

Towards a testbed for dynamic vehicle routing algorithms

Michał Maciejewski^{2,1}, Joschka Bischoff¹, Sebastian Hörl³, Kai Nagel¹

¹ Transport Systems and Transport Telematics, Technische Universität Berlin, SG12, Salzufer 17–19, 10587 Berlin, Germany

² Division of Transport Systems, Poznan University of Technology, Piotrowo 3, 60-965 Poznan, Poland

³ IVT, ETH Zürich, CH-8093 Zürich, Switzerland

Abstract. Since modern transport services are becoming more flexible, demand-responsive, and energy/cost efficient, there is a growing demand for large-scale microscopic simulation platforms in order to test sophisticated routing algorithms. Such platforms have to simulate in detail, not only the dynamically changing demand and supply of the relevant service, but also traffic flow and other relevant transport services. This paper presents the DVRP extension to the open-source MATSim simulator. The extension is designed to be highly general and customizable to simulate a wide range of dynamic rich vehicle routing problems. The extension allows plugging in of various algorithms that are responsible for continuous re-optimisation of routes in response to changes in the system. The DVRP extension has been used in many research and commercial projects dealing with simulation of electric and autonomous taxis, demand-responsive transport, personal rapid transport, free-floating car sharing and parking search.

Keywords: dynamic vehicle routing, DVRP, multi-agent, traffic flow, MATSim, autonomous taxis, shared taxis, demand-responsive transport, DRT, testbed, benchmark

1 Introduction

The recent technological advancements in ICT provide novel, on-line fleet management tools, opening up a broad range of possibilities for more intelligent transport services: flexible, demand-responsive, safe and energy/cost efficient. Significant enhancements can aid in both traditional transport operations, like regular public transport or taxis and introduction of novel solutions, such as demand-responsive transport (DRT) or personal rapid transport (PRT). However, the growing complexity of modern transport systems, despite all benefits, increases the risk of poor performance, or even failure, due to the lack of precise design, implementation and testing.

A traditional approach to test algorithms are benchmark instances (e.g. [26,11]). A number of instances are made available, often on the web, and algorithms to solve these problems are collected, together with performance numbers. To make performances comparable, all algorithms should be run on the

same computers, maybe provided by the team that is providing the benchmark. And ideally there would be a “blind” part of the testing, where the submitted algorithms are run on benchmark instances that were not published beforehand.

Benchmark instances have been successfully applied to analyse and compare performances of algorithms solving different kinds of static Vehicle Routing Problems (VRP) [10,31], where, in the most basic version of the problem, one wants to determine minimal-cost vehicle routes, which begin at the depot, visit a subset of the customers, and return to the depot.

The *Dynamic* VRP (DVRP, [24,2,23]) is different from the static problem in that not all information relevant to the planning of the routes is known when the routing process begins, and some information, such as expected travel times, may be imprecise and stochastic. In the dynamic case, like in other online problems, where algorithms need to run *while* the controlled system is evolving [8], the use of static benchmark datasets is problematic. For example, with a static benchmark dataset one needs to be really diligent not to make the algorithm look into the future when the future (e.g. future requests) is already available in the input data. Also, with a pre-computed benchmark, it is impossible to have the system react to the control – for example cancelling requests when pick-ups take too long. Moreover, even simple pre-planned event sequences may lead into inconsistencies: for example, serving a request between its scheduled submission and cancellation. Finally, the controlled system may be stochastic, in which case there needs to be either an analytic description of the stochastic properties of the system, or some way to generate random draws from the system. Consequently, there has been no reference benchmarks for DVRP so far [23].

These above problems could potentially be addressed by theoretical approaches to the analysis of online optimisation algorithms, such as competitive analysis [8]. However, their use is mostly limited to simplified cases, and for large and complex real-life problems the simulation approach is often the only viable way to evaluate, compare and refine dynamic algorithms [12]. For example, for dynamic vehicle routing problems, one could have a simulation tool that generates requests and travel times randomly. A to-be-evaluated dispatch algorithm then needs to direct a fleet of vehicles to serve these requests.

