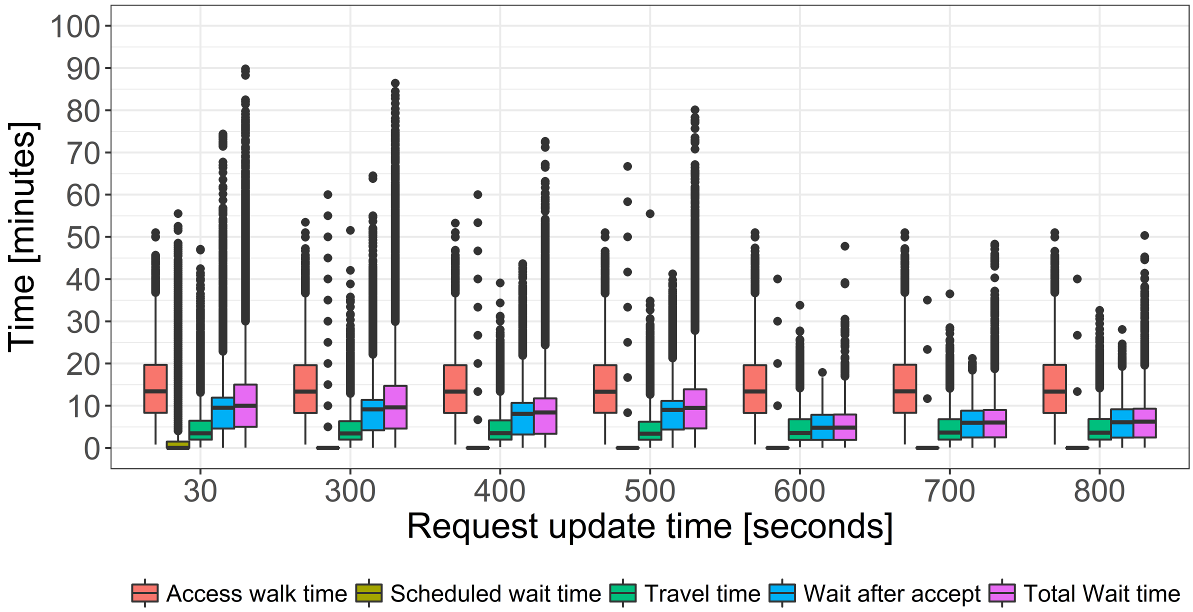


Simulation of Autonomous Transit On Demand for Fleet Size and Deployment Strategy Optimization

Biyu Wang

Master thesis

Institute for Transport Planning and Systems **January 2018**

MATSim
Multi-Agent Transport Simulation

(FCL) FUTURE CITIES LABORATORY 未来城市实验室

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

IVT Institut für Verkehrsplanung und Transportsysteme
Institute for Transport Planning and Systems

Contents

List of Figures

List of Tables

Master thesis

Simulation of Autonomous Transit On Demand for Fleet Size and Deployment Strategy Optimization

Biyu Wang
IVT
ETH Zürich

CH-8093 Zürich
phone: +86 13581662461

wangb@student.ethz.ch

January 2018

Abstract

Autonomous transit on demand (ATOD) is a potential future public transit mode, which appeals to a lot of researchers and policy-makers. In the project, ATOD is simulated in MATSim to explore the optimal fleet size and deployment strategy to help policy-makers to decide how to introduce the new transport system in the future. The simulation enables the system to explore the optimization automatically under specific constraints with the MATSim evolutionary algorithm.

Keywords

Autonomous transit on demand; MATSim; Agent-based modeling

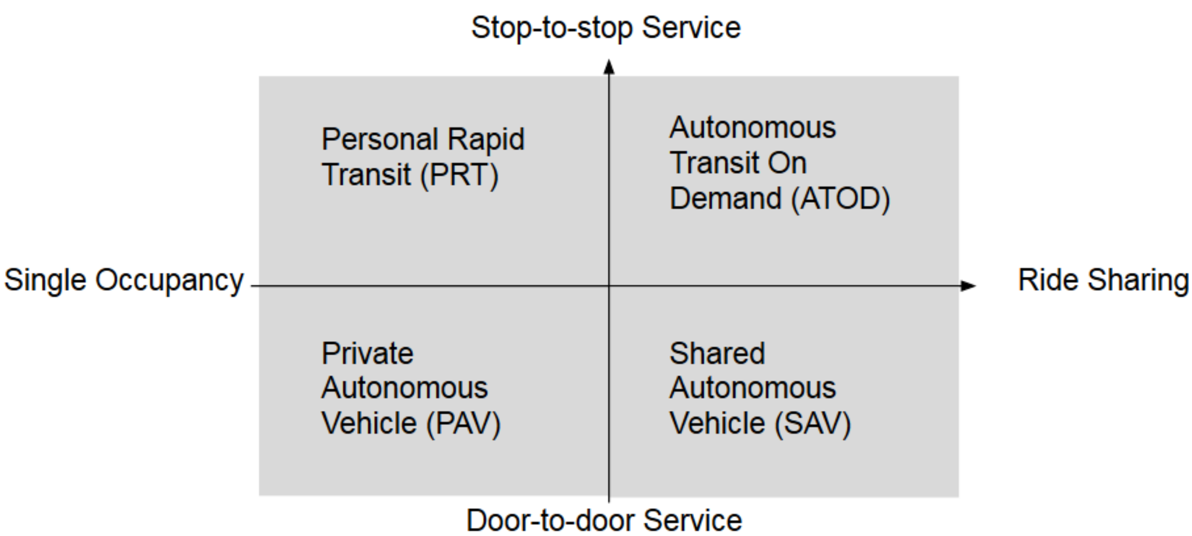
Preferred citation style

Wang, B. (2018) Simulation of Autonomous Transit On Demand for Fleet Size and Deployment Strategy Optimization, *Master thesis, Institute for Transport Planning and Systems*, ETH Zurich, Zurich.

