs best while RES is slightly worse. At medium load, n/m between 30 and 55, RES is definitely the best performing strategy. However, at high load, above 55 requests per cab, RES-PT yields lowest T_W .

At first sight, the most curious is the advantage of NOS over RES at low load. One would expect that RES, having a broader choice set (idle and busy taxis) and using exactly the same travel time data, should outperform the former. That would happen in a scenario with uniformly distributed origin and destination locations, which is not true in the Mielec scenario, where people leave homes in the morning and return there in the evening. As a result, one can imagine a situation presented in Figure 4, where pick-up and drop-off

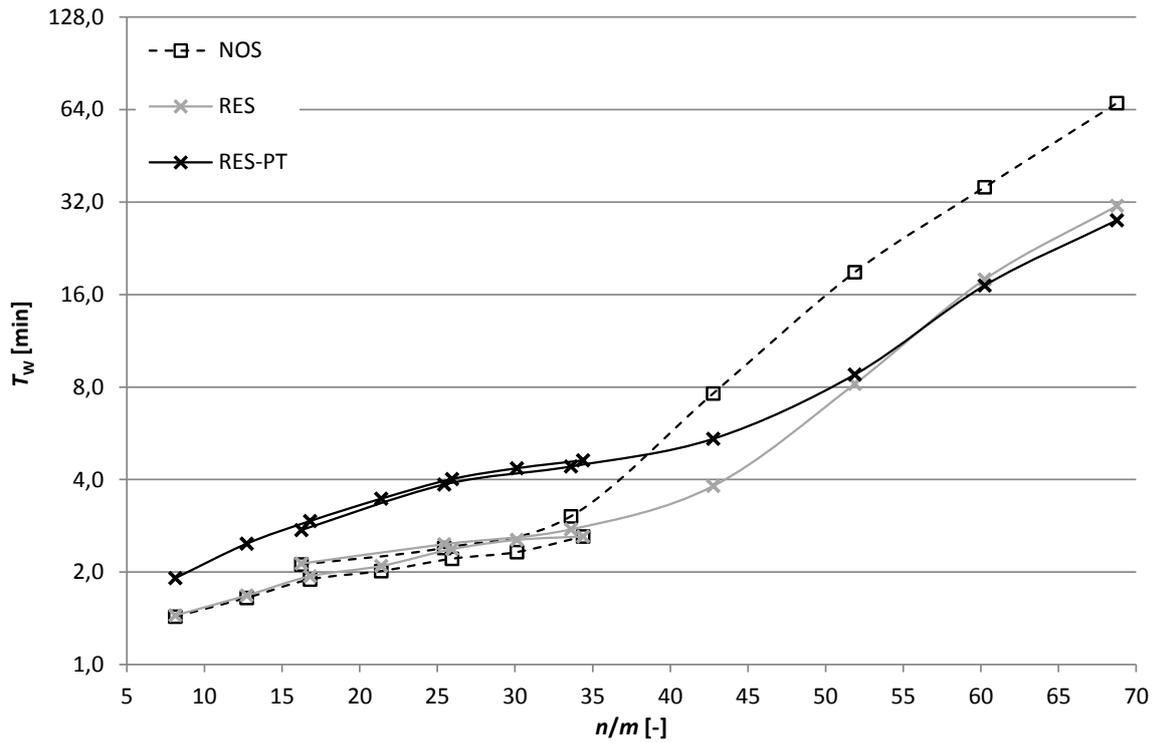


Figure 2: Average passenger waiting time T_w at different demand-supply ratios.