In order to be realistic, such tools have to model, in detail, not only the dynamically changing demand and supply of the relevant service, but also traffic flow and other existing transport services, including mutual interactions/relations between all these components. Although several approaches have been proposed [25,1,17,9], as far the authors know, no existing solutions provide large-scale microscopic simulation that include all the components above. [28] provides a recent review of existing agent-based simulators for DRT.

2 MATSim and DVRP

One possible solution to simulation-based benchmarking of online algorithms is to use an existing transport/traffic simulator instead of creating one from scratch. Such a tool should allow for detailed modelling of complex interdepen-

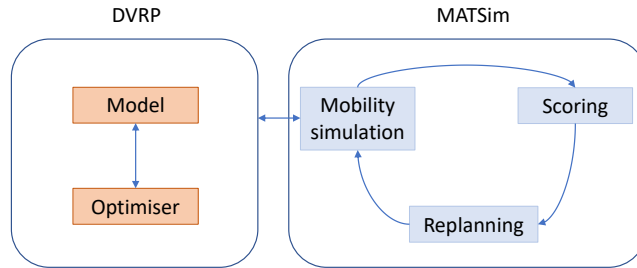


Fig. 1. Integration of the DVRP module in the MATSim multi-iterative simulation process. [14]

dependencies between the three main components, that is customer demand, traffic flow, and vehicle fleet, and be able to run large-scale simulation. MATSim (Multi-Agent Transport Simulation) [15] offers a comprehensive set of features that renders it suitable for benchmarking purposes. First of all, it is a multi-agent activity-based microsimulation system for daily transport demand analysis. Secondly, due to a fast and efficient traffic simulation, it is able to conduct analyses for large scenarios, even concerning a whole country. Last but not least, MATSim modularity and openness (open-source software) allow for extending and adjusting its functionality to one’s needs.

To enable simulation of vehicle routing in MATSim, the DVRP (*Dynamic Vehicle Routing Problem*) extension [19] has been developed. The extension is designed to be highly general and customizable to model a wide range of dynamic vehicle routing problems, including the so-called *Rich DVRPs*. Compared to the classic VRP, the major model enhancements are: (a) one-to-many (many-to-one) and many-to-many topologies, (b) multiple depots, (c) dynamic requests, (d) request and vehicle types, (e) time windows for requests and vehicles, (f) time-dependent stochastic travel times and costs, and (g) network-based routing (including route planning, vehicle monitoring and diversion). The DVRP extension is in direct interaction with the simulation during the execution of the traffic flow simulation. Figure 1 shows the interaction of the main MATSim modules and the DVRP extension. One can easily extend the existing model even further to cover other specific cases (see Section 5), which then require at least a specific optimizer (see Section 4).

3 DVRP model and simulation

3.1 Basic model

The DVRP extension is responsible for modelling both demand and supply of the transport service in question, while MATSim simulates them together with other components of the overall transport system.

In the most basic setup, the DVRP problem is defined by a fleet of `Vehicles` that serves submitted `Requests`. Requests come in over time, and not all requests

need to be known when the first request is served. In addition, the traffic state is not fully known ahead of time, although travel time and cost estimates are available.

The way a vehicle serves assigned requests is defined by its schedule that consists of a sequence of **Tasks**, such as driving from one location to another, or staying at a given location. Schedules are computed and managed by the optimiser algorithm (Sec. 4). In order to build a valid schedule, the optimiser needs to take into account all constraints. Among them are network constraints, such as turning or vehicle restrictions, and time constraints, such as time windows or travel times. Therefore, when planning a drive from one place to another, the optimiser uses one of the least cost path calculators available in MATSim. It finds a minimum-cost path through the network, given the departure time, vehicle type and cost function.