1 Introduction

Public transit is a significant transport mode worldwide but still faces some challenges. On the one hand, from passenger's perspective, issues such as detouring, the uncertainty of wait time and frequent transfers makes public transit inconvenient or unattractive. On the other hand, from operator's perspective, low occupancy may lead to economic loss. With the development of autonomous vehicles (AVs), public transit may step into a new era. Some researchers treated AVs as an application of shared economy and a promising alternative to conventional taxi service. They simulated AVs as shared autonomous vehicles (SAVs) with agent-based modeling, a dynamic door-to-door ride-sharing service. Several simulations of SAV were recently done for Austin (Fagnant and Kockelman, 2015b), Berlin (Maciejewski and Bischoff, 2017) and Zurich (Boesch et al., 2016) among others. Some researchers argued that AV may become a substitute for conventional public transit, which offers dynamic, reliable and comfortable stop-to stop service. Some simulated the personal rapid transit (PRT), which offers stop-to-stop service on demand with a single passenger on board. For example, Chebbi and Chaouachi (2015) simulated PRT with agent-based modeling. In this thesis, the autonomous transit on demand (ATOD) system is implemented, simulated and analyzed, which consists of stop-to-stop shared rides in minibuses following dynamic routes and responding to requests. The difference and the relationship of the private autonomous vehicle (PAV), SAV, PRT and ATOD can be found in Fig. 1.

Figure 1: The definition of PAV, SAV, PRT and ATOD based on vehicle occupancy and type of origin and destination



AV is widely acknowledged as potential future transport mode which may start a new revolution of public transport. From the point of view of transport, the main difference between AV and the traditional vehicle is that AV can be precisely and centrally controlled. First, Human drivers vary in driving behaviors, which is influenced by both internal physical and external environmental factors. Therefore it is always difficult to simulate and predict the actual traffic situation precisely. Nevertheless, AV is controlled by the machine with few variance of traffic behavior. From a microscopic point of view, precisely controlled AV will increase the capacity of the road through avoiding stop-and-go traffic, interactive intersection without the fixed traffic light, platooning, etc., which are impossible for human drivers due to safety issues. From a macroscopic point of view, AV enables dynamic, precisely and real-time network control and prediction, offering more reliable public service with more flexibility. Second, the simulation of transit on demand can also be implemented theoretically with human-driven vehicles, but the human drivers can hardly be controlled by centralized system precisely for the optimization of system performance. Nowadays, services such as Uber or other dynamic ride-sharing software offers similar service as PAV or SAV with the human driver, but these platforms cannot replace public transport service. One reason is that drivers always try to maximize the profit of themselves, which may harm the systematical benefit. For example, some rural area cannot be covered by this service because the cost is too high for drivers to benefit from such journey. Also, different drivers have their criteria for accepting the request, which is not as reliable and predictable of public transport. The intervention of various travel behavior may lead to a more various and unpredictable performance of the system. The Uber equilibrium can reach user equilibrium but hardly reach the system optimal, where the benefit of some vehicles may be sacrificed. Therefore, it is convincing that the dynamic transit on demand should be introduced into cities with AV but not the vehicle with human drivers.

For the policy makers and operators, optimal fleet size and deployment strategies are the main concerns. Several studies are conducted to define optimal fleet size with both analytical and simulation approaches. (See more details in Section 2) However, few researchers consider the influence of fleet size and deployment strategies on vehicle occupancy, which is one of the crucial characteristics of public transit. The thesis explores the optimal fleet size and deployment strategies of ATOD, considering the benefit of ride sharing. The purpose is to help operators, authorities and planners decide the appropriate fleet size and the deployment strategies of ATOD to satisfy the demand with high-occupancy vehicles. In addition, Sioux Falls scenario is selected in the thesis, which is an easy-computing but complete enough scenario for a meaningful and reasonable result. The base scenario together with tested scenarios of different input population file and different AV-related parameters are simulated for Sioux Falls, aiming at exploring the sensitivity and robustness of the model and the influence of different parameters on the result.

The thesis will be structured as follows: Section 1 is a brief introduction ; Section 2 will review the previous research on AV and optimal fleet size; Section 3 introduces the framework of the simulation, including the introduction to the selected simulation platform and corresponding extension, MATSim (Multi-agent Travel simulator) and DRT (Demand Responsive Transit), and the overview of the simulation; Section 4 shows in details the implementations of the project, on-demand vehicles deployment, two new modes: new DRT call and DRT request, the incentive to limit the fleet size, a cost-based routing module, as well as the modification of existing modules, such as walking scoring function, dwelling time and request accepting constraints; Section 5 introduces the overview of simulated scenarios with different configurations for the project; Section 6 analyzes the result of different scenarios from the perspective of computation time, mode share, score, the experience of passengers, number of passengers per vehicle, vehicle occupancy, fleet size and fleet deployment; Section 7 concludes and discusses the possible improvement of the project and the potential future development of ATOD.

2 Literature review

2.1 Autonomous vehicle(AV)

Figure 2: Definition of Levels of Automation in SAE International Standard J3016

SAE level	Name	Narrative Definition	Execution of Steering and Acceleration/ Deceleration	*Monitoring* of Driving Environment	Fallback Performance of *Dynamic Driving Task*	System Capability *(Driving Modes)*
***Human driver* monitors the driving environment**						
0	**No Automation**	the full-time performance by the *human driver* of all aspects of the *dynamic driving task*, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	**Driver Assistance**	the *driving mode*-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	Human driver and system	Human driver	Human driver	Some driving modes
2	**Partial Automation**	the *driving mode*-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	**System**	Human driver	Human driver	Some driving modes
***Automated driving system* ("system") monitors the driving environment**						
3	**Conditional Automation**	the *driving mode*-specific performance by an *automated driving system* of all aspects of the dynamic driving task with the expectation that the *human driver* will respond appropriately to a *request to intervene*	System	**System**	Human driver	Some driving modes
4	**High Automation**	the *driving mode*-specific performance by an automated driving system of all aspects of the *dynamic driving task*, even if a *human driver* does not respond appropriately to a *request to intervene*	System	System	**System**	Some driving modes
5	**Full Automation**	the full-time performance by an *automated driving system* of all aspects of the *dynamic driving task* under all roadway and environmental conditions that can be managed by a *human driver*	System	System	System	**All driving modes**

Source: SAE International Standard J3016

According to Fig. 2 (SAE, 2014), vehicle can be defined at six different levels from level 0 (no-automated) to level 5 (full-automated). Vehicles with level 3, where the driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task with the expectation that the human driver will respond appropriately to a request to intervene, or higher levels are regarded as AV. People have been dreaming of AV for a long time, but the achievement was never as close as nowadays. With the availability of accumulated research on computer vision technology, the booming in the high-speed processor (Sun et al., 2006), and the increasingly hot transport issues, the autonomous vehicle becomes the favorite topic of technology companies. Both conventional vehicles manufactures, such as Mercedes Benz and Ford, and information technology companies, such as Google and Apple, see great market potential and benefit of AV in the near future. Motivated by the approaching autonomous vehicle, relevant authorities all over the world are preparing for the revolution of transport. Studies show that AVs have the potential to (Fagnant and Kockelman, 2015a):

- reduce traffic accidents
- alleviate traffic congestion and fuel consumption
- generate new traffic demand and VMT (Vehicle Miles Traveled), especially for kids and elderly people
- release parking-occupied urban space

Considering the possible profound influence of AV on transportation, numerous studies looked at AV from the perspective of transport with both analytical and simulation approach. In spite of the importance of the microscopic simulation of AV, such as for platooning and interactive intersection, the thesis only review the most relevant area, macroscopic city-scale simulations with agent-based modeling.

