

Maciejewski, M.; Nagel, Kai

Towards Multi-Agent Simulation of the Dynamic Vehicle Routing Problem in MATSim

Conference paper | Accepted manuscript (Postprint)

This version is available at <https://doi.org/10.14279/depositonnce-8331>



The final authenticated version is available online at https://doi.org/10.1007/978-3-642-31500-8_57.

Maciejewski, M.; Nagel, K. (2012). Towards Multi-Agent Simulation of the Dynamic Vehicle Routing Problem in MATSim. Lecture Notes in Computer Science, 551–560.
https://doi.org/10.1007/978-3-642-31500-8_57

Terms of Use

Copyright applies. A non-exclusive, non-transferable and limited right to use is granted. This document is intended solely for personal, non-commercial use.

WISSEN IM ZENTRUM
UNIVERSITÄTSBIBLIOTHEK

Technische
Universität
Berlin

Towards Multi-Agent Simulation of the Dynamic Vehicle Routing Problem in MATSim

Michał Maciejewski¹ and Kai Nagel²

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
`nagel@vsp.tu-berlin.de`

Abstract. The paper presents the idea and the initial outcomes of integrating MATSim, a multi-agent transport simulation system, with the DVRP Optimizer, an application for solving the Dynamic Vehicle Routing Problem. First, the justification for the research is given and the state of the art is outlined. Then, MATSim is presented with a short description of the recent results in areas related to the one proposed in the paper, followed up by the discussion on the DVRP Optimizer functionality, architecture and implemented algorithms. Next, the process of integrating MATSim and the DVRP Optimizer is presented, with the distinction of two phases, the *off-line* and *on-line* optimization. Then, a description of the *off-line* optimization is given along with the results obtained for courier and taxi services in urban areas. The paper ends with conclusions and future plans.

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

1 Introduction

The Vehicle Routing Problem (VRP) [1, 2] is among the most complex and fundamental components of modern fleet management. From the very beginning, research on VRP was focused on providing solutions aimed at cost efficiency while skipping other aspects of routing. However, nowadays companies must be not only cost effective, but also more open for the customer. Consequently, they offer more sophisticated and flexible services concerning also the timeliness and responsiveness to changing customers' needs. This leads ultimately to the Dynamic Vehicle Routing Problem (DVRP) [3], where the time dimension is one of the main concerns.

Despite the huge technical potential for running dynamic optimization, the existing routing software concerns mainly static routes, and transportation management is mainly performed by human planners and dispatchers, even in large companies [3]. The main problem is that throughout the decades of research on

VRP, the great emphasis was put on heuristics accuracy and speed, whereas simplicity and flexibility were out of focus [4]. As a consequence, the best state-of-the-art algorithms give very good results for many theoretical test instances of the static VRP, but they are hard to adapt to the dynamic real-world problems.

Therefore, it is necessary to focus the future research on realistic VRPs. But this requires the development of a system that will be able to simulate various real-world vehicle routing problems thus allowing for both testing optimization algorithms and planning transport services. Such a realistic simulation has to incorporate realistically modelled dynamism of customer demand, traffic flow phenomena and fleet management operations. Especially the optimization of transport services in urban areas is extremely challenging 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). Moreover, when considering city-logistics policies, many additional and often mutually conflicting objectives appear as, for example, the reduction of the negative influence on the environment and on the local society, or the positive influence on city development.

In the recent years several approaches that combine vehicle routing and traffic simulation have been proposed and implemented. In one of the first works in this field Regan et al. [5] have proposed a simplified simulation framework for evaluation of dynamic fleet management systems for the truckload carrier operations. Taniguchi et al. [6] have analysed integration of vehicle routing optimization and traffic simulation for optimization of city-logistics oriented problems. Another attempt is an application of the AIMSUN simulator for optimization of the VRP in cities by Barcelo et al. [7]. Liao et al. [8] have developed a system for evaluation DVRP strategies using real-time information. Unfortunately, all these approaches are only partial solutions, as they do not include realistic demand modelling and simulation. This limits their application mainly to problems with static and known a priori demand. As a consequence, it is impossible, for instance, to 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 small.