3.2 Dynamic agents

The standard day-to-day re-planning approach in MATSim [14] assumes that agents can make complete daily activity plans only between simulation runs. During simulation, plans are executed without any change. This renders MATSim’s standard implementation of agents not suitable for simulating Dynamic Vehicle Routing Problems, where each driver agent (either human or robot) behaves dynamically and follows tasks coming continuously from the dispatcher. In order to overcome this limitation, the dynamic agent was introduced as an alternative to the standard pre-planned agent. The dynamic agent provides the foundation for simulating dynamically behaving agents in a wide range of different simulation scenarios, DVRP being only one of them.

In general, the dynamic agent can actively decide what to do at each simulation step instead of using a pre-computed plan. It is up to the agent whether decisions are made spontaneously or (re-)planned in advance. In some applications, the agent is fully autonomous and acts according to his/her desires, beliefs and intentions, whereas in other ones, it may just follow orders systematically issued from the outside, which is the case of dynamic vehicle routing.

Contrary to the standard MATSim agent executing a fixed plan of static activities and legs, the dynamic agent performs dynamic activities and legs, which are usually created on the fly, and can be modified at any time. Moreover, dynamic activities and legs are simulated. For instance, when driving a car, the agent can change the route, destination or even decide about picking up or dropping off somebody on the way.

In the present paper, the dynamic agent will be used to represent the taxicab driver. In general, however, it can be used to represent arbitrary agent types.

3.3 Simulation

Realistic simulation of DVRP requires a proper behavioural model of the taxicab driver agent. At a higher level, this agent follows its dynamically changing schedule. After completing one task, it switches to the next one. The new task

is then translated into a dynamic action, which is performed by the agent. For instance, a pickup task is translated into a dynamic pickup activity, where the driver agent first waits for the passenger agent (if not yet there), and then lets it board the vehicle. In this specific case, simulation of the task includes a direct interaction between agents. In other cases, the interaction may be indirect, e.g. an agent moving through the network contributes together with other drivers to the overall traffic.

While executing scheduled tasks, the driver can be continuously monitored with task trackers that offer functionality similar to GPS navigation, such as tracking its movement or predicting task completion time. Moreover, the communication can be two-way. For instance, while executing a drive task, the agent can be ordered to divert from its current destination.

The current implementation of the driver agent logic assumes that drivers strictly follow the schedules which are managed centrally by the optimiser. However, the logic can be modified in order to give more autonomy to drivers (e.g. planning routes, accepting/rejecting requests, specifying their availability). In the most extreme case, drivers could fully decide for themselves, while the optimiser would act as a middleman, only establishing contact between customers and drivers.

4 Optimiser

As in real-life online fleet management systems, the central element in the DVRP extension is the optimisation algorithm. The optimiser reacts to events generated during simulation, which could be: request submissions, vehicle departures or arrivals, etc. Additionally, it can monitor the movement of individual vehicles, as well as query other sources of online information, e.g. current traffic conditions. In response to changes in the system, the optimiser may update vehicles' schedules, either by applying smaller modifications or re-optimizing them from scratch. Vehicles are notified about changes in their schedules and adjust to them as soon as possible, including immediate diversion from their current destinations.

The extension's architecture allows the plugging in of various on-line vehicle routing algorithms (optimisers). In order to plug an optimiser into the simulator, it has to implement the base `VrpOptimizer` interface or one of its more specialised subinterfaces. The base interface contains the following two methods:

- `requestSubmitted(Request request)`—called on submitting `request`; in response, the optimiser either adapts vehicles' schedules so that `request` can be served, or rejects it.
- `nextTask(Vehicle vehicle)`—called whenever `vehicles`'s current task has been completed and the vehicle will switch to the next planned task; this is the last moment to make or revise the decision on what to do next.