Some studies begin to build up a macroscopic simulation framework of AV. Azevedo et al. (2016) proposed several extensions at the short-term and midterm levels to model and simulate autonomous vehicle system and its effects on travel behavior in Singapore using SimMobility. Carlino et al. (2012) demonstrated a simple application using AORTA through an experiment testing intersection policies for AV at a city-wide scale. Maciejewski and Nagel (2011) presented the idea and the initial outcomes of DVRP (Dynamic Vehicle Routing Problem) Optimizer in MATSim. Several studies were conducted to show the above mentioned future effects of AV on transport. Some people claimed that SAV may alleviate congestion. Maciejewski and Bischoff (2017) pointed out that large fleets operating in cities may have a positive effect on traffic if road capacity increases accordingly using agent-based model MATSim. Levin et al. (2016) concluded an opposite argument with a dynamic traffic assignment (DTA) simulator using the cell transmission model (CTM), where SAVs can cause significant congestion because of the additional trips made to reach travelers' origin. Bösch et al. (2017) also worried that the introduction of AV may have a negative influence on the accessibility of transport system. Liu et al. (2017a) suggested that the lower fare of SAV increase the number of empty vehicles, which may turn a congested city to a far more congested city. Bischoff and Maciejewski (2016) contradict Levin's opinion and show that higher congestion effects are not necessarily expected to occur despite the increase of the total travel time due to empty trips, which is compensated by more fluent traffic flow and no parking search. Besides, Fagnant and Kockelman (2014) stated that the overall emission savings of SAVs can be sizable due the quick fleet turnover. To conclude, whether AV will reduce congestion and emission is still controversial, and reducing empty kilometers traveled of AV might release traffic congestion.

2.2 Optimal fleet size and deployment

Apart from solving existing on-road problems, from a long-term point of view, AV is a possible substitute for the private vehicle, which occupied vast amounts of urban space. In the future, a single AV may replace several private vehicles, as a result, a huge number of parking areas could be reused for better urban quality. Therefore, AV is widely simulated as SAV. The replacement rate as well as the optimal fleet size to satisfy traffic demand attracts attention.

Hörl et al. (2017) simulated SAV in Zurich and found that different dispatching and rebalancing algorithms give rise to a variable result of fleet sizes, waiting times and fleet occupancy. Zhu and Kornhauser (2017) conducted a similar project in New Jersey and found that repositioning vehicles locally can decrease the fleet size significantly, maintaining a high level-of-service(5 minutes departure delay). Zhu and Kornhauser (2017) also simulated AV as ATOD but did not take dynamic ride sharing into consideration. Fagnant et al. (2015) studied SAV's replacement of conventional vehicles applying the best-performed relocation strategy from his previous research (Fagnant and Kockelman, 2014) and the study shows that each SAV can replace around eleven conventional vehicles. Brownell and Kornhauser (2014) calculated the greatest number of vehicles among 48 30-min segments as the optimal fleet size with two different rebalancing strategies. These papers more or less focus on the influence of pre-defined rebalancing strategies on the AV fleet size. Aside from simulation studies, some researchers also solved the optimization of fleet size from an analytical perspective. Spieser et al. (2016) assessed the optimal fleet size, taking the cost of vehicles, customer walk access and the expense of moving empty vehicles into account. Li et al. (2010) calculated optimal fleet size with a cost effectiveness analysis.

3 Simulation framework

3.1 Simulation platform

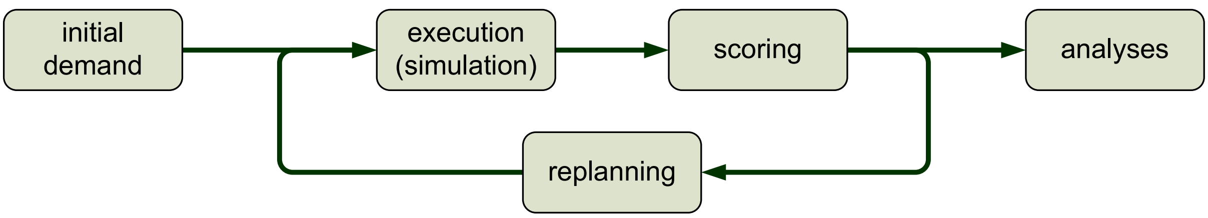
The optimal fleet size and deployment problem with dynamic ride-sharing and individual travel behavior require complicated real-time matching (dispatching algorithm) of AV and passengers, which is not easy to solve with an analytical approach and conventional transport models. Therefore, the simulation is based on existing DRT (Demand Responsive Transit) contrib of the Multi-agent Travel simulator (MATSim). Contribs is the name for extensions in MATSim.

MATSim is chosen as the simulation platform for the project because:

- MATSim is agent-based which can simulate real-time dynamic matching problem of millions of agents.
- NP-hard optimization problem under specific constraints can be approached with the co-evolutionary algorithm of MATSim.
- MATSim is applicable for large-scale macroscopic simulations with a massive amount of individual travelers.
- MATSim is expendable and modularized, where modification and new implementation can be easily adapted to the existing system.
- Several extensions to simulate dynamic ride-sharing and ATOD are available in MATSim.

3.1.1 Introduction to MATSim

Figure 3: MATSim loop



Source: Horni et al. (2016)

MATSim is an activity-based, extendable, open-source multi-agent simulation framework implemented in Java. (Horni et al., 2016) A typical MATSim loop consists of initial demand, simulation, scoring, replanning and analyses and the loop will be iterated during simulation. During the iterations, with the co-evolutionary algorithm, each agent will optimize their score individually and reach an equilibrium. (Horni et al., 2016) It is noted that normally MATSim simulates the traffic behavior of a single day.

