

A Tool for Simulating Demand Responsive Transport Systems in SUMO

1st Maria Giuliana Armellini
Institute of Transportation Systems
German Aerospace Center (DLR)
Berlin, Germany
Maria.Armellini@dlr.de

Abstract—Demand responsive transport (DRT) has become very important in the last years and is being tested in different cities worldwide. Due to the complexity and diversity of these services, their planning is quite challenging. This paper introduces a tool to simulate different types of DRT services. The open source micro-simulation Eclipse SUMO (Simulation of Urban MObility) is used as a framework. The tool was written in Python 3 and calculates the best routes for each DRT vehicle based on information about requests, vehicle fleet and network. As test case, multiple requests between a peri-urban area and the city center of Brunswick (Germany) were simulated for different DRT services. The results show the practicality of the tool as a form of analysis and planning of DRT systems.

I. INTRODUCTION

In recent years, technological advances and improvements in computer power and digitization have made it possible to develop new forms of demand-based mobility, which are a booming market and are already being used or tested in several cities worldwide. Demand Responsive Transport (DRT) also referred to as ride-sharing services like "UberPool" and "Lyft Shared" are an example of latter. This shared service without fixed routes seeks to bundle requests in minimizing the number of vehicles and route lengths without compromising passenger travel times. Resulting, according to various simulations, in a more efficient service compared to taxi and ride-hailing services ("Uber" or "Lyft") [1]–[3]. A significant impact on vehicle mileage and traffic in general only occurs if many customers switch from individual car-based transport. According to Feigon et al. [4], only New York City has so far published sufficient data on DRT systems to analyze and evaluate their impact. Based on the latter data, Schaller [5] found that in fact only 20% of the trips are shared and that the majority of the customers switched from non-vehicle-based modes of transport (e.g. public transport, bicycle and walking). Additionally most of the times the service is only used by one person, which leads to an increase in traffic instead of the planned reduction. The acquisition of passengers from public or non-motorized transport is a critical point, since DRT systems are not well suited for high-demand connections [6]. Conventional high capacity public transport, such as trains, subways or Bus Rapid Transit (BRT) are best suited for this

purpose due to their higher operational efficiency [7]. Hence, the combination of both systems by using the DRT as a feeder system for high capacity transit would be the first best solution.

DRT systems vary not only if they work as feeder system of transit or as a unique system. The results vary significantly depending, for example, on the capacity and number of vehicles, whether requests can be booked in advance or whether the service is provided door-to-door or with virtual stops. The results also vary depending on how the routes are calculated, which is referred to as the Dial a Ride Problem (DARP). According to [8], the DARP generalizes a number of vehicle routing problems such as the vehicle routing problem with pickups and deliveries (PDVRP) and the VRP with time windows (VRPTW). The main difference between this problem and other routing problems is that the DARP transports passengers instead of goods. Therefore other constraints must be contemplated to ensure the quality of the service from a human perspective. Due to the wide and varied constraints that a DARP can present, there are different methods and models for solving them either approximately or exactly. Another important classification is between static and dynamic solution. In a static DARP, all requests are known in advance, meanwhile in the dynamic DARP requests are revealed and managed in real time [9].

Several simulation programs such as Eclipse SUMO (Simulation of Urban MObility) and MATSim (Multi-Agent Transport Simulation) have been further developed to allow the simulation of DRT services. However, to the best knowledge of the author, they are generally very specific and do not allow the evaluation of different DRT services. Over the last months, a new element called Taxi Device has been developed to allow the simulation of on-demand services with different dispatching algorithms in SUMO. However, at this moment, the device does not yet allow for real-time modification of the route, which is indispensable for evaluating shared DRT services.

This paper introduces a Python tool to simulate different types of DRT services in SUMO [10]. The tool is written in Python and calculates the best routes by solving an static DARP. To test the tool, a set of requests between a peri-urban area and the city center of Brunswick (Germany) were simulated for different DRT services. The paper is organized as follows. First a description of the developed tool and the

method used to solve the DARP is given. This is followed by a summary of the different DRT services that can be simulated within the tool. Finally, the simulation results are presented and discussed.