This basic functionality can be freely extended. References to more complex optimisers used for such services like taxi, shared taxi or demand-responsive transport can be found in Sections 5 and 6.

In general, there are two ways of responding to the incoming events. They can be handled either *immediately (synchronously)* or *between simulation time steps (asynchronously)*. In the former case, schedules are re-calculated (updated or re-optimised) directly, in response to the calling of the optimiser’s methods. This simplifies accepting/rejecting new requests, since the answer is immediately passed back to the caller. In the latter case, all events observed within a simulation period are buffered and then processed collectively just before the next simulation period begins. Mixing both approaches by answering immediately to some events, and buffering other ones is also possible. Regardless of the approach taken, special care must be given to thread safety when running multi-threaded simulation and/or optimisation.

5 Example

This section discusses creating a DVRP optimiser and running a MATSim simulation for autonomous taxi (AT) service, which dispatches ATs in response to continuously incoming taxi calls. The central part is the `ATOptimizer` class (Listing 1). As stated above, by implementing the standard `VrpOptimizer` interface, the optimiser can react to two types of events: submission of a new request, and switching to the next tasks. The former is handled in the `requestSubmitted(Request)` method, where the optimiser schedules the newly submitted `request`. The latter, in turn, is handled in the `nextTask(Vehicle)` method, which is called when `vehicle` finishes its current task. In this case, the optimiser updates this vehicle’s schedule, and then switches `vehicle`’s current task to the next one.

Listing 1. `ATOptimizer.java`

```
public class ATOptimizer implements VrpOptimizer {
    public void requestSubmitted(Request request) {
        scheduleRequest(request);
    }

    public void nextTask(Vehicle vehicle) {
        updateScheduleAndSwitchToNextTask(vehicle);
    }
    ...
}
```

The standard procedure of running a MATSim simulation consists of the following four steps:

1. load a config file (containing all simulation parameters)
2. load scenario data (i.e. read data and initialise all object structures)
3. create and configure a simulation controller
4. run simulation

Because simulating ATs requires the DVRP extension to be added, the base scheme has to be extended with loading the fleet data and then adding the implemented optimiser to the controller (between steps 3 and 4). Listing 2 presents

a code that configures and runs AT simulation. In this example, all *taxi* trips will be served by fleet of ATs according to the optimisation algorithm defined in the `ATOptimizer` class.

Listing 2. AT simulation

```
Config config = loadConfig(); // (step 1)
Scenario scenario = loadScenario(config); // (step 2)
Controller controller = new Controller(scenario); // (step 3)

Fleet fleet = loadFleet();
addVrpOptimizer(controller, "taxi", fleet, ATOptimizer.class);

controller.run(); // (step 4)
```

Complete, executable examples presenting the use of the DVRP extension to simulate different on-demand transport services can be found in the following places in the MATSim project:

- `RunOneTaxiExample` (see <http://matsim.org/javadoc> → `dvrp`) – the most basic example on dispatching a single taxi to serve incoming requests
- `RunRobotaxiExample` (see <http://matsim.org/javadoc> → `av`) – a robotaxi (AV taxi) fleet is used to replace all private car trips of a random population in the city of Cottbus, Germany
- `RunTaxiExample` (see <http://matsim.org/javadoc> → `taxi`) – used to benchmark the taxi optimisers available in the taxi extension; a fleet of taxis is dispatched to serve taxi calls in the city of Mielec, Poland
- `RunSharedTaxiExample` (see <http://matsim.org/javadoc> → `taxi`) – a simple shared taxi algorithm for up to two passengers with different origins and destinations set in Braunschweig, Germany.
- `RunTaxibusExample` (see <http://matsim.org/javadoc> → `taxi`) – an example for a DRT shuttle system where passengers want to reach a train station in Braunschweig, Germany.