Figure 4: Typical plan in MATSim

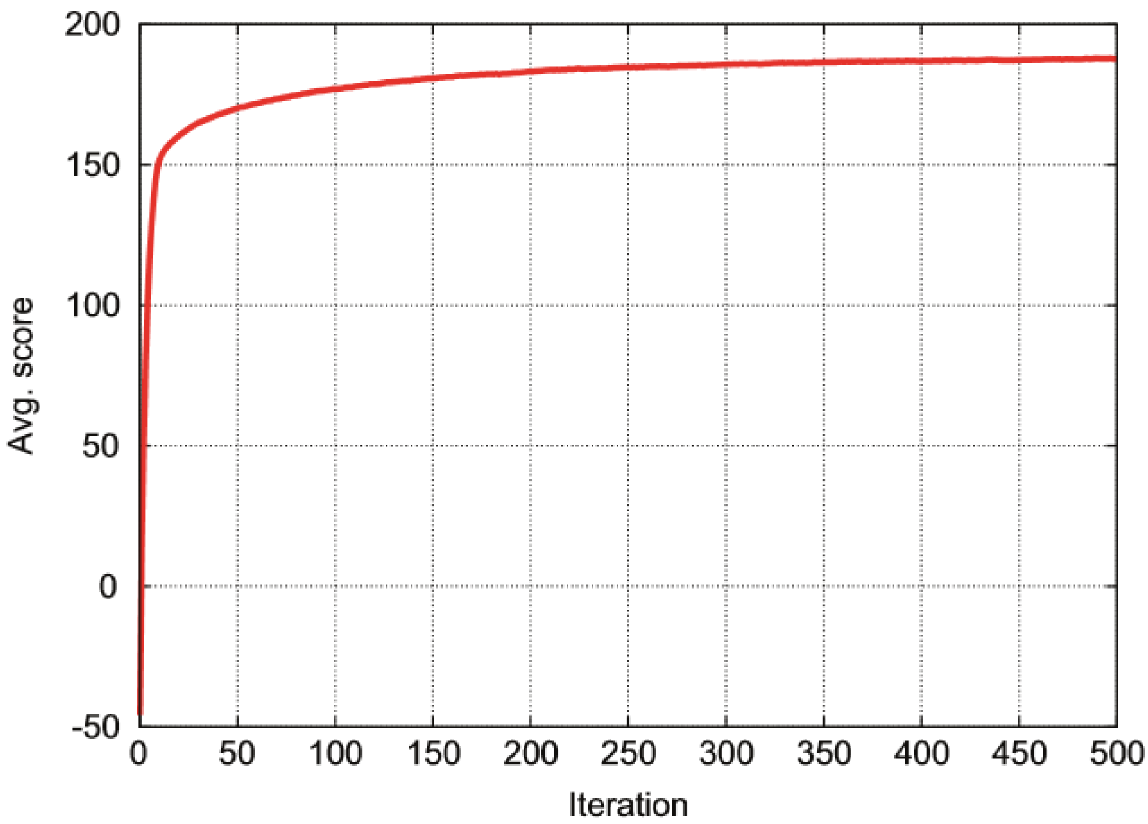
```
<person id="10012_2">
  <attributes>
    <attribute name="age" class="java.lang.Integer" >26</attribute>
    <attribute name="carAvail" class="java.lang.String"
      >always</attribute>
    <attribute name="employed" class="java.lang.Boolean" >true</attribute>
    <attribute name="sex" class="java.lang.String" >f</attribute>
  </attributes>
  <plan score="19.704615430161144" selected="yes">
    <activity type="home" link="109283506_1" facility="5821_18"
      x="686463.2969000004" y="4824239.2903" end_time="08:21:24" >
    </activity>
    <leg mode="drt" trav_time="00:05:48">
    </leg>
    <activity type="work" link="238014193_0" facility="7540_16"
      x="684918.0522999996" y="4823849.636299999" start_time="07:14:19"
      end_time="17:29:58" >
    </activity>
    <leg mode="drt creation" trav_time="00:06:33">
    </leg>
    <activity type="home" link="109283506_1" facility="5821_18"
      x="686463.2969000004" y="4824239.2903" >
    </activity>
  </plan>
</person>
```

Source: Sious Falls scenario

Initial demand Distinct from traditional assignment platform, such as PTV Vissum, initial demand in MATSim is not a zone-based OD matrix, but agent-based activity chains. The activity chains are usually derived from empirical data through sampling or discrete choice modeling.(Horni et al., 2016). As depicted in Fig. 4, activity chain consists two parts, leg and activity. Activity describes the purpose of the trip and leg includes transport mode, travel time, departure time, etc.. Besides, the score is recorded after each iteration to estimate the performance of each activity chain. In addition, a typical plan in MATSim also contains the socio-demographic information of the agent, such as age, car availability, employment and sex.

Execution Mobsim (Mobility simulation) is a core part of MATSim, although different traffic model can be installed for different simulation purpose. Default QSim, which applies for queue-based traffic flow model, is enough for the large-scale macroscopic simulation. At the beginning of the simulation, each agent will choose a plan from their memory depending on the executed score of plans from previous iterations. The selected plan will be executed and evaluated for the current iteration with QSim.

Figure 5: Typical converged score curve



Source: Sioux Falls scenario

Scoring and replanning After simulation, the performance of each agent will be evaluated during scoring and the plan will be modified accordingly during replanning. The normal scoring function is the sum of the linear positive utility function of activity, negative utility function of leg and customized event-based scoring. Several replanning strategies with different weights can be defined in the configuration to modify respectively routing, transport mode, departure time, destination etc.. For a simulation, normally two or more strategies will be defined and the weights indicate the proportion of plans following different strategies. Each agent will store a specific number of plans and the worst-performed plan will be obsolete once the memory is full. Innovation in MATSim means whether to choose new plans from memory. Before turning off innovation, agents can try any plan under the condition of the replanning strategies; while after turning off innovation, agents must select plans from their memory and update the score of existing plans. After an adequate number of iterations, the scoring function should have converged and the system will reach its equilibrium. Fig. 5 shows a typical converged score curve. This iterative scoring and replanning process is important components of the co-evolutionary algorithm in MATSim.