II. METHODOLOGY

To be able to simulate DRT systems in SUMO, a Python tool was developed, which allows to communicate with the simulation framework and calculates the routes for the DRT vehicles and person. As output, the founded routes for each vehicle and person are written in SUMO format, what allows to use them for further simulations. SUMO proves to be the best option in this context, since it allows the simulation of large road networks with different modes of transportation on a microscopic level and its open source license allows the implementation and testing of new algorithms.

To model the DRT services, the exact solution method developed by [11] was taken as benchmark and extended to allow for more service options and a coupling to SUMO. In the following, the basic problem is explained. Afterwards the more complex configurations will be listed.

The algorithm assigns a set of n requests $R = \{r_1, \dots, r_n\}$ to a set of m vehicles $V = \{v_1, \dots, v_m\}$, in order to minimize a cost function V and satisfy a set of constraints Z , allowing multiple passengers per vehicle. All travel requests are known in advance and consist at least of the desired pick-up time, a pickup location and a drop-off location. Each vehicle of the fleet has a capacity ν and a depot location, where the vehicle starts and ends the service. The locations are defined as a position on a lane (edge) in the SUMO network.

The algorithm consists of three steps: (i) calculating the shareability graph and (ii) the feasible trips graph and finally (iii) solving an integer linear programming (ILP) to find the optimal routes. In sections II-A to II-C this steps will be described in detail.

A. Shareability graph

The first step of the algorithm is to compute a pairwise shareability graph with all possible combinations between a vehicle and a request and between two requests. As the DRT system can be shared, a combination between two requests can be by combining the pick-ups or drop-offs of both requests or by combining the pick-up of a request and the drop-off of the second. The pair corresponding to the pick-up and drop-off of the same request is also considered.

For each pair, the number of passengers (+1 for pick-up and -1 for drop-off) and the travel time are stored. The travel time consists of the sum of the service time at the stop (15 seconds as default) and the minimum travel time, which is estimated using the SUMO application DUAROUTER.

Initially only the travel times for each request (pick-up to drop-off) are estimated, being comparable to the direct travel times. Based on this time and a direct route factor (drf), the time windows for pick-up and drop-off are defined. The direct route factor is the maximum relation between the travel time with the DRT and the direct travel time. As default, a drf of 2

is taken. The time window for the request pick-up and drop-off are defined as:

- earliest pick-up = desired pick-up time,
- earliest drop-off = earliest pick-up + direct travel time,
- latest drop-off = earliest pick-up + direct travel time * drf,
- latest pick-up = latest drop-off - direct travel time.

Once the time windows are set, the travel times for all pairs are estimated and the feasibility of each pair is verified. A pair vehicle-request pick-up is feasible if the start time of the vehicle service plus the travel time is smaller than the request pick-up time. A pair request drop-off -vehicle is feasible if the request pick-up time plus the travel time is smaller than the end time of the vehicle service. Two requests can be combined if the time windows of both requests are compatible. For example, for the pair request 1 drop-off - request 2 pick-up is feasible if the earliest drop-off time of request 1 plus the pair travel time is smaller than the latest pick-up of request 2.

If a pair is not feasible, it will be deleted of the shareability graph.

B. Feasible trips graph

Next, the cliques of the shareability graph are explored to find feasible trips for each vehicle. A trip $T = \{r_1, \dots, r_{n_T}\}$ is a set of n_T requests transported by a specific vehicle. A trip is feasible if:

- all requests are picked up and dropped off,
- time windows of each request are satisfied,
- vehicle returns to the depot on time,
- the vehicle capacity is never surpassed and
- the maximum waiting time (default 180 seconds) for a request pick-up is not exceeded if passengers are in the vehicle.

The time windows are checked by adding the travel times of each pair and the capacity by adding the number of passenger of each pair.

The algorithm to compute the feasible trips proceeds incrementally in trip size for each vehicle, starting from the vehicle-request pair in the shareability graph. If two vehicles have the same characteristics, the exhaustive search to compute the feasible trips graph will be performed only once.