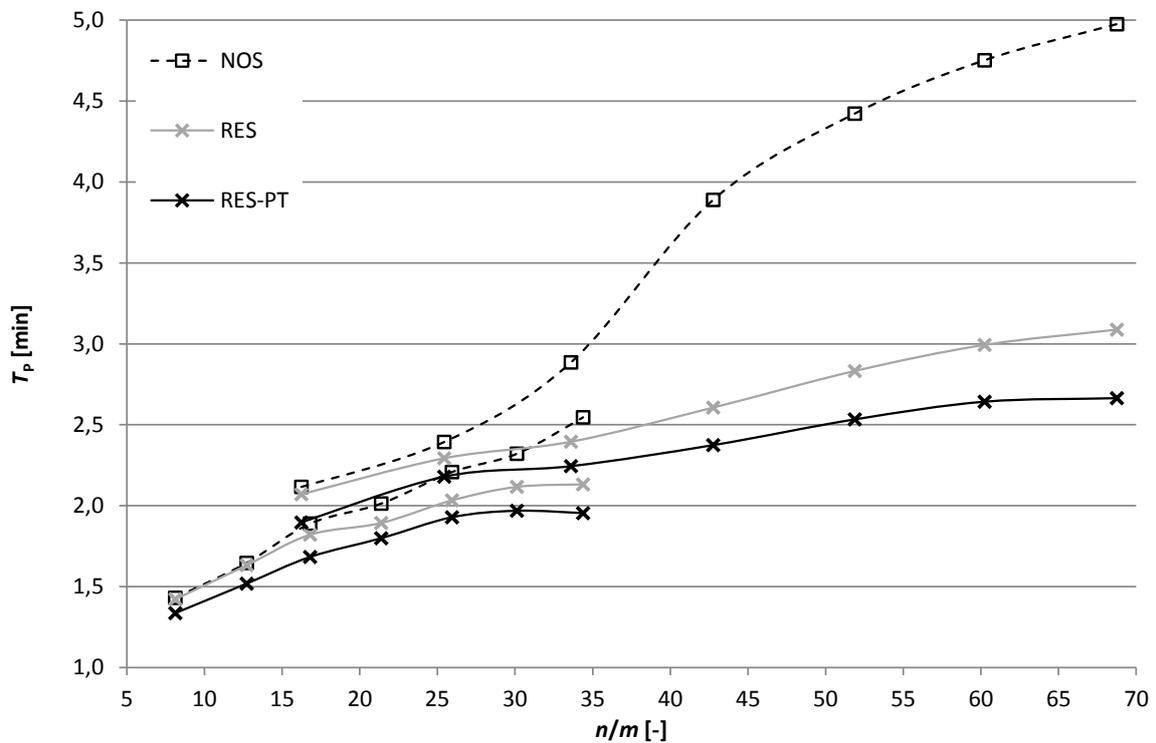


Figure 3: Average pick-up trip time T_p at different demand-supply ratios.

locations are concentrated in two different parts of the city. Initially, request 1 is already submitted and cab 1 is idle. Soon request 2 will be submitted and cab 2 will become idle. In such a situation, NOS would assign request 1 to cab 1 and then request 2 to cab 2, while RES would do the opposite (assuming that cab 2 is *closer in time* to request 1 than cab 1). The decision made by RES is better from the perspective of request 1, which has priority over request 2. However, the opposite would be better for the overall minimization of T_W , as it would increase slightly the waiting time of request 1 and, at the same time, reduce it significantly for request 2.

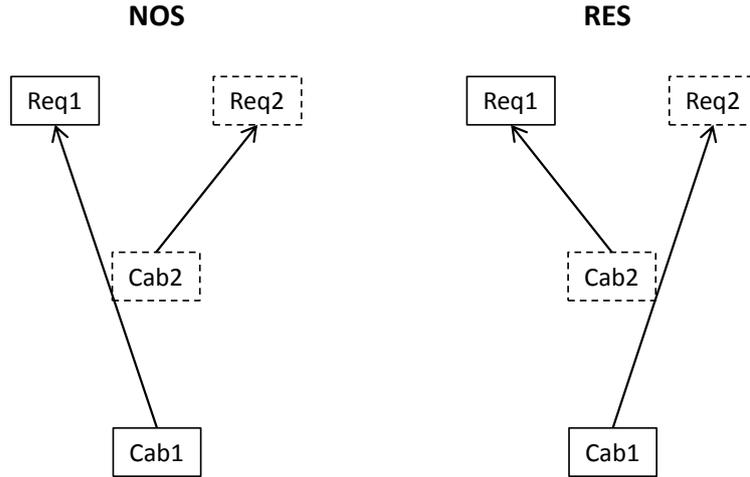


Figure 4: A comparison of the operation of NOS and RES.

As expected (see Section 3), RES-PT outperforms RES at high load. By minimizing T_P , instead of T_W , taxis spend less time on pick-up rides, which, in turn, increases the system throughput. Eventually, higher throughput means lower T_W .

Looking at Figure 3, the results there seem very plausible. RES-PT gives the lowest values since the main aim of this strategy is the minimization of the pick-up trip times. RES is better than NOS, as the possibility of choosing a busy taxi reduces the pick-up trip times — not only does it help in minimization of T_i^2 , but also it allows for higher T_i^1 , which eventually results in lower T_P . At low load, where almost all taxis are idle, NOS and RES perform similarly. This changes as demand rises and the number of idle taxis drops, which narrows the choice of taxis in NOS.

There is an interesting relation between 25- and 50-taxi series in Figure 3. They are not adjacent and we obtain higher values for smaller fleets. This is due to the fact that the average distance to the closest taxi drops with the growing number of taxis. The same pattern occurs in Figure 2 for NOS, and partially for RES, however, not for the RES-PT series. This is caused by the fact that minimization of T_W (NOS and RES) requires T_P to be small as well. Additionally, NOS results in the equality $T_W = T_P$ as long as the system is not overloaded (i.e. there is always at least one taxi at the dispatcher’s disposal; in the Mielec scenario, this is true up to $n/m \approx 30$). On the other hand, minimization of T_P (RES-PT)

does not require low T_W . Moreover, it works better if the choice of awaiting requests is large, which relates to higher values of T_W . For example, one can imagine a strategy that waits until the end of a day (i.e. until all requests are submitted) and then builds routes for taxis. This strategy would provide very low T_P but at the cost of unacceptably high T_W .