3.1.2 Introduction to DRT contrib

Several extensions to simulate AV are available in MATSim, DRT (Demand Responsive Transit), DVRP (Dynamic Vehicle Routing Problem) and taxi contrib. DVRP is a simulation framework which can be described as the Dynamic Multi-Depot Vehicle Routing Problem with Time Windows and Time-Dependent Travel Times and Costs (Maciejewski and Nagel, 2011). Compared to the normal population agent, the dynamic agent in DVRP can modify its dynamic activity chain during the day. On the basis of DVRP, taxi extension and DRT extension (also called taxibus extension) were developed. DRT contrib is an extension of MATSim which allows to serve several passengers on board at the same time (Bischoff et al., 2016). Compared to other existing contribs in MATSim, the simulation of ATOD is close to the simulation of taxibus in the DRT contrib, where minibuses with dedicated stops, centralized dispatching algorithm and dynamic ride-sharing have already been successfully implemented. The DRT in MATSim is a transport simulation system, which can switch between stop-to-stop service and door-to-door service.

Vehicle DRT contrib needs to import vehicle file into the simulation. Although in ATOD simulation, initial fleet size is zero and vehicles are generated during simulation, the attribute of generated vehicles are still the same, which contains vehicle ID, capacity, service begin and end time and start link.

Transit schedule In DRT contrib, minibuses travel dynamically among existing transit stops, therefore transit stops file should be input in the simulation framework. Different from conventional PT transit schedule, only stop facilities is needed as an input of simulation. The origin and destination of DRT routing will be constrained to the location of the stop facilities.

Figure 6: Input transit schedule file in DRT contrib

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE transitSchedule SYSTEM
  "http://www.matsim.org/files/dtd/transitSchedule_v1.dtd">