Depending on the number of requests and the shareability of them, the exhaustive search may take a long time and memory. To allow for a faster (but not exact) solution the search time for the trips of a determined size can be limited.

C. Integer linear programming

Finally, the optimal trips are calculated by solving an integer linear programming (ILP) with the Python LP modeler PuLP [12].

A binary variable $\epsilon_{i,j} \in \{0, 1\}$ is defined for each trip $T_i \in \tau$ of a vehicle $v_j \in V$. If $\epsilon_{i,j} = 1$, then the trip T_i of vehicle v_j is selected. An additional binary variable $\rho_{i,n} \in \{0, 1\}$ is defined for each request $r_n \in R$ in trip $T_i \in \tau$. If the trip T_i is selected and therefore the request r_n is served, the variable $\rho_{i,n} = 1$. If a request $r_n \in R$ is not served by any of the selected trips, then $\sum_{i \in T} \rho_{i,n} = 0$.

The objective of the ILP is defined in 1. Three cost terms are considered. The cost $c_{i,j}$ is the calculated travel time (service time at stops included) for the trip T_i with the vehicle v_j . The cost c_{ko} is a large constant to penalize the rejection of requests. Finally, a constant and small cost c_b is introduced to prevent the use of several vehicles, if the requests can be served with fewer vehicles at comparable costs.

$$\sum_{optimal} := \operatorname{argmin} \sum_{i \in T, j \in V} (c_{i,j} + c_b) \epsilon_{i,j} - c_{ko} \sum_{i \in T, n \in R} \rho_{i,n} \quad (1)$$

$$\sum_{i \in T, j \in V} \epsilon_{i,j} \leq 1 \quad \forall v_j \in V \quad (2)$$

$$\sum_{i \in T, n \in R} \rho_{i,n} \leq 1 \quad \forall r_n \in R \quad (3)$$

$$\sum_{j \in V} \sum_{i \in T} \epsilon_{i,j} \leq 1 \quad \forall v_j \in V \quad (4)$$

Two types of constraints are included. Constraint 2 imposes that each vehicle is assigned to one trip at most and Constraint 3 imposes that each request is assigned to a single trip or ignored. The last constraint 4 only ensures that at least one trip is defined.

The trips found by the ILP are then written into a summary file with information about each vehicle, the request and the total DRT system. Another file with the trips written in SUMO format is given for further simulations.

D. Variations of the DRT service

As mentioned before, the tool is able to simulate different DRT services. In section II the model for a basic DRT shared system was explained. By adding extra information or changing some parameter, different variations of the system can be obtained. These are listed below:

- Requests admit the following extra parameters: number of passengers, number of wheelchair passengers, explicit time window for pick-up, explicit time-window for delivery and value for drf.
- Vehicles admit the extra parameters: vehicle cost, number of wheelchair spaces, service area, start and end time of service.
- Public transport services (route, schedule and stops) that can be served by the DRT, now working as a feeder system.
- Option for search the minimum number of vehicles to serve all requests.
- Maximum time in seconds for search trips of the feasible trips graph, allowing a fast (but not exact) result.
- Maximum waiting time for passengers on the vehicle to pick-up another request.

III. TESTING SCENARIOS

For a demonstration of the proposed DRT tool, different DRT systems were simulated in a real-world study case. The study area includes six villages located in the west of the city of Brunswick. The area shows the characteristic mobility problems of peri-urban and rural areas, such as high percent of commuters traffic, limited local supply and limited public transport. This study area has been analyzed by [13] and used in some simulation studies [14] but with a more simpler DRT algorithm.

In [13], [14] the implementation of a Bus Rapid Transit (BRT) line with a frequency of 15 minutes to serve the high demand between the peri-urban area and the city center was evaluated with good results. This BRT line will be contemplated for the simulation scenarios as the only transit service in the area. Fig. 1 shows the study area and the BRT line and its mobility hubs.

All test scenarios are simulated for 55 requests arriving between 18:30 and 19:30 during the afternoon peak hour. The travel requests have either origin or destination in the study area. The simulations were carried out in standard modern laptop with an Intel i7 core processor. Each scenario is described below.

