Michal Maciejewski, Kai Nagel

The Influence of Multi-agent Cooperation on the Efficiency of Taxi Dispatching

WISSEN IM ZENTRUM
UNIVERSITÄTSBIBLIOTHEK

Technische
Universität
Berlin

The influence of multi-agent cooperation on the efficiency of taxi dispatching

Michał Maciejewski[1,2] and Kai Nagel[2]

[1] Institute of Machines and Motor Vehicles, Faculty of Machines and Transportation,
Poznan University of Technology, ul. Piotrowo 3, 60-965 Poznan, Poland
`michal.maciejewski@put.poznan.pl`
[2] Transport Systems Planning (VSP), Institute for Land and Sea Transport Systems,
TU Berlin, Salzufer 17-19 Sekr. SG12, D-10587 Berlin, Germany
`{maciejewski,nagel}@vsp.tu-berlin.de`

Abstract. The paper deals with the problem of the optimal collaboration scheme in taxi dispatching between customers, taxi drivers and the dispatcher. The authors propose three strategies that differ by the amount of information exchanged between agents and the intensity of cooperation between taxi drivers and the dispatcher. The strategies are evaluated by means of a microscopic multi-agent transport simulator (MATSim) coupled with a dynamic vehicle routing optimizer (DVRP Optimizer), which allows to realistically simulate dynamic taxi services as one of several different transport means, all embedded into a realistic environment. The evaluation is carried out on a scenario of the Polish city of Mielec. The results obtained prove that the cooperation between the dispatcher and taxi drivers is of the utmost importance, while the customer–dispatcher communication may be reduced to minimum and compensated by the use of more sophisticated dispatching strategies, thereby not affecting the quality of service.

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

1 Introduction

Taxi dispatching is one of the most crucial components of operational taxi fleet management. Although there have been several theoretical studies investigating the on-line taxi dispatching problem[5], sophisticated dynamic routing approaches are hard to analyse theoretically (e.g. competitive analysis), and therefore, simulation tools have to be used. Some studies suggest the use of multi-agent simulation to model (partial) independence of taxi drivers[3,9,1], while other studies assume a fully centralised management[4,10].

Since the quality of the results obtained depends on the simulation method, in transport-related problems, simulation has to incorporate realistically modelled dynamism of customer demand, traffic flow phenomena and fleet management

operations. These aspects are even more crucial when considering urban areas due to high dynamics of traffic flow resulting in continuously changing travel times and often, depending on the type of services, in high volatility of demand (e.g. taxi). Particularly in the case of taxi dispatching, the use of microscopic traffic simulators allows to evaluate the performance of the service under different, often extreme, scenarios, such as sport/cultural events, bad weather conditions 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, i.e. Paramics in [4] and MITSIMLab in [9].

To sum up, we think that the best approach is to use multi-agent traffic simulation that allows for running large-scale scenarios at the microscopic level of detail. Out of various simulation platforms considered, MATSim[2] is arguably the one that is closest to meet all the requirements stated; see [6] and references therein for additional justification.

2 Cooperation in taxi dispatching

Providing efficient taxi services requires proper collaboration between the main actors, that is customers, taxi drivers and the dispatcher. There are various models of providing the services, each assuming a different level of collaboration between these actors. In the simplest one, customers order a taxi without informing about their destination, which is given only to taxi drivers after entering the taxi. Moreover, taxi drivers do not notify the dispatcher about the status of the current request until it is completed. On the opposite pole, one may consider a taxi service where customers provide the dispatcher with their destinations, taxi drivers are monitored on-line, and orders may be dynamically reassigned between taxi drivers in reaction to the current situation. One may expect that the more collaboration between the actors occurs, the more efficient the service is. However, to investigate this issue thoroughly, one may use multi-agent simulation that would mimic the interaction between actors, and between actors and the environment, such as urban traffic.

Within the current research, the following options have been considered:

- *destination knowledge* – the destination is known a priori if a customer informs the dispatcher about his/her destination.
- *request execution monitoring* – the dispatcher may monitor taxis and constantly update the timing of their schedules, which is necessary if a longer scheduling horizon is considered. 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.

Enabling any of these properties implies some extra collaboration 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.

3 Platform for simulation of taxi services

In order to carry out the research, an integrated simulation platform has been developed[7]. The system consists of two fundamental components, namely MATSim[2] and the DVRP Optimizer[6]. The former is used for modelling transport supply (including a taxicab fleet) and demand (including taxi demand) and providing queue-based traffic flow simulation. The latter is responsible for managing a fleet of taxis (or in general, any vehicle fleet) within the simulation.