6 Existing applications

The development of the DVRP testbed originated in the simulation of taxi fleets and was more recently extended to larger fleets of AVs and demand-responsive transport. One first use case was the simulation of taxi demand in the Polish city of Mielec. For this scenario, different dispatch algorithms have been developed and tested. The model has been used for comparison of off-line and on-line taxi dispatching optimization algorithms [18,22] and simulation of electric taxicab fleets [3].

Based on the actual demand for taxi trips in Berlin [6] and Barcelona [30], several different taxi dispatch algorithms were measured in their performance under different demand levels. The results suggest that First-In-First-Out algorithms are capable of handling typical taxi demand levels reasonably well. In

sudden demand peaks, however, strategies focusing on minimizing idle mileage for vehicles (thus maximising throughput) rather than customer waiting times perform far better in terms of waiting time for the majority of passengers. Also approaches based on solving a linear assignment problem were experimented [20]. The strategy produced better results for both drivers (less idle driving) and passengers (less waiting) than rule-based approaches, but is more demanding in terms of computational requirements.

Another field of usage of the DVRP testbed is the investigation of demand-responsive transport (DRT) systems, which generally incorporate the transportation of several passengers with different destinations at the same time. The conceptual use cases for them are wide ranged and can possibly include one-to-many or many-to-many relations with different kinds of operational schemes in terms of spatial limitation or usage of stops and pre-bookings.

In one case study, a pre-booked dial-a-ride DRT service for commuter transit of a car production plant was developed and assessed for the Wolfsburg region in central Germany. With car traffic to and from the production plant causing considerable congestion during peak hours, a high-quality DRT service picking up passengers at home and dropping them off at their workplace could possibly help to reduce congestion around arterials. To achieve this, while keeping the commuting times of DRT users at an acceptable level, a compromise between vehicle capacity, waiting times and in-vehicle travel times was found [7]. Using 8-seat vehicles, travel times of DRT passengers were simulated to be only slightly higher than using private cars.

Further DRT applications include an application to evaluate several DRT schemes for a thinly populated area in Australia [27]. Based on the same infrastructure, a DRT service that includes both passenger and parcel distribution has also been simulated [29]. A possible DRT system using public transport stops and allowing passengers with common trip patterns to share rides was evaluated on a pseudo-realistic scenario of the Cottbus region (Germany)[16].

More recently, the ability to simulate very large fleets of (automated) taxis have been assessed. The largest scenario calculated so far is based on the inner-city car trips in Berlin. These accumulate to roughly 2.5 million trips per day and are usually handled by a fleet of roughly 1.2 million privately owned cars. In [5,4] these trips were shifted from car mode to shared autonomous vehicle (SAV). While this resembles an extreme scenario (no more private cars), it demonstrates the overall potential of SAVs and opens up further branches of research. In order to cope with a fleet size of hundreds of thousands of vehicles, vehicle dispatch is based on rule-based heuristics that either assign the vehicle closest to a request (in times when there is an oversupply of vehicles) or assign a vehicle to the closest request. Zonal registers were used to pre-filter vehicles and requests close to each other. Results suggest that a fleet of 100 000 to 110 000 vehicles is sufficient to replace all car trips, leading to a possible fleet reduction of 90 %. Waiting times for passengers can be kept in a range between 3 to 5 minutes for the majority of customers even during peak times. On average, vehicles spend more than 80% of the driven mileage with a customer on board. Pick-up trips are generally

longer in the outskirts of the city [4], leading to the question whether offering such a service in less-densely populated areas is of commercial or communal interest. During further research, the possible positive impact of an improved traffic flow of automated vehicles has been assessed, concluding that a minor increase in traffic flow capabilities of SAVs would be enough to (over)-compensate the negative congestion impact of the additional empty trips of SAV fleets. [21]