1) *Scenario 1*: The basis scenario contemplates a DRT feeder to the BRT line. The DRT service has a fleet of three vehicles with a capacity of 3 passengers each. The service is door-to-door and the vehicles operate in the entire area, being able to serve any of the three mobility hubs. Passengers could be transported only if the total travel time from origin to destination does not exceed the travel time by car by 2.5 times. This is equivalent to a drf of 2.5.

2) *Scenario 2*: This scenario search the number of vehicles needed to serve all requests of scenario 1.

3) *Scenario 3*: In this case three different DRT service with 2 vehicles each will be simulated. The first DRT service operates in the area of Vechelde (pink area in Fig. 1), the second in the area of Denstorf (green area) and the last DRT service can only serve request from/to the area of Lamme (blue area). Each DRT can only serve the respective mobility hub of the area.

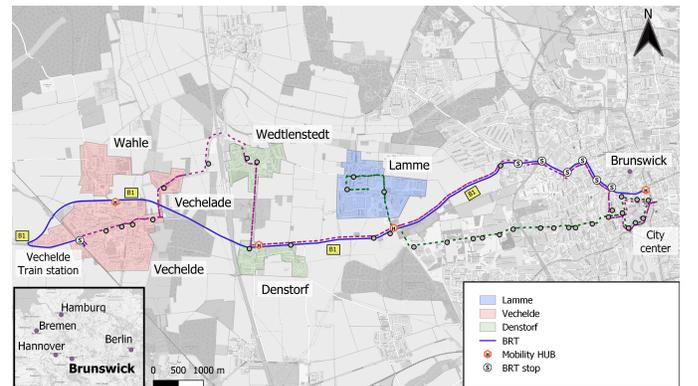


Fig. 1. Study case (Background map from Google Map)

4) *Scenario 4*: For this scenario the maximum computing time was incremented to see if relevant changes in the routes were made and on which computing time cost.

5) *Scenario 5*: In this case, a higher drf was used to allow later arrival times. A higher drf does not implied directly longer routes, but the possibility of contemplate them.

6) *Scenario 6*: The DRT service will no longer be used as feeder system for the BRT line, meaning no combination with public transport. The vehicles have now a capacity of 12 passengers and the required fleet will be searched. Passengers can be picked up between an interval of half an hour. This is simulated by using time windows for the pick-up.

A. Results

Table I summarizes the relevant results for each scenario. As it is shown, the results vary significantly. In the basis scenario 1, in which the fleet counts with only three vehicles with a capacity of 6 passengers, 39 of 55 requests must be rejected. 28 of them, could not be served, because there where no possible combination with the BRT line. This means, that by taken the next bus, the arrival time to final destination would surpass the possible arrival time with the private car by the given drf of 2.5. The rest 11 requests could not be served because of fleet limitations.

The scenario 2 shows that the necessary number of vehicles to served all incoming requests is 5. This kind of simulation is important to estimates the required fleet for peak hours and the fleet for low demand hours. Having both results, the evaluation of the service can be carried out with a specific fleet.

Based on the results of scenario 3, when DRT vehicles operate in a single area (scenario 1) the service is more efficient. The total trip duration and the total waiting time are considerably lower for scenario 1 and using one less vehicle. The only positive difference for this scenario is that only 33 requests instead of 39 were rejected.

In the scenario 4, the maximum step time to calculate the shareability graph (see section II-A) was incremented from 5 to 20 seconds. This means a total increase in calculation time of only 5 minutes. The results are nevertheless comparable. Although the travel times were reduced by 8%, the number of requests transported remained constant.

For the scenario 5, the drf was incremented, allowing more connections with the BRT line to take place. The travel times for the vehicles were not significantly changed. The travel times for each request show slight increments of less than 5 minutes. What has changed significantly is the variation between the desired departure time given by the passenger at request time and the actual pick-up time.