Since both components are integrated tightly, one can simulate taxi services dynamically (MATSim), where events, such as request submissions, vehicle departures or arrivals, trigger optimization or update procedures (the DVRP Optimizer). Taxi drivers are modelled as dynamic (reactive) agents that move in a city network, together with other drivers, and communicate with the dispatcher and customers. Each taxi driver has a schedule made up of the following task objects:

- `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 customer.

When a taxi customer wants to take a taxi, he/she calls the taxi service (time T_i^0; the order is registered as request i) and waits until the taxi arrives. In response, the taxi dispatcher assigns the new request to one of taxis, according to a given algorithm. At time T_i^1, the selected taxi sets off for the customer. The taxi arrives at the pick-up location at time T_i^2, and after the customer is picked up, it departs (time T_i^3). Finally, the taxi drops off the passenger at the destination location at time T_i^4. At this moment, the taxi driver may start out for the next request or wait. Because of the stochasticity of taxi demand and traffic flow, times T_i^1, T_i^2, T_i^3 and T_i^4 are subject to change during simulation. The correspondence between the taxi schedule and the respective customer's plan is presented in Fig. 1.
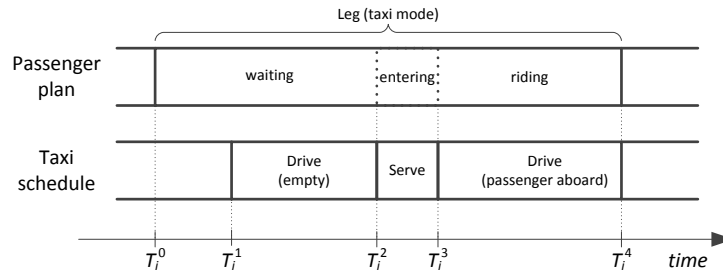

Fig. 1. A planned taxi leg and the corresponding sequence of taxi tasks

4   On-line taxi dispatching algorithms

Taxi dispatching deals with dynamic and stochastic demand and supply. Typically, a dynamic optimization algorithm reacts to various changes, represented as events, that occur over time. The simplest approach, commonly used by taxi companies, consists in responding to submissions and completions of requests (events $\mathbf{E}_i^0$ and $\mathbf{E}_i^4$, respectively). To increase efficiency, the algorithm can be triggered also when taxis set off for, arrive at and depart from pickup locations (events $\mathbf{E}_i^1$, $\mathbf{E}_i^2$ and $\mathbf{E}_i^3$, respectively). On the opposite pole, all taxicabs can be monitored on-line so that the algorithm can be executed in case any taxi is even slightly ahead of/behind schedule. In this research, it has been assumed that vehicles are not monitored and the algorithm responds only to events $\mathbf{E}_i^0$–$\mathbf{E}_i^4$.

Since on-line dispatching procedures must be time efficient, fast local optimum search methods, or even just local update methods, have to be used instead of global optimization. Provided that these procedures are fast enough (respond almost instantly to events), we can assume that they operate on static data, which are considered a kind of snapshot of the current system state.

Since customers perform immediate taxi calls and then wait for a taxi to come, minimization of the total waiting time is the optimization objective. To assure fairness of the dispatching process, all taxi requests are prioritized according to their submission time and scheduled based on the *first-come, first-served* policy.

Three different strategies, namely no-scheduling, one-time scheduling and re-scheduling, have been implemented. The first one assumes minimal communication between drivers and the dispatcher. The next one monitors request execution, while the last one extends the second one by adding the functionality of reassigning requests. All of them can be applied both with or without the a priori destination knowledge.

No-scheduling strategy (NOS) This strategy reacts to the following events:

- $\mathbf{E}_i^0$ – the *nearest*[3] vehicle among the idle ones is dispatched to this request; if no vehicle is available at that time, the request is queued in a FIFO queue
- $\mathbf{E}_i^4$ – the vehicle that has just completed request i is dispatched to the first request in the FIFO queue; otherwise, the vehicle becomes idle

This strategy imitates the way orders are assigned to taxis in a typical taxi company. The main advantage is low demand for computational power. Moreover, this strategy 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. The drawback is that its performance deteriorates as the number of idle taxis decreases — if all taxis are busy, the first idle one may appear on the opposite side of a city. The a priori destination knowledge is not taken into account.

---

[3]   Depends on the distance measure chosen; usually means 'nearest in time'.