2 Multi-Agent Simulation and the DVRP

One possible solution to overcome the deficiencies of the existing solutions is to use a system that allows for detailed modelling of complex interdependencies between the three main components, that is customer demand, traffic flow and vehicle fleet, and that is able to run large-scale simulation. The first requirement implies use of a multi-agent simulation approach that includes all the actors and components, that is *inhabitants* that generate *traffic* in a network and, as a *customers*, create *demand* for services which are provided by *companies* which in turn have *vehicles* that are dispatched to serve the customers' requests and thus also participate in *traffic* generation. The services may be connected with transport of passengers (e.g. taxi) or goods (e.g. courier).

Since fast and realistic traffic flow simulation is the key issue, before choosing a concrete simulation system, four popular traffic simulators, namely MATSim, TRANSIMS, VISSIM and SUMO, had been tested [9, 10]. All the systems except for SUMO gave, after calibration, correct results. However, comparing the systems' functionality, MATSim (Multi-Agent Transport Simulation) [11] offers the most comprehensive set of features concerning the research goals. 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.

In recent years MATSim has been applied in several research works that are to some extent related with the one proposed. Dobler [12] has looked at the so-called within-day replanning approaches for MATSim. In the resulting implementation, synthetic travellers are able to change their routes and/or their future activity locations. Rieser [13] has investigated and implemented a public transit (pt) module for MATSim, where passengers, being able to plan pt routes, can walk to pt stops, where they wait until they are picked up by a corresponding pt vehicle; the pt vehicle drops them off at their destination, from where they continue, either to another pt vehicle or to their next activity. Neumann et al. [14] have implemented a simple control strategy for bus drivers, which implies that a bus would delay itself if the bus behind is delayed. Schroeder et al. [15] have simulated synthetic firms that generate demand for shipments. On the other hand, concerning passenger transport, Ciari et al. [16] have used MATSim for the estimation of car sharing demand. That study does, however, not include an implementation of actual car sharing in MATSim. Overall, despite many advances there is a considerable gap between the current MATSim capabilities and what is needed for the DVRP simulation. This paper will look at this gap in more detail.

3 The DVRP Optimizer

3.1 Properties and Possible Applications

The DVRP Optimizer is a program written in JAVA for solving the DVRP. The optimizer is intended to be as general as possible and to work on customizable instances of the DVRP. The program is constantly being developed and currently supports different versions of the DVRP that can be described as the Dynamic Multi-Depot Vehicle Routing Problem with Time Windows and Time-Dependent Travel Times and Costs. As of now it can be used for the one-to-many (many-to-one) vehicle routing, while the many-to-many problems are only supported if trips are not shared (like in taxi services). Currently, it is only possible to solve problems with hard time windows while soft time windows are not supported.

Concerning the dynamism of requests, the optimizer supports not only on-line submissions but also on-line modifications and cancellations by the customers. A request can also be rejected by the company due to, for instance, a shortage of

resources to serve it. However, operations other than submission of new requests are not included into benchmark instances as this leads to the open problem of results non-comparability (discussed by Flatberg et al. [17]). Thus, it is hard to evaluate the efficiency of the proposed algorithms provided that these operations are permitted.

The next enhancements are connected with the inclusion of time-dependent travel times and costs which are represented as functions of both the departure time from an origin vertex $t_{ij}^D(t)$ and the arrival time at a destination vertex $t_{ij}^A(t)$. Although it is common practice to provide travel time data dependent on the departure time, more elaborated strategies of dispatching vehicles require the provision of travel times depending on the arrival time. By default, the travel time data is provided in a form of 3D matrices, t_{ijk}^D or t_{ijk}^A respectively, where i, j denote a pair of origin-destination vertices while k is a time step within a considered time horizon (the time horizon is divided into time intervals of equal size, e.g. 15 minutes). The travel times are then approximated by linear interpolation from the input data. It is possible to supply only the travel times on the departure time t_{ijk}^D , in that case, the travel times on the arrival time are calculated according to the relation

$$t_{ij}^A(t + t_{ij}^D(t)) = t_{ij}^D(t) . \quad (1)$$

In order to obtain correct optimization results it is necessary that the input travel time data meets the FIFO property. It means that if two vehicles start from vertex i and go to vertex j , then the vehicle which starts earlier will always arrive earlier at the destination. Therefore, the input data is validated against the following constraints ($\epsilon > 0$):

$$t_{ij}^D(t + \epsilon) > t_{ij}^D(t) - \epsilon \quad (2)$$

$$t_{ij}^A(t + \epsilon) < t_{ij}^A(t) + \epsilon \quad (3)$$

Another important feature of the DVRP Optimizer is its flexibility in reassigning requests to vehicles. There exist four different pre-defined strategies for reassignment:

- *Never* – a request is never reassigned to another vehicle
- *Only planned routes* – a request may be reassigned to another vehicle only if a vehicle to which the request is already assigned has not started yet
- *Time horizon* – a request may be reassigned unless it is planned to be served within a specified time horizon (e.g. 2 hours)
- *Always* – a request assignment is subject to change as long as the assigned vehicle has not started out for serving it

Selection of the strategy depends on the problem properties. From the drivers' point of view, for example, the most preferable is the first option (*Never*) since they prefer less changeable routes. On the other hand, the more flexible the strategy is (e.g. *Always*) the more cost-effective and customer-oriented routes are. To achieve even higher customizability of the strategies, it is possible to

differentiate requests on whether it is a pickup or a delivery and then to apply a different reassignment strategy to each of these two types of requests.

Due to the considerable flexibility of the model, the current implementation of the DVRP Optimizer can be applied to a wide spectrum of the real-world DVRP examples, such as long-distance courier mail, distribution of heating oil, taxi services, feeder services for the Demand Responsive Transport systems (e.g. mini-buses) and others.

3.2 Algorithms

By default, the DVRP Optimizer uses a memetic algorithm, consisting of an evolutionary algorithm and a local search procedure, for solving the DVRP. Nonetheless, the current version of the system is open for extending it with other optimization approaches, such as tabu search or simulated annealing. The implemented memetic algorithm can be characterized in brief as follows:

- *Genotype coding* – for each vehicle a list of requests to be served
- *Crossing-over* – exchanges routes of a randomly chosen vehicle between two parent individuals; since exchanging routes may lead to invalidity of the both newly created genotypes, a repair procedure is executed removing doubled requests and adding missing ones
- *Mutation* – exchanges requests within/between routes
- *Selection for reproduction* – roulette wheel or tournament selection; both with fitness scaling functions
- *Succession* – with (e.g. elitist succession) or without overlapping populations
- *Termination criterion* – different possible criteria (that may be combined): generations count, convergence threshold, total fitness evaluations count etc.
- *Local search* – steepest ascent hill climbing algorithm with the neighbourhood search operator exchanging requests within/between routes

At the beginning, the memetic algorithm is run to get the initial solution for all *advance* (i.e. known a priori) requests. Then, each time the optimizer is notified of any change in the whole system (such as request modifications, travel times updates or vehicle breakdowns) only the local search procedure needs to be run. The DVRP Optimizer does not use insertion heuristics since their applicability is limited to those cases when only new *immediate* requests may appear while everything else in the system (e.g. already submitted requests, travel times/costs) is static. One should note that as long as changes in the system are not very dynamic, the local search algorithm is sufficient for updating solutions. However, when the dynamism is high, it may be necessary to re-run the whole memetic algorithm.

3.3 General System Architecture

In order to keep all necessary data up to date and to calculate valid and efficient routes, the DVRP Optimizer has to be coupled in a real-time mode with

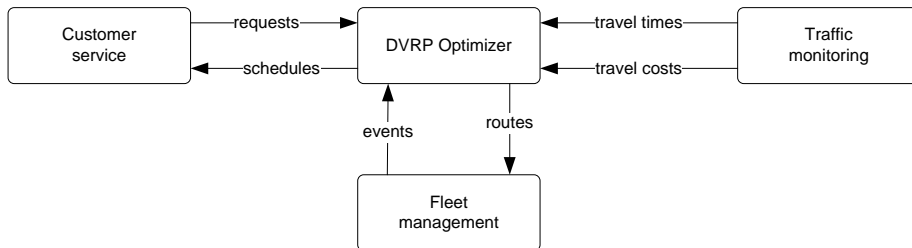


Fig. 1. Data flow between the system components

external modules (Fig. 1). Each of these modules is responsible for a specific domain of the routing process. The *customer service* module informs the optimizer about changes in the customer demand, the *fleet management* module notifies of changes in the state of the vehicle fleet, while the third one, *traffic monitoring*, provides the DVRP Optimizer with the current traffic data and calculates time-dependent shortest paths. Provided that the communication is carried out in real-time, each time any change in the system occurs, the DVRP Optimizer is instantly notified of it, which allows for immediate reaction to a new situation by replanning vehicle routes/schedules. As soon as new plans has been calculated, they are sent to the external modules.