Scenario 6 shows the results of implementing the DRT as a unique system, with no connection to public transport. For this, a total of 13 vehicles are needed. Regarding the implementation of larger vehicles with a capacity of 12 passengers, the system was not able to accommodate more than 4 passengers at the same time. Comparing the travel times of all vehicles with the scenario 2, there is an relevant increase, which implies an increase in the mileage.

In this study case, the use of DRT as a feeder system operating in an unique service ares proves to be the best option. Fig. 2 shows a picture of the simulation of scenario 1 in SUMO. The DRT vehicles are shown in red and the green buses represent the BRT vehicles. Requests are modeled as person elements and are represented as the blue triangles. The further simulation in SUMO allows to calculate more results, such as emissions, time losses and stopped times.

IV. CONCLUSION AND FUTURE WORK

Technological advances and improvements in computer power and digitization have made the implementation of DRT systems possible in real life, making them a booming market. Due to the wide range of possible services and therefore variation in results, these services pose a problem for the planner. The use of micro-simulations to evaluate different service options can contribute to their optimization.

This paper presents a tool that allows the simulation of a wide range of DRT configurations using the micro-simulation Eclipse SUMO as a framework. The option of choosing between an approximate or an exact solution of the DARP makes it possible to obtain fast or optimal results depending on the user's needs.

The limitations of the tool are related to the complexity of solving static and exact DARP. The possible future extension of the software SUMO to allow dynamic route changes of DRT vehicles (in SUMO called Taxi devices) would allow the extension of the present tool for the dynamic case. This would reduce calculation times significantly, as the number of combinations between requests would decrease.

REFERENCES

- [1] J. Bischoff, M. Maciejewski, and K. Nagel, *City-wide shared taxis: A simulation study in Berlin*. IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017. [Online]. Available: <https://svn.vsp.tu-berlin.de/repos/public-svn/publications/vspwpl/2017/17-11/BischoffMaciejewskiNagel2017SharedTaxiITSC.pdf>
- [2] M. Lokhandwala and H. Cai, *Dynamic ride sharing using traditional taxis and shared autonomous taxis: A case study of NYC*. Transportation Research Part C: Emerging Technologies, Dec. 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0968090X18307551>
- [3] J. Schwieterman and C. Smith, *Sharing the ride: A paired-trip analysis of UberPool and Chicago Transit Authority services in Chicago, Illinois*. Journal of the Transportation Research Forum (57), Nov. 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0739885918302683>
- [4] S. Feigon, C. Murphy, and T. McAdam, *Private Transit: Existing services and emerging directions*. Washington, DC: The National Academies Press, 2018. [Online]. Available: <https://www.nap.edu/catalog/25020/private-transit-existing-services-and-emerging-directions>
- [5] B. Schaller, *The New Automobility: Lyft, Uber and the Future of American Cities*, 2018, available at <http://www.schallerconsult.com/rideservices/automobility.pdf>.
- [6] J. Mageean and J. Nelson, *The evaluation of demand responsive transport services in Europe*. Journal of Transport Geography (11), 2003.
- [7] M. Mörner, *Sammelverkehr mit autonomen Fahrzeugen im ländlichen Raum*. Darmstadt: Dissertation. Technischen Universität Darmstadt, 2018. [Online]. Available: http://tuprints.ulb.tu-darmstadt.de/7483/9/M%C3%B6rner_Sammelverkehr_mit_autonomen_Fahrzeugen_im_%C3%A4ndlichen_Raum.pdf
- [8] J.-F. Cordeau and G. Laporte, "The dial-a-ride problem (darp): Models and algorithms," *Annals OR*, vol. 153, pp. 29–46, 06 2007.