-<transitSchedule>
-<transitStops>
<stopFacility linkRefId="127" y="-600" x="0" id="1"/>
<stopFacility linkRefId="152" y="-200" x="800" id="2"/>
<stopFacility linkRefId="217" y="-600" x="200" id="3"/>
<stopFacility linkRefId="240" y="-400" x="400" id="4"/>
</transitStops>
</transitSchedule>
```

Source: DRT contrib

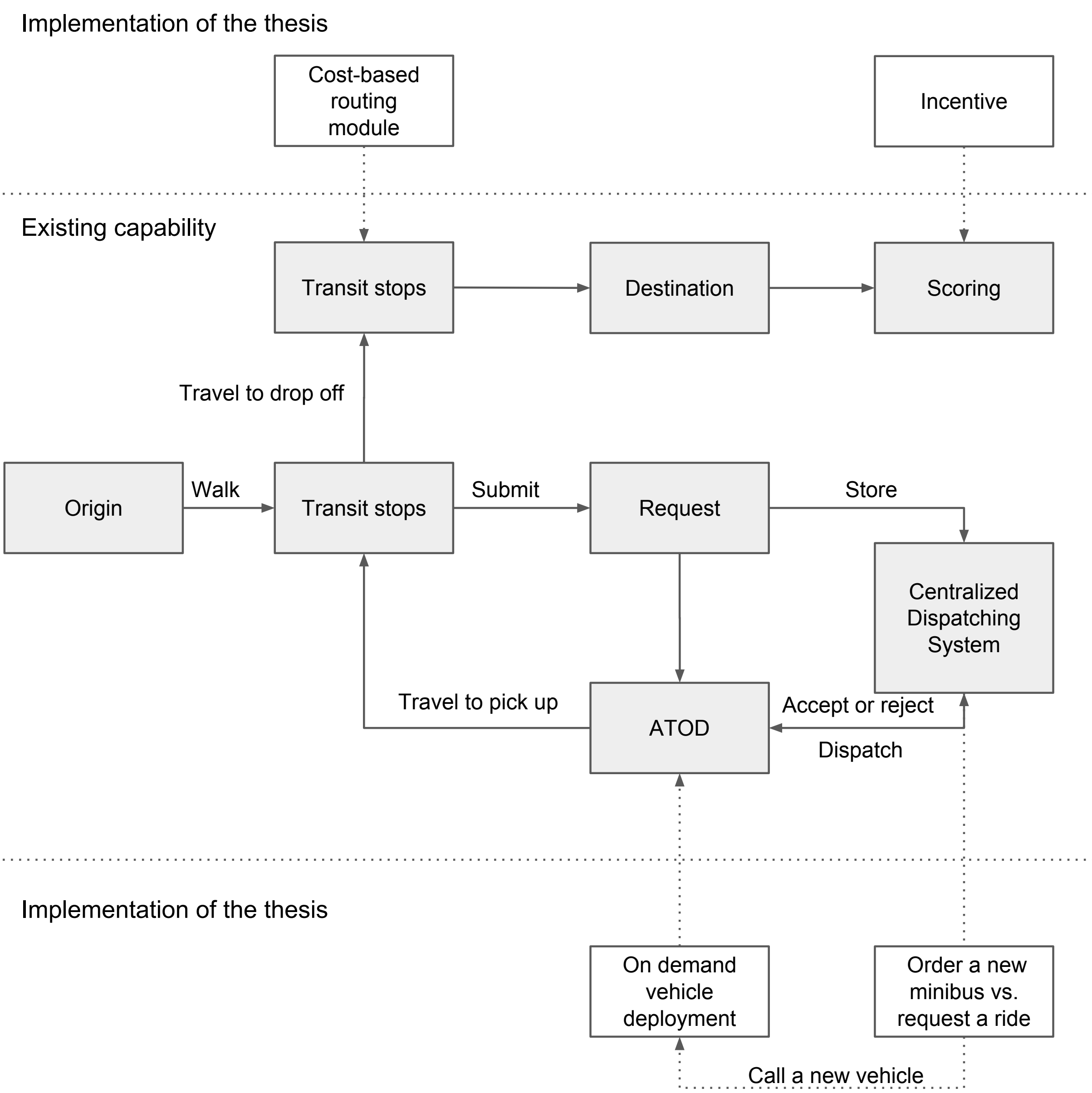
3.2 Simulation framework

As shown in Fig. 7, in the DRT module, a typical user goes first to the closest transit stops, submits a travel request upon arrival and waits for a minibus. Among all the available minibuses, the request will be dispatched to the one with least time loss, taking both pick-up time loss and drop-off time loss into account. Giving priority to the accepted passengers, the system will reject the request once specific accepting constraints are not satisfied. Then the passenger will compare the travel cost of all accepted vehicles and submit the request to the minibus with least cost. If no vehicle accepts the request, the request will be labeled as "rejected" in the system. Later, ATOD will travel to pick up the passenger and drop off the passenger at the closest transit stops to the destination. After the simulation of the whole day, the above-mentioned process will be evaluated during scoring.

In most simulations, the initial fleet size and location is pre-defined, but some researchers use simulation to find an optimal fleet size. Fagnant et al. (2015) run a *seed simulation* to define an appropriate fleet size, which generates vehicles once the passenger waits for more than 10 minutes. This approach can guarantee the passengers' maximum waiting time, however, as there is no penalty for extra vehicles with low-occupancy in the model, the fleet size is overestimated. Therefore, as shown in Fig. 7, the following functionalities are incorporated into the DRT module:

- On-demand vehicles deployment
- Mode choice: Order a new minibus vs. Request a ride
- Incentive
- Selection of origin and destination stops

Figure 7: Workflow of ATOD simulation



The main objective of these functionalities is to create a trade-off between calling a new minibus, and requesting a ride in one of the already deployed vehicles. If an agent calls a new minibus, it will not have to wait long at the transit stop, it will not be aborted by the simulation, but that vehicle must be popular for other agents or indispensable. If an agent tries to travel in one of the moving minibuses, it will have to wait, it can be aborted if its request is rejected many times, but it will not be penalized for calling a new vehicle. These agents can also decide to walk longer at the beginning or the end of their trips to find an appropriate vehicle.

Apart from above mentioned newly implemented functionalities, following mentioned existing modules in DRT contrib will also be modified to fit the purpose of the simulation.

- Walking scoring function
- ATOD dwelling time
- Request accepting constraints
- Annealing

Driven by the evolutionary algorithm, the simulation will try to reach an equilibrium with least aborted trips, least number of vehicles and most ride sharing. The fleet size and deployment in the equilibrium will be considered the optimal solution.

4 Implementations

4.1 On-demand vehicles deployment

On-demand vehicle deployment consists of the on-demand vehicle generator and vehicle removal algorithm. The generator will create a new vehicle for the passenger who calls a new vehicle at the passenger's current location. If a vehicle is idle for more than I minutes (I is defined as the vehicle idle time in the configuration), the vehicle will be marked as rarely-used and removed from the simulation.

The on-demand vehicles deployment can be regarded as a re-balancing strategy under the assumption that the empty traveling time from or to depots is neglected. Since no initial minibus is existing in the system, the number and the location of minibuses is highly dependent on demand. The maximum number of vehicles in the system can represent the optimal fleet size on demand.

4.2 Mode choice: Order a new minibus vs. Request a ride

Agents in MATSim can evolve to equilibrium through changing mode, routing, and time allocation. In order to find the optimal fleet size in the system, agents are allowed to choose between two modes, *new DRT call* and *request DRT*.

For agents with the *new DRT call* mode, minibuses are created whenever they arrive at the stations. Agents with the *request DRT* mode are not allowed to create any vehicles and they have to wait till the request is accepted by the existing minibuses. Every U minutes (U is defined as the request update time in the configuration), the rejected request will be submitted again. To improve the computational performance, once a *DRT requester* waits for more than W minutes (W is also defined in configuration as the maximum passenger waiting time), the agent will be labeled as "stuck and abort" in the system and it will get a significant penalty for not finishing its daily plan. The negative value is extremely huge, which is the most negative marginal utility during the trip multiplied by 24 hours. In other words, the aborted trip is the same as the trip where agent spend a whole day (24 hours) to travel by the most unpleasant mode or wait.

In other words, agents who cannot find a deployed minibus are encouraged to call new vehicles; while a *new DRT caller* whose vehicle is not frequently-used will be penalized. The ratio of *new*

DRT callers and *DRT requesters* emerges from the evolutionary algorithm of MATSim, and the number and location of new DRT caller represent the optimal fleet size and the initial location of the minibuses.

4.3 Incentive

In order to find the optimal fleet size, it is essential to limit the number of *new DRT callers*. In MATSim, all plans will be rerouted based on the score of previous iterations. Therefore, ride-sharing can be encouraged through a bonus or penalty. In the simulation, only popular vehicle incentive is implemented.

Popular vehicle incentive I_t is calculated directly by the total number of passengers traveling with the vehicle. The utility function of new DRT call mode with popular vehicle incentive is defined as:

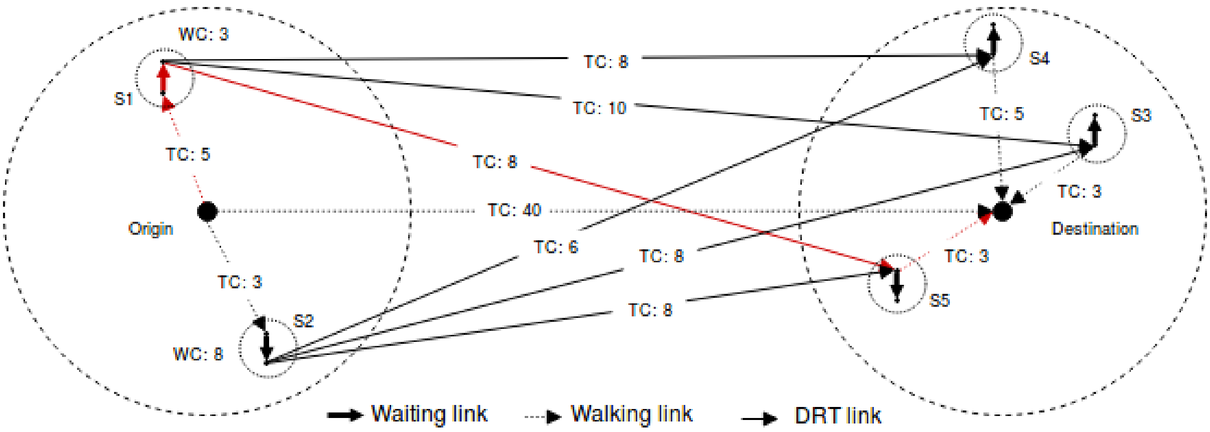
$$U_t = \underbrace{\beta_r * P + C_r}_{incentive} + \beta_{DRT} * tt \tag{1}$$