4 MATSim and the DVRP Optimizer Integration

4.1 The Idea

Integrating the DVRP Optimizer with MATSim allows for the simulation-based evaluation of different dynamic vehicle routing algorithms as well as planning of various DRT services in MATSim. In both cases MATSim is responsible for the multi-agent simulation of the transport system with a high level of detail. Besides simulating traffic flow, MATSim serves as a platform for simulating a vehicle fleet and customer demand. Each time a change in demand, traffic conditions or fleet availability occurs, the DVRP Optimizer is informed about it and re-plans routes. After the routes are re-planned, the DVRP Optimizer sends new routes and schedules to MATSim.

The above-presented approach may be implemented in two steps:

1. *Off-line* optimization – combines the time-dependent travel times and costs resulting from simulation in MATSim with external data sources describing supply and demand (i.e. vehicles, depots, customers, requests, etc.); this step requires *sequential* execution of simulation in MATSim and then estimation of the time-dependent travel times/costs prior to the *off-line* optimization
2. *On-line* optimization – runs *concurrently* both the simulation and optimization processes that are integrated and interacting with each other.

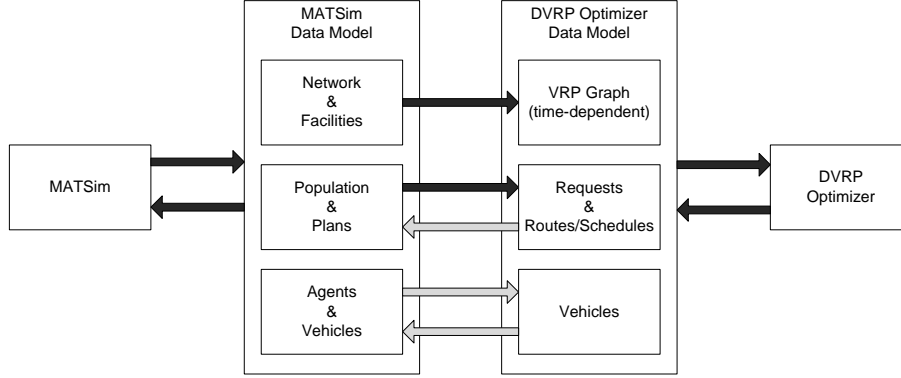


Fig. 2. Data flow between MATSim and the DVRP Optimizer

The data flow between MATSim and the DVRP Optimizer is presented in Fig. 2. The black flows are used in the *off-line* optimization, whereas the grey ones are additionally required for the *on-line* optimization. As of now, the first step, the *off-line* optimization, has been completed and is described later in this section.

4.2 Estimation of Time-Dependent Travel Times and Costs

Prior to the *off-line* optimization, the MATSim simulation results describing the time-dependent link travel times have to be converted into the time-dependent travel times/costs for the VRP graph according to the following steps:

1. Map depots and customers (requests) to links within the MATSim network model. These links form a set of vertices V in the VRP graph $G(V, A)$; a set of arcs A consists of all ordered pairs of the vertices.
2. For each arc (i, j) in A and for each time step k find the shortest-time (or least-cost) path in the MATSim network for a vehicle departing at time step k and calculate the corresponding travel cost c_{ijk}^D . This step is done by means of one of the shortest path search algorithms available in MATSim [18].
3. Based on Eq. 1 calculate travel times on the arrival time t_{ijk}^A . Alternatively, one may use a *backward* shortest path search, which gives more accurate results, but at the cost of longer computation time.

4.3 Results for the Off-line Optimization

In order to validate the above-presented approach and to assess the efficiency and correctness of the proposed algorithms, a set of tests was designed and carried out (Tab. 1). In the tests, four different networks and two kinds of services were used. Although all the networks were artificially created, their layouts are typical for urban areas, and the generated traffic is also characteristic for cities.