One-time-scheduling strategy (OTS) This strategy updates the existing taxi schedules by appending a new request to the schedule of the nearest vehicle among all vehicles (both idle or busy), where *nearest* again typically means "nearest in time". Therefore, the knowledge of travel times is mandatory. This strategy monitors execution of requests and constantly updates the timelines of schedules, but without request reassignment. The strategy acts in the following way:

- $\mathbf{E_i^0}$ – request i is appended to the schedule of the *nearest* taxi
- $\mathbf{E_i^1}$ – $\mathbf{E_i^4}$ – if the vehicle serving request i is ahead of/behind time, the timing of its schedule is updated, while the assignments remain unchanged

This strategy considers all the available vehicles, not only the idle ones, which broadens the choice of taxis and thus increases the chances of finding a better assignment. However, when destinations are unknown, the planning horizon is limited up to one pick-up ahead (event $\mathbf{E_i^3}$), and therefore, vehicles with already one planned pick-up cannot be considered when scheduling a new request. The weakest point of OST is the permanence of assignments, even if a vehicle is seriously delayed.

Re-scheduling strategy (RES) This strategy is an enhanced version of the previous one. The difference is that requests may be re-assigned between taxis if the other one appears to be *nearer*. The strategy is defined as follows:

- $\mathbf{E_i^0}$ – request i is appended to the schedule of the *nearest* taxi
- $\mathbf{E_i^1}$ – $\mathbf{E_i^4}$ – if the vehicle serving request i is ahead of/behind time, full re-scheduling is carried out; the assignments are subject to change

In consequence, the scheduling algorithm runs in two modes: on the arrival of a new request (exactly as in case the OTS strategy), and when one of taxicabs is reported to be ahead of/behind schedule (all planned request are re-scheduled). RES overcomes the weaknesses of the previous ones, but at the cost of extra computational time.

5 Test scenario

The computational analysis was carried out for Mielec, a city in south-eastern Poland, with a population of over 60'000 inhabitants. The model was derived from a macroscopic model of private transport in Mielec, used as a small-size test instance in several studies, e.g.[8]. The network model consists of over 200 nodes and 600 links. The whole study area is divided into 13 zones; nine of them represent city districts, while the rest – external areas. Based on the original model that described afternoon 1-hour peak, the demand data have been artificially generated to cover the period between 6:00 am and 8:00 pm, resulting in over 42'000 private transport trips that represent a hypothetical case of day traffic, including both morning and afternoon rush-hour traffic.

The performance of the proposed optimization strategies was tested against different variants of the scenario (each repeated 20 times). The taxi demand was modelled as 3, 5 and 7% of the intra-urban private transport demand (i.e. the number of orders n equalled 917, 1528 or 2175, respectively), while the taxi fleet size m was 50 and 100. Since the fleet size was constant over the entire simulation and the taxi demand peak coincided with the private transport demand peak, the rush hours were the most challenging for taxi dispatching.

6   Simulation results

Different performance measures were used during the simulation studies, all described in [7]. They represented either the customers' or taxi company's point of view, often mutually conflicting. The selected ones are:

- average passenger waiting time, $T_{\mathrm{W}} = \sum_{i \in N}(T_i^2 - T_i^0)/n$
- maximum passenger waiting time, $T_{\mathrm{W}}^{\max} = \max_{i \in N}(T_i^2 - T_i^0)$
- average pickup trip time, $T_{\mathrm{P}} = \sum_{i \in N}(T_i^2 - T_i^1)/n$

Figures 2–4 show the results obtained for different m and n. Separate curves were plotted for the taxi fleet size of 50 (n/m between 18.34 and 43.5) and of 100 (n/m between 9.17 and 21.75). The following notation is used:

- *NOS* – NOS (destination knowledge n/a)
- *OTS w/o D* – OTS *without* the destination knowledge
- *OTS w/D* – OTS *with* the destination knowledge
- *RES* – RES both *with* and *without* the destination knowledge (aggregated due to negligible differences)

Figure 2 shows, in terms of T_{W}, that RES and OTS give comparable results for low and medium load. However, as the system gets overloaded, OTS *without* the destination knowledge behaves similarly to NOS, producing random assignments as the search space gets reduced to only 1 taxi (explained below). These relations are clearly visible in the experiments with 50 taxis and demand equal to 5–7%, where periods of high load alternate with ones of lower load. In these cases, OTS *without* the destination knowledge performs similarly either to NOS or to RES, thereby giving in-between outcomes.