The constant C_r of each new DRT calling trip and marginal disutility of DRTβ_{DRT} is defined in the configuration, β_r is 1 util/passenger for the simulation, P is the number of passengers the vehicle serves. In other words, the incentive will be awarded if the minibus the caller creates serve more than C_r passengers. The more popular the vehicle is, the higher score for the caller of the vehicle can get.

4.4 Cost-based Routing Module

For traditional public transit, passengers always compare different transit stops and choose the most satisfying one in terms of traveling time. In the DRT module, passenger always walks to the closest stop in terms of Euclidean distance to call a minibuses. In reality, for conventional taxi service, people are eager to walk a little bit more to call a taxi in the main street or to call a taxi on the other side of the road to avoid the detour. To simulate these behaviors, a new routing module is implemented in the system under the assumption that agents prefer stops with less waiting time and less detouring based on existing MATSim contrib, event-based PT router. It took the given schedule as a start point for the initial iteration, but information on travel times, occupancy of the public transport vehicles, and waiting time was updated for subsequent iterations.(Ordóñez Medina, 2016) As mentioned in the Section 1, DRT can be simulated as a public transport with dynamic routing, therefore similar routing module, a cost-based routing module, is implemented in the system with some modifications.

Figure 8: Cost-based Routing Module(WC: Waiting Cost, TC: Traveling Cost, Red Line: Least Cost Routing)



The waiting time of each stop is zero at first iteration, and the waiting time of each stop will be updated each iteration with the actual value of previous iteration. Similar to the transit router, the new routing module will compare the total travel time, which is the sum of traveling time of the access walk, waiting time and traveling time of minibuses of all stops within the 1000m of origin or destination and choose the one with least total travel time. The waiting time is the average waiting time of the passengers who depart at +/-15 minutes from the stop in the last iteration. For example, if one DRT requesting passenger is going to depart at 15:00 at iteration 3, the passenger will take for all stops within the 1000m radius of origin as possible origin stops and all stops within the 1000m radius of destination as possible destination stops. For all possible origin and destination stops, the passenger will calculate and compare the sum of cost of walking to any possible origin stops, average waiting time of origin stops from 14:45 - 15:15 at iteration 2, traveling from origin stops to destination stops, walking from possible destination stops to the destination. The agent will choose the origin and destination stop with least cost and compare the least DRT traveling cost with direct walking cost and decide the travel route with the corresponding routing module. It is worth noting that new DRT caller is routed with the similar routing module but excluding waiting time in cost calculation due to the assumption that new DRT caller will be immediately served. As shown in the Figure 8, each stop is depicted as two nodes with a connecting link in the model. Both nodes share the similar ID, but one of them ends with "_W". Cost of traveling of access walk will be saved in the link between origins and origin stops or between destinations and destination stops, cost of traveling of DRT will be saved in the link between origin stops and destinations, cost of waiting will be saved in the connecting link between two stop nodes, which is a directed link. Link of access walk can only depart or arrive at the stop node without "_W" and link of DRT can only start from the stop node with "_W" and end at the stop node without "_W". Implementing this algorithm as well as multi-node Dijkstra algorithm, the agents can find the least total cost route. In the Figure 8 among the six possible DRT routing and one direct walk routing, despite the fact that the travel cost from origin to Stop 1(S1) is higher than to S2 as well as the travel cost from S2 to S4 is the least, the agent will choose the red path, origin - S1 - S5 - destination, because of the compensation of lower estimated waiting cost.

It is expected that agents will always try the stops with least travel cost of access walk and in-vehicle, given that initial waiting time of all stops is zero and only travel time counts. However, with the accumulated experience, agents may try further stops with less average waiting time. For example, if agents have perfect experience with closest stops, it will keep its choice next iteration, otherwise it will keep trying a further stops with probably shorter waiting time till the best combination of travel cost of access walk, waiting cost and travel cost of DRT is reached. Similar to public transport hubs in reality, some stops may be more attractive than others and will offer more frequent service. As a result, it is supposed that some stops with high demand will become big transport hub and may attract more minibuses as well as passengers. Passengers who live in rural and low demand area may have to walk more to gather in transport hub while passengers who live in high demand area may walk less. Besides, the short-distance traveler may prefer to walk considering the unpredictable and unreliable time. Mode share of walking may increase with iterations.

4.5 Walking scoring function

In the simulation, walking is the only substitute of the two DRT modes. Walking is highly recommended for short-distance trips but not for long-distance trips. In MATSim, linear walking as well as in-vehicle traveling and waiting scoring function without any constraints is implemented by default. However, different from in-vehicle traveling and waiting, walking has its limitation. It is widely acknowledged that a half-mile (approx. 800m) is an appropriate and comfortable upper limit for walking. Therefore, in the simulation, when calculating the direct walking score in cost-based routing module, linear walking utility function with 800-meter limitation is implemented, which can be interpreted that agents will choose DRT mode without hesitation once the direct walk distance is more than 800m. It is noted that during scoring, since access walk and walk will be scored together, no limitation should be applied because sometimes the nearest stop for passengers may be further than 800m. Under this condition, access walk for more than 800m to request or call a minibus is the only option for the agent.