Table 1. Properties of the test cases

No	Network layout	Nodes	Links	Service type	Requests	Immediate requests	Pickup requests
1	grid	62	170	courier	100	49	49
2	custom	129	326	courier	100	52	52
3	spider's web	57	208	taxi	100	95	n/a
4	custom	93	276	taxi	100	96	n/a

Also the requests were designed in such a way that they would imitate real services within urban areas (e.g. the advance-to-immediate requests ratio, time windows, pickup-to-delivery ratio). The size of a network influences the speed of estimating time-dependent travel times/costs since the computation time of the shortest path search depends on the number of nodes and links; however, in case of the *off-line* optimization, it does not affect the the routing algorithm speed which is dependent (mostly) on the number of submitted requests.

The test procedure was carried out according to the following rules:

- Prior to the *off-line* optimization, for each test case the MATSim simulation was carried out, and then the time-dependent travel times and costs matrices were calculated with a 15-minute resolution. It was assumed that the travel costs are equal to the travel times.
- Tests instances were run both as the *static* and *dynamic* VRP.
- Each test instance was run with time-dependent travel times (*TD TT*) and with constant (a 24-hour average) travel times (*averaged TT*). In the latter case, the obtained results were then re-evaluated using the time-dependent travel times to show the real costs considering changeable road conditions.

Comparing the results for the *courier services* tests (Tab. 2), one can see that the precise knowledge of travel times leads to much better results. By knowing the time-dependent travel times, it is possible to arrange schedules in a way which minimises travelling along arcs at the moment they are at their highest-level of congestion. Moreover, it is very likely that solutions calculated with the averaged travel times may not satisfy all the constraints; in the experiments, almost all solutions violated the time window constraints, and in extreme cases, the total lateness (being after a time window is closed) was almost 1.5 hours. Comparing *dynamic* vs. *static* mode, it is clear that, in case of courier services, having known all the requests in advance, one can yield better results.

Different results were obtained for the *taxi services* tests. First of all, the solutions found for the averaged and the time-dependent travel times are similar concerning the cost of travelling. This is due to the properties of taxi services – taxi services work more in a request-response manner with narrow time windows for serving requests, which makes it impossible to flexibly change the order of requests and thus to avoid traversing arcs when they are congested. This is also one of the reasons of smaller differences when comparing *dynamic* vs. *static* mode than it was for the courier services test cases. However, the real difference lies in constraint satisfaction – the solutions obtained with the averaged travel times

Table 2. Best results obtained for the test cases

No	Type	TD TT	Averaged TT	
1	static	13728	17192	(+25,2%)
	dynamic	14766	17342	(+17,4%)
2	static	22435	26351	(+17,5%)
	dynamic	23879	28133	(+17,8%)
3	static	54017	53778	(-0,4%)
	dynamic	54307	54327	(+0,0%)
4	static	95579	96406	(+0,9%)
	dynamic	96451	97158	(+0,7%)

were always violating some of the time windows. This means that when using averaged travel times, some of the customers were always served too late.

Since the tests were carried out for instances created specifically for traffic simulation in MATSim, it is impossible to directly compare the results of this research with other studies, however, some similarities may be shown. Both Ichoua et al. [19] and Van Woensel et al. [20], using popular VRP test instances, proved that knowledge of accurate travel times results in substantial improvement of solutions. The scale of this improvement depends on many factors, including the level of time-dependency. In the second study cost decrease was 25% on average, which is similar to the outcomes for the courier services (Tab. 2).

5 Conclusions and Future Work

In this paper, an application of MATSim as a multi-agent simulation platform for DVRP optimization is presented. The approach takes into account several factors which are essential for finding optimal routes in the real world, like traffic in urban areas, and which are missing in the classical VRP models. The results show that the lack of accurate travel-time data may lead not only to higher costs, but also to numerous violations of time-window constraints, which in the end results in the deterioration of service quality.

The implemented *off-line* optimization approach is an important step towards the full integration of MATSim with the DVRP Optimizer. The authors work on creating an *on-line* optimization system that will serve for planning taxi and Demand Responsive Transport services in MATSim, according to the idea presented in Section 4.1. Additionally, it will be possible to adapt such a system to serve as a simulation framework for carrying out very detailed evaluations of different dynamic time-dependent vehicle routing algorithms. Both applications seem very important, appealing and timely.