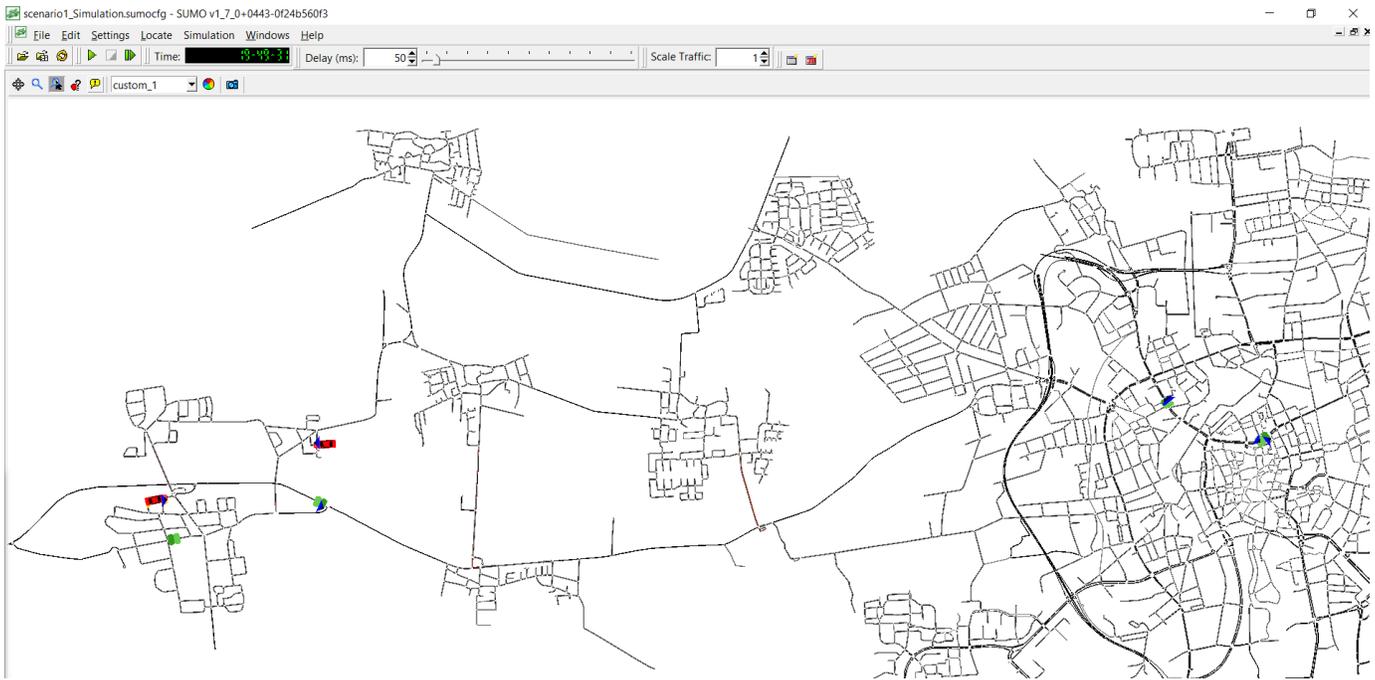


Fig. 2. Scenario 1 in SUMO

TABLE I
RESULTS OF SIMULATION SCENARIOS

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Number of Vehicles [veh.]	3	5	4	3	3	13
Transit connection not possible [req.]	28	28	32	28	19	0
Total of requests rejected [req.]	39	28	33	39	32	0
Total duration [min.]	95	162	88	87	91	207
Total waiting time [min.]	8	79	101	14	3	0
Computing time [min.]	2.2	3.3	1.0	7.3	3.7	10.4

- [9] Y. Molenbruch, K. Braekers, and A. Caris, "Typology and literature review for dial-a-ride problems," *Annals of Operations Research*, vol. 259, no. 1-2, pp. 295–325, 2017.
- [10] P. Alvarez Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: <https://dlr.de/124092/>
- [11] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," Jan. 2017. [Online]. Available: <http://www.pnas.org/content/114/3/462>
- [12] S. Mitchell, M. O'Sullivan, and I. Dunning, *PuLP : A Linear Programming Toolkit for Python*, 2011. [Online]. Available: <https://github.com/coin-or/pulp>
- [13] M. G. Armellini, "Optimierung der buslinie 450 in braunschweig durch on-demand-zubringer," Master's thesis, Fachhochschule Münster, 2019.
- [14] M. G. Armellini and L. Bieker-Walz, "Simulation of a demand responsive transport feeder system: A case study of brunswick," *Proc. SUMO User Conference 2020*, submitted for publication.