The behaviour of OTS *without* the destination knowledge at high loads is caused by the fact that when a taxi with customer i aboard departs (event $\mathbf{E_i^3}$) it notifies the dispatcher about the current destination, and therefore, the dispatcher can schedule the next event, $\mathbf{E_i^4}$, and then append customer j (the first from the queue) to this taxi's schedule. This resembles the behaviour of NOS, with the exception that in the latter, the assignment is done after $\mathbf{E_i^4}$, not $\mathbf{E_i^3}$. Actually, making a random assignment following $\mathbf{E_i^3}$ is even worse since it adds more variability in terms of the waiting time ($T_j^2 - T_j^0$ depends on $T_i^4 - T_i^3$ now). As a result, the behaviour gets less fair (some customers must wait considerably longer than others), which is proved by larger $T_{\mathrm{W}}^{\max}$ (see Fig. 3).
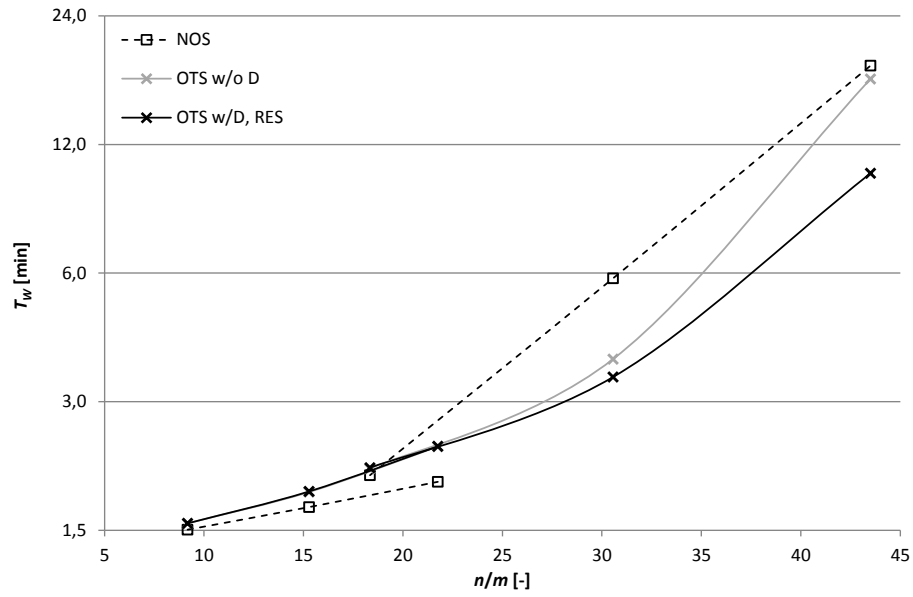
Fig. 2. Average passenger waiting time T_{W} at different demand-supply ratios
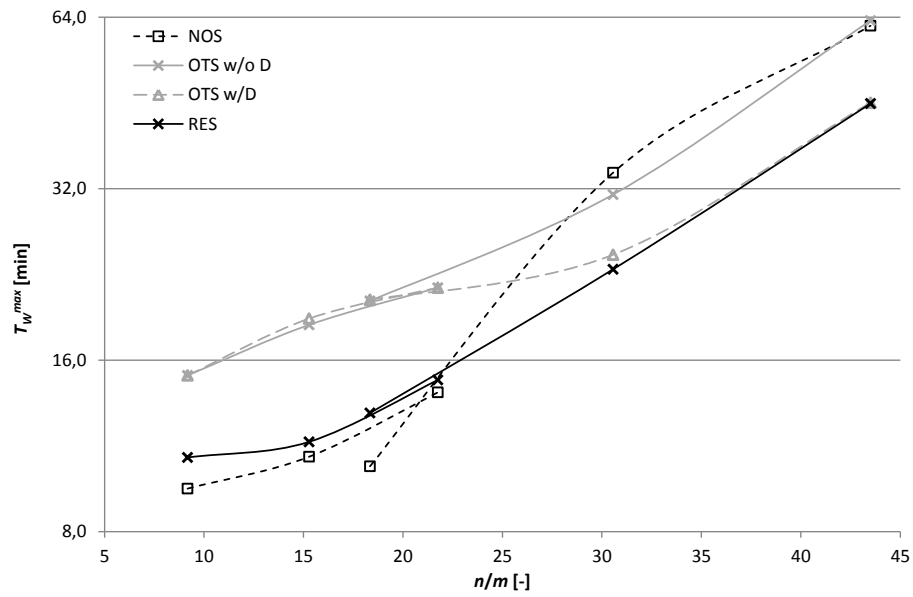