The interplay of multiple dynamic AV taxi services has been tested in combination with a responsive demand in [13]. A coherent framework of utility scoring for traditional modes of transport, as well as for autonomous vehicles has been set up, such that the artificial simulation population is able to choose between conventional means of transport and the new services. On the supply side, single-passenger SAVs have been offered in competition to multi-passenger (“pooled”) AVs. Both services used a dispatching heuristic by Maciejewski and Bischoff [5], while the latter one aggregated up to four yet unserved requests with origin and destination locations not being farther apart than 400 m. The study showed that with MATSim and DVRP it is possible to model coherent demand reactions to new forms of dynamic transportation. While both AV offers were used at peak-hours, customers favoured the pooled AVs at off-peak hours, where they were able to weigh the slightly increased travel times but low prices of the pooled service against considerably higher prices (but shorter travel times) of the AV taxi service.

7 Conclusions

This paper presents the current functionality and typical use cases of the DVRP extension. By providing an abstraction layer for modelling of fleet management operations, the extension allows for simulation of dynamic vehicle routing in MATSim. Since 2011, the extension has been actively developed and continuously new features are being added. Currently, our efforts are focused on: (a) simplifying the process of plugging in custom-made VRP optimisers and models by means of dependency injection, (b) speedup and parallelisation of large-scale simulations, and (c) standardisation of the DVRP interfaces. By being able to simulate large and complex transport systems at the microscopic level of detail, and offering code modularity and openness, MATSim and DVRP can serve as a testbed platform for dynamic vehicle routing problems.

References

1. Barcelo, J., Grzybowska, H., Pardo, S.: Vehicle routing and scheduling models, simulation and city logistics. pp. 163–195. Springer, NY (2007)
2. Bektas, T., Repoussis, P.P., Tarantilis, C.D.: Chapter 11: Dynamic Vehicle Routing Problems, pp. 299–347. <http://epubs.siam.org/doi/abs/10.1137/1.9781611973594.ch11>
3. Bischoff, J., Maciejewski, M.: Agent-based simulation of electric taxicab fleets. *Transportation Research Procedia* 4, 191–198 (2014), also VSP WP 14-10, see <http://www.vsp.tu-berlin.de/publications>

4. Bischoff, J., Maciejewski, M.: Autonomous taxicabs in berlin – a spatiotemporal analysis of service performance. *Transportation Research Procedia* 19, 176–186 (2016)
5. Bischoff, J., Maciejewski, M.: Simulation of city-wide replacement of private cars with autonomous taxis in Berlin. *Procedia Computer Science* 83, 237–244 (2016), <http://www.sciencedirect.com/science/article/pii/S1877050916301442>
6. Bischoff, J., Maciejewski, M., Sohr, A.: Analysis of Berlin’s taxi services by exploring GPS traces. In: *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015 International Conference on*. pp. 209–215 (June 2015)
7. Bischoff, J., Soeffker, N., Maciejewski, M.: A framework for agent based simulation of demand responsive transport systems. Tech. rep., presented at OR2016 (2016)
8. Borodin, A., El-Yaniv, R.: *Online computation and competitive analysis*. Cambridge University Press (2005)
9. Certicky, M., Jakob, M., Pibil, R., Moler, Z.: Agent-based simulation testbed for on-demand transport services. In: *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*. pp. 1671–1672. AAMAS ’14, International Foundation for Autonomous Agents and Multiagent Systems, Richland (May 2014), <http://dl.acm.org/citation.cfm?id=2615731.2616118>
10. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. *Management science* 6(1), 80–91 (1959)
11. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: The KITTI vision benchmark suite (accessed 2017-02-16)
12. Grötschel, M., Krumke, S.O., Rambau, J., Winter, T., Zimmermann, U.T.: Combinatorial online optimization in real time. In: *Online optimization of large scale systems*, pp. 679–704. Springer (2001)
13. Hörl, S.: Agent-based simulation of autonomous taxi services with dynamic demand responses. Tech. rep., *Arbeitsberichte Verkehrs- und Raumplanung* 1229 (2017)
14. Horni, A., Nagel, K., Axhausen, K.W.: Introducing MATSim. In: Horni, A., Nagel, K., Axhausen, K.W. (eds.) *The Multi-Agent Transport Simulation MATSim*, chap. 1. Ubiquity, London (2016), <http://matsim.org/the-book>
15. Horni, A., Nagel, K., Axhausen, K.W. (eds.): *The Multi-Agent Transport Simulation MATSim*. Ubiquity, London (2016), <http://matsim.org/the-book>
16. Hosse, D., Neumann, A.: Modelling of operational variants for the use of personal rapid transit in public transit (2015), https://www.innoz.de/sites/default/files/heart_2015_submission_88_final.pdf
17. Liao, T.Y., Hu, T.Y., Chen, D.J.: Object-oriented evaluation framework for dynamic vehicle routing problems under real-time information. *Annual Meeting Preprint 08-2222*, Transportation Research Board, Washington, D.C. (January 2008)
18. Maciejewski, M.: Benchmarking minimum passenger waiting time in online taxi dispatching with exact offline optimization methods. *Archives of Transport* 30(2), 67–75 (2014)
19. Maciejewski, M.: Dynamic transport services. In: Horni, A., Nagel, K., Axhausen, K.W. (eds.) *The Multi-Agent Transport Simulation MATSim*, chap. 23. Ubiquity, London (2016), <http://matsim.org/the-book>
20. Maciejewski, M., Bischoff, J., Nagel, K.: An assignment-based approach to efficient real-time city-scale taxi dispatching. *IEEE Intelligent Systems* 31(1), 68–77 (Jan 2016)
21. Maciejewski, M., Bischoff, J., Nagel, K.: Congestion effects of Autonomous Taxi fleets. *Transport in preparation* (2017)