6 Conclusions

The developed simulation system combining MATSim and the DVRP Optimizer proved to be useful for realistic simulation of taxi services. The high level of detail used for describing demand, supply and traffic allows for modelling precisely collaboration between the main actors, that is customers, taxi drivers and the dispatcher, all embedded into a larger transport system of a city. This collaboration takes advantage of modern ICT solutions that enable the dispatcher to smoothly coordinate the fleet, including (re-)assignments of requests to taxis. Although the Mielec scenario is not a fully real-life scenario (some data were generated artificially), there is an ongoing work on simulation of taxi dispatching for Berlin and Poznan (Poland's fifth largest city) that is both microscopic and large in its scale. Moreover, after some adaptations, the software may be used for a broad spectrum of vehicle routing and scheduling models (e.g. emergency services, demand-responsive and shared transport, or commercial fleet operations) in order to facilitate development of efficient ITS systems.

The detailed analysis of the simulation results gives us insight on the characteristics and performance of various implemented dispatching strategies, out of which three have been described in this paper. The outcomes obtained show that neither of them is best, and therefore, there is a need for a kind of 'super strategy' that would combine different qualities of the original ones. Compared to RES, the new strategy should apply a broader perspective when assigning requests to taxis. In particular, it should consider all awaiting requests (at high load), anticipate future ones (at both low and high load), and finally, focus not only on the first queued request but on the whole system's performance.

References

- [Als09] A. ALSHAMSI, S. ABDALLAH, and I. RAHWAN. "Multiagent self-organization for a taxi dispatch system". In: *8th International Conference on Autonomous Agents and Multiagent Systems*. 2009, pp. 21–28.
- [Bal08] M. BALMER, K. MEISTER, M. RIESER, K. NAGEL, and K. AXHAUSEN. "Agent-based simulation of travel demand: Structure and computational performance of MATSim-T". In: *Innovations in Travel Modeling (ITM) '08*. Also VSP WP 08-07, see www.vsp.tu-berlin.de/publications. Portland, Oregon, 2008.
- [Bar] J. BARCELO, H. GRZYBOWSKA, and S. PARDO. In: *Dynamic Fleet Management: Concepts, Systems, Algorithms & Case Studies*. Ed. by V. ZEIMPEKIS, TARANTILIS, CH.D., G. GIAGLIS, and I. MINIS. NY.

-
- [Che11] S. CHENG and T. NGUYEN. “Taxisim: A multiagent simulation platform for evaluating taxi fleet operations”. In: *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 02*. IEEE Computer Society. 2011, pp. 14–21.
- [Lee04] D. LEE, H. WANG, R. CHEU, and S. TEO. “Taxi dispatch system based on current demands and real-time traffic conditions”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1882.-1 (2004), pp. 193–200.
- [Lia08] T.-Y. LIAO, T.-Y. HU, and D.-J. CHEN. *Object-Oriented Evaluation Framework for Dynamic Vehicle Routing Problems Under Real-Time Information*. Annual Meeting Preprint 08-2222. Washington D.C.: Transportation Research Board, 2008.
- [Mac12] M. MACIEJEWSKI and K. NAGEL. “Towards multi-agent simulation of the dynamic vehicle routing problem in MATSim”. In: *Parallel Processing and Applied Mathematics (PPAM), Revised Selected Papers, Part II*. Ed. by R. WYRZYKOWSKI ET AL. Lecture Notes in Computer Science. Springer, 2012, pp. 551–560. DOI: 10.1007/978-3-642-31500-8_57.
- [Mac13a] M. MACIEJEWSKI and K. NAGEL. *Simulation and dynamic optimization of taxi services in MATSim*. VSP Working Paper 13-05. TU Berlin, Transport Systems Planning and Transport Telematics, 2013. URL: www.vsp.tu-berlin.de/publications.
- [Mac13b] M. MACIEJEWSKI and K. NAGEL. *The influence of multi-agent cooperation on the efficiency of taxi dispatching*. VSP Working Paper 13-10. TU Berlin, Transport Systems Planning and Transport Telematics, 2013. URL: www.vsp.tu-berlin.de/publications.
- [Reg98] A. REGAN, H. MAHMASSANI, and P. JAILLET. “Evaluation of Dynamic Fleet Management Systems: Simulation Framework”. In: *Transportation Research Record* 1645 (1998), pp. 176–184.
- [Seo10] K. SEOW, N. DANG, and D. LEE. “A collaborative multiagent taxi-dispatch system”. In: *Automation Science and Engineering, IEEE Transactions on* 7.3 (2010), pp. 607–616.
- [Wan09] H. WANG, D. LEE, and R. CHEU. “PDPTW based taxi dispatch modeling for booking service”. In: *Natural Computation, 2009. ICNC’09. Fifth International Conference on*. Vol. 1. IEEE. 2009, pp. 242–247.

Corresponding author: Michał Maciejewski, Poznan University of Technology, Faculty of Machines and Transportation, Institute of Machines and Motor Vehicles, ul. Piotrowo 3, 60-695 Poznan, Poland, phone: +48 61 665 5957, e-mail: michal.maciejewski@put.poznan.pl, Berlin Institute of Technology, Institute for Land and Sea Transport Systems, Transport Systems Planning (VSP), Salzufer 17-19 Sekr. SG12, D-10587 Berlin, Germany, phone: +49 30 314-23308, e-mail: maciejewski@vsp.tu-berlin.de