Fig. 3. Maximum passenger waiting time $T_{\mathrm{W}}^{\mathrm{max}}$ at different demand-supply ratios

Another interesting result is that the NOS series in Fig. 2 are not adjacent. This is due to the fact that the average distance to the closest idle taxi is proportional to $1/\sqrt{m_{\text{idle}}}$, where m_{idle} is the number of idle taxis. Therefore, T_{W} in experiment *3%:50* (i.e. 3% demand and 50 cabs) is about $\sqrt{2}$ times higher than in *3%:100*, and slightly higher compared to *7%:100*.

A quite surprising outcome, in terms of T_{W}, is the poorer performance of RES and OST in experiments with 100 taxis, compared to NOS. This is caused by a biased arrival time estimator for on-going trips that is used for scheduling. This estimator ignores the fact that the longer the trip lasts the more probable that it will end later than previously expected, and consequently, gives too optimistic estimates (discussed in [7]). As a result, the OTS and RES strategies give preference to non-idle vehicles, which results in shorter pickup trips (see Fig. 4), but at the cost of longer taxi awaiting.

Figure 3 shows the maximum waiting times, $T_{\text{W}}^{\text{max}}$, that may be interpreted as the upper bound (or a first-guess estimate) of T_{W} during morning and afternoon peaks. In-depth analysis of $T_{\text{W}}^{\text{max}}$ reveals that in the low load regime, NOS is the most fair strategy, while OTS and RES suffer from the biased estimation. However, wrong decisions made by RES can be reversed to some extend by later reassignments of requests, thereby reducing the performance gap. On the other hand, at high n/m ratios, the randomness of NOS and OTS *without* the destination knowledge results in higher $T_{\text{W}}^{\text{max}}$.

Figure 4 represents the taxi company's perspective. Although the main motivation of using OTS and RES is the reduction of passenger waiting times, they have a beneficial side effect of minimizing the operating costs. They both yield lower T_{P} since by considering also non-idle vehicles, they not only minimize T_i^2 (and thus T_{W}) but also allow for $T_i^1 > \tau$, which reduces T_{P} even further. The discontinuity of the 50-taxi and 100-taxi series is caused by the fact that $T_{\text{P}} \sim 1/\sqrt{m_{\text{idle}}}$, as explained earlier.

Similarly to the case of T_{W}, RES and OTS give comparable T_{P} for low and medium load. However, in an overloaded system, OTS *without* the destination knowledge operates similarly to NOS, while OTS *with* the destination knowledge performs similarly to RES. Lower T_{P} implies more idle taxis at the dispatcher's disposal, hence the dearth of taxis during the high peak is shorter and less acute.

7   Conclusions

The main goal of the paper was to propose and evaluate various schemes of collaboration between agents. The platform integrating microscopic, behaviour-based traffic simulation (MATSim) and dynamic vehicle routing optimization (DVRP Optimizer) allowed to realistically model, simulate and evaluate dynamic taxi services for the scenario of the Polish city of Mielec.

Out of three different optimization strategies, each assuming a different collaboration scheme between agents, that is customers, taxi drivers and the dispatcher, the best results have been obtained for the RES strategy. RES imposes
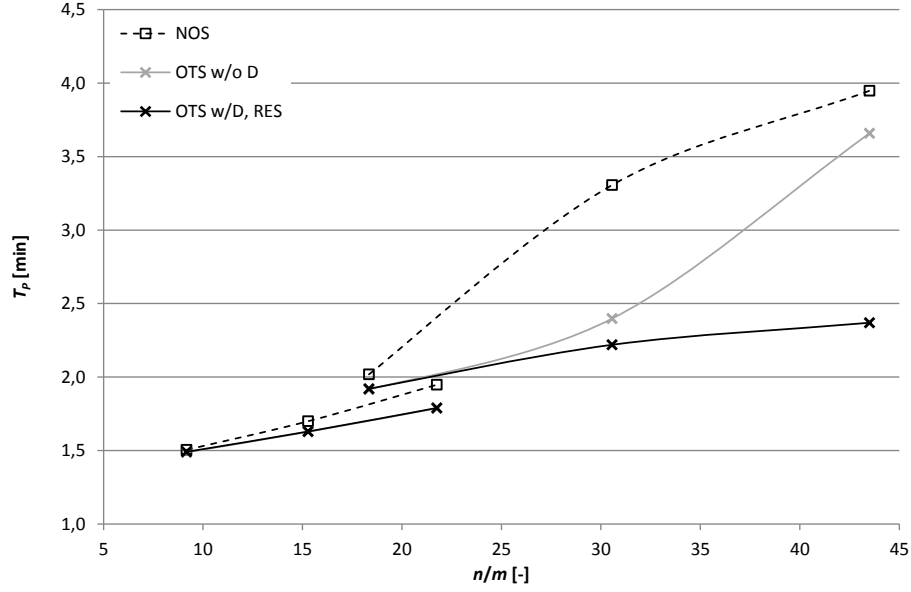
Fig. 4. Average pickup trip time T_P at different demand-supply ratios