22. Maciejewski, M.: Benchmarking minimum passenger waiting time in online taxi dispatching with exact offline optimization methods. *Archives of Transport* 30(2), 67–75 (2014)
23. Pillac, V., Gendreau, M., Guéret, C., Medaglia, A.L.: A review of dynamic vehicle routing problems. *European Journal of Operational Research* 225(1), 1 – 11 (2013), <http://www.sciencedirect.com/science/article/pii/S0377221712006388>
24. Psaraftis, H.N., Wen, M., Kontovas, C.A.: Dynamic vehicle routing problems: Three decades and counting. *Networks* 67(1), 3–31 (2016)
25. Regan, A., Mahmassani, H., Jaillet, P.: Evaluation of dynamic fleet management systems: Simulation framework. *Transportation Research Record* 1645, 176–184 (1998)
26. Reinelt, G.: TSPLIB95. <https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/> (accessed 2017-02-16)
27. Ronald, N.: Yarrowonga and mulwala: Demand-responsive transportation in regional victoria, australia. In: Horni, A., Nagel, K., Axhausen, K.W. (eds.) *The Multi-Agent Transport Simulation MATSim*, chap. 95. Ubiquity, London (2016), <http://matsim.org/the-book>
28. Ronald, N., Thompson, R., Winter, S.: Simulating demand-responsive transportation: A review of agent-based approaches. *Transport Reviews* 35(4), 404–421 (2015)
29. Ronald, N., Yang, J., Thompson, R.G.: Exploring co-modality using on-demand transport systems. *Transportation Research Procedia* 12, 203 – 212 (2016), <http://www.sciencedirect.com/science/article/pii/S2352146516000600>
30. Salanova, J.M., Romeu, M.E., Amat, C.: Aggregated modeling of urban taxi services. *Procedia - Social and Behavioral Sciences* 160, 352 – 361 (2014), <http://www.sciencedirect.com/science/article/pii/S187704281406248X>
31. Toth, P., Vigo, D.: *Vehicle Routing*. Society for Industrial and Applied Mathematics, Philadelphia, PA (2014), <http://epubs.siam.org/doi/abs/10.1137/1.9781611973594>