References

1. Toth, P., Vigo, D. (eds.): The Vehicle Routing Problem. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2001)
2. Golden, B., Raghavan, S., Wasil E. (eds.): The Vehicle Routing Problem: Latest Advances and New Challenges. Springer, NY (2008)

3. Zeynepkis, V., Tarantilis, Ch.D., Giaglis, G.M., Minis, I. (eds.): *Dynamic Fleet Management: Concepts, Systems, Algorithms & Case Studies*. Springer, NY (2007)
4. Cordeau, J-F Gendreau, M., Laporte, G., Potvin, J-Y, Semet, F.: A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 53, 512–522 (2002)
5. Regan, A.C., Mahmassani, H.S., Jaillet, P.: Evaluation of Dynamic Fleet Management Systems: Simulation Framework. *Transportation Research Record*, 1645, 176–184 (1998)
6. Taniguchi, E., Thompson, R.G., Yamada, T., Duin van, R.: *City logistics – Network Modelling and Intelligent Transport Systems*. Pergamon, Amsterdam (2001)
7. Barcelo, J., Grzybowska, H., Pardo, S.: Vehicle routing and scheduling models, simulation and city logistics. In: Zeynepkis, V., Tarantilis, Ch.D., Giaglis, G.M., Minis, I. (eds.): *Dynamic Fleet Management: Concepts, Systems, Algorithms & Case Studies*, pp. 163–195. Springer, NY (2007)
8. Liao, T-Y, Hu, T-Y, Chen, D-J: Object-Oriented Evaluation Framework for Dynamic Vehicle Routing Problems Under Real-Time Information. 87th TRB Annual Meeting, Washington (2008)
9. Maciejewski, M.: A comparison of microscopic traffic flow simulation systems for an urban area. *Transport Problems*, 5(4), 27–38 (2010)
10. Maciejewski, M.: Parametric Calibration of the Queue-Based Traffic Flow Model in MATSim. In: Janecki, R., Krawiec, S.: *Contemporary Transportation Systems. Selected Theoretical And Practical Problems. Models of Change in Transportation Subsystems*, pp. 217-226. Wydawnictwo Politechniki Slaskiej, Gliwice (2011)
11. Balmer, M., Meister, K., Rieser, M., Nagel, K., Axhausen, K.W.: Agent-based simulation of travel demand: Structure and computational performance of MATSim-T. 2nd TRB Conference on Innovations in Travel Modeling, Portland (2008)
12. Dobler., C.: Implementations of within day replanning in MATSim-T. *Arbeitsbericht Verkehrs- und Raumplanung* 598, IVT, ETH Zurich (2009)
13. Rieser, M.: Adding transit to an agent-based transportation simulation concepts and implementation. PhD thesis, TU Berlin, (also: VSP WP 10-05) (2010)
14. Neumann A., Nagel, K.: Avoiding bus bunching phenomena from spreading: A dynamic approach using a multi-agent simulation framework. VSP Working Paper 10-08, TU Berlin, Transport Systems Planning and Transport Telematics (2010)
15. Schröder, S., Zilske, M., Liedtke, G., Nagel, K.: Towards a multi-agent logistics and commercial transport model: The transport service provider’s view. In: 7th International Conference on City Logistics, CITY LOGISTICS 2011, Mallorca Island, Spain, June 7-9 (2011)
16. Ciari, F., Schüssler, N., Axhausen, K.W.: Estimation of car-sharing demand using an activity-based microsimulation approach. Working Paper, 632, IVT, ETH Zurich (2010)
17. Flatberg, T., Hasle, G., Kloster, O., Nilssen, E.J., Riise, A.: Dynamic and stochastic vehicle routing in practice. In: Zeynepkis, V., Tarantilis, Ch.D., Giaglis, G.M., Minis, I. (eds.): *Dynamic Fleet Management: Concepts, Systems, Algorithms & Case Studies*, pp. 41–63. Springer, NY (2007)
18. Lefebvre, N., Balmer, M., Axhausen, K.W.: Fast shortest path computation in time-dependent traffic networks. Working Paper, 439, IVT, ETH Zurich, Zurich (2007)
19. Ichoua, S., Gendreau, M., Potvin, J-Y: Vehicle dispatching with time-dependent travel times. *European Journal Of Operational Research*, 144(2), 379–396 (2003)
20. Van Woensel, T., Kerbache, L., Peremans, H., Vandaele, N.: Vehicle routing with dynamic travel times: A queueing approach. *European Journal Of Operational Research*, 186(3), 990–1007 (2008)