the highest level of cooperation between taxi drivers and the dispatcher. In particular, at high demand-to-supply ratios, it turns out to be very effective, both for customers (lowest T_W and T_W^{max}) and the company (lowest T_P). On the other hand, in the low load regime, all three strategies perform similarly well. Interestingly, NOS shows more preference towards customers, while OTS and RES offer more benefit to the company. This issue is related to the biased travel time estimation, and will be addressed in future by adding on-line vehicle monitoring. With this enhancement, the RES strategy is expected to outperform the other ones regardless of the demand-supply relation.

Another interesting outcome is the limited use of the a priori destination knowledge, which implies that the dispatcher may respect customers' privacy and not ask them about their destinations. This information is not used by NOS, and has a negligible impact on RES. Only in the case of OTS, the not-knowing the customer's destination deteriorates taxi dispatching since contrary to RES, wrong decisions cannot be reversed. However, the lack of this knowledge may be compensated by the use of RES, which is superior, or at least equal, to OTS in every respect.

To conclude, the cooperation between the dispatcher and taxi drivers is of the utmost importance, while the customer–dispatcher communication, if necessary, may be reduced to minimum. The lack of the knowledge of destination can be balanced out by means of more sophisticated dispatching strategies so that the quality of service remains unaffected. Among plans for the future is the extension of the platform functionality with an on-line vehicle tracking module in order to

simulate the real-time driver–dispatcher cooperation. Another aim is to extend taxi drivers' independence by letting them to follow their own *cruise-or-wait* strategies when idle.

References

1. Alshamsi, A., Abdallah, S., Rahwan, I.: Multiagent self-organization for a taxi dispatch system. In: 8th International Conference on Autonomous Agents and Multiagent Systems. pp. 21–28 (2009)
2. Balmer, M., Meister, K., Rieser, M., Nagel, K., Axhausen, K.: Agent-based simulation of travel demand: Structure and computational performance of MATSim-T. In: Innovations in Travel Modeling (ITM) '08. Portland, Oregon (June 2008), also VSP WP 08-07, see www.vsp.tu-berlin.de/publications
3. Cheng, S., Nguyen, T.: 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. pp. 14–21. IEEE Computer Society (2011)
4. Lee, D., Wang, H., Cheu, R., Teo, S.: Taxi dispatch system based on current demands and real-time traffic conditions. Transportation Research Record: Journal of the Transportation Research Board 1882(-1), 193–200 (2004)
5. Ma, W., Wang, K.: On the on-line weighted k-taxi problem. Combinatorics, Algorithms, Probabilistic and Experimental Methodologies pp. 152–162 (2007)
6. Maciejewski, M., Nagel, K.: Towards multi-agent simulation of the dynamic vehicle routing problem in MATSim. In: R. Wyrzykowski et al (ed.) Parallel Processing and Applied Mathematics (PPAM), Revised Selected Papers, Part II, pp. 551–560. Lecture Notes in Computer Science, Springer (2012)
7. Maciejewski, M., Nagel, K.: Simulation and dynamic optimization of taxi services in MATSim. VSP Working Paper 13-05, TU Berlin, Transport Systems Planning and Transport Telematics (2013), `www.vsp.tu-berlin.de/publications`
8. Piatkowski, B., Maciejewski, M.: Comparison of traffic assignment in VISUM and transport simulation in MATSim. Transport Problems (2013 (in press))
9. Seow, K., Dang, N., Lee, D.: A collaborative multiagent taxi-dispatch system. Automation Science and Engineering, IEEE Transactions on 7(3), 607–616 (2010)
10. Wang, H., Lee, D., Cheu, R.: PDPTW based taxi dispatch modeling for booking service. In: Natural Computation, 2009. ICNC'09. Fifth International Conference on. vol. 1, pp. 242–247. IEEE (2009)