4.6 ATOD dwell time

Table 1: A Summary of results from international works

Study	Stop design	Avg. deceleration/ acceleration time	Boarding time per passenger
Xu, Kwami, & Yang, 2010	Bus bay	9.0s/10.7s	2.1s(single-channel door) 1.7s(double-channel doors)
	Curb-side stop	8.5s/10.9s	-
Chen, Zhou, Zhou & Mao, 2013	Bus bay	11.11s/11.12s	2.22s(load factor < 0.7) 2.37s(load factor >= 0.7
	Curb-side stop	9.74s/10.2s	1.82s(load factor < 0.55) 2.49s(load factor >= 0.55)
Transport for London, 2006	Bus bay	-	2.8-3.8s
	Curb-side stop	-	0.5-1s faster than bus bay
Genivar, 2011	Bus bay	-	3.5s
	Curb-side stop	-	
Wang, Ye, Wang Xu & Wang, 2016	Bus bay	-	2.5–4.0s(single-channel door)
	Curb-side stop	-	0.6–2.5s (multiple- channel doors)

Source: Liu et al. (2017b)

ATOD dwell time is similar to bus dwell time. In the DRT contrib, dwell time is fixed, which is the 60s for all stops. However, in reality, bus dwell time is influenced by several factors, including the type of vehicles, the design of stop bay, the number of doors, number of boarding passengers etc. The first three factors are neglected in the simulation. Thus, in order to accurately describe dwell time and at meanwhile simplify the issue, it is assumed in the simulation that bus dwell time is only a linear function of the number of boarding passengers, therefore the dwell time can be described with the following equation:

$$T_{dwell} = \theta_p * P_{boarding} + D \qquad (2)$$

where T_{dwell} is noted as ATOD dwell time, θ_p is boarding time for each passenger and D is a constant which describes fixed dwell time, such as deceleration and acceleration time of minibus. According to Liu et al. (2017b) summary of international works, both deceleration and acceleration time are around 10s and boarding time per passenger is around 2s. It is assumed that acceleration and deceleration will take twice as much time as driving with free flow speed, which means the time loss due to processing such as acceleration and deceleration will be $(10 + 10) * 0.5 = 10$. Therefore, the θ_p in the model is 2 and D is 10. It is noted that given the above equation of ATOD dwell time, the waiting time of new DRT call can be longer than DRT request. For example, new DRT caller needs to wait for at least 10s (acceleration and deceleration) + 2s (boarding time for one passenger) + 1s (reacting time) = 13s for a minibuses; while if DRT requester happens to arrive at the stop when the vehicle is going to depart, the requester will only wait for 2s (boarding time for one passenger) + 1s (reacting time) = 3s.

4.7 Request accepting constraints

Request accepting constraints is of great significance for the ride-sharing problem. A simple but important request accepting constraint can be the rejection of a new request from a full vehicle. Another crucial constraint for the ride-sharing problem can be detour accepting constraints. As it is rare to find some ideal passengers whose origin and destination happen to be on the way of other passengers, detouring to some extent is necessary. In other words, higher occupancy requires the sacrifice of time by passengers. Detouring can happen during both pick-up and drop-off. For example, minibuses may detour a few minutes to pick up more passengers. Then the in-vehicle passengers' travel time will be sacrificed and drop-off time will be postponed. At the meanwhile, if the minibuses accept more requests on the way to pick up passengers. Then the waiting time for these passengers may also be influenced by the further request and pick up time will be postponed. Once the request is accepted by the minibus, the agent has no chance to submit to another minibus; while for the new-approaching request, it is possible to be accepted by other vehicles once rejected. Given that accepted requests cannot be canceled, the experience of accepted passengers have higher priority and need to be guaranteed. The request accepting constraints is set for a better experience of accepted passengers. Apart from the constraints of detour loss, minibuses too far away will be excluded when new requests are coming even though a minibus is idle to reduce long-distance empty traveling.

4.7.1 Detour loss

As detouring is the key point of the request accepting constraint, it needs to be quantified for comparison. Detour loss is a number to measure detouring, which consists of pick-up detour loss, drop-off detour loss and stop duration. Each request can be divided into two tasks, pick-up task and drop-off task. Theoretically, pick-up task and drop-off task can be inserted into any position of any vehicle, but different insertions or vehicles leads to different detour losses.

In the simulation, the detour loss of every insertion of every vehicle will be looped, calculated and compared. The detour loss of rejected insertions is $Double.MAX$ and the best insertion will be the one with least detour loss. If the least detour loss is $Double.MAX$, it means the request is rejected by all insertions of all vehicles.

Figure 9: Illustration of the calculation of detour loss

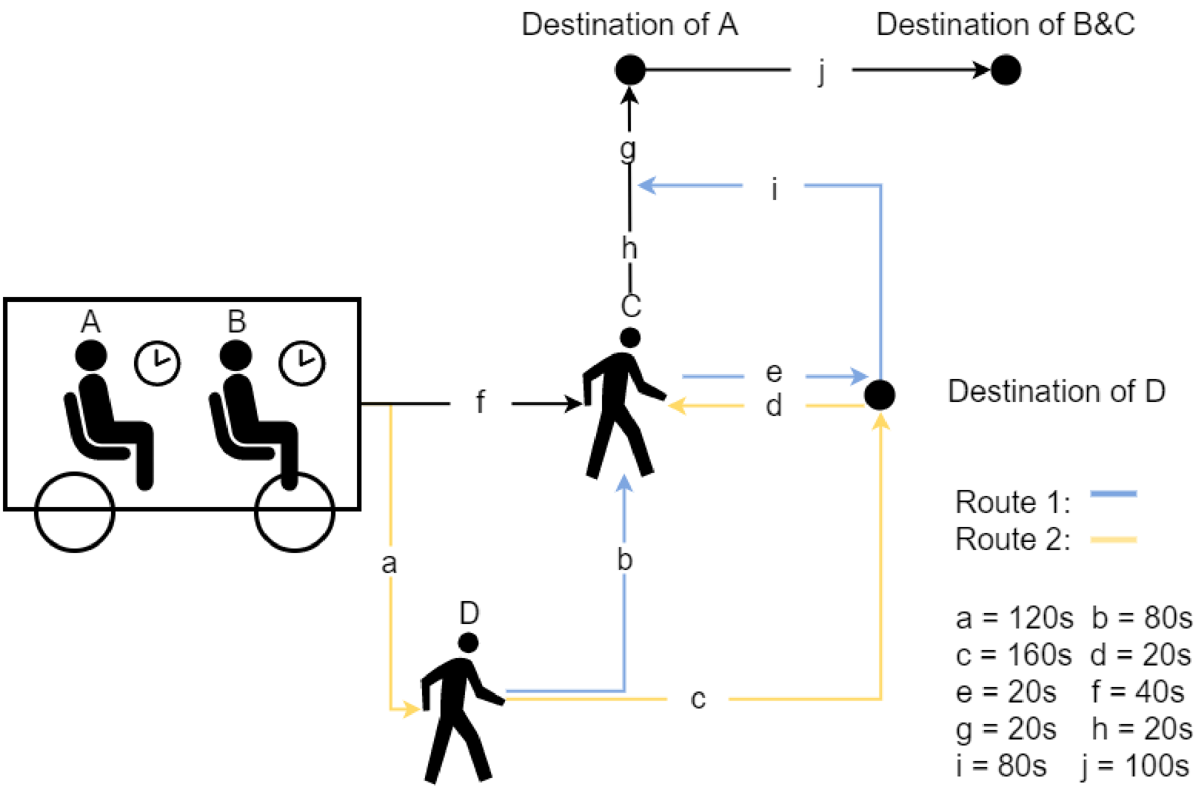


Fig. 9 illustrates an example of the calculation of detour loss. Passenger A&B are in-vehicle passengers and passenger C is waiting for a minibus and the request is already accepted. Passenger D just submitted the request and the vehicle is evaluating the influence of the request. The original task order is pick-up C - drop-off A - drop-off B&C, when D request comes, pick-up D and drop-off D can be inserted into arbitrary position among insertions current status - pick-up C, pick-up C - drop-off A, drop-off A - drop-off B&C and after drop-off B&C, as long as drop-off D is after pick-up D. Thus, there will be ten combinations of insertions of pick-up and drop-off tasks. Among the potential six different routes, only accepted routes are marked in the illustration (The specific accepting constraints will be discussed in next section). Route 1 is current status - *pick-up D* - pick-up C - *drop-off D* - drop-off A - drop-off B&C; Route 2 is current status -

pick-up D - *drop-off D* - pick-up C - drop-off A - drop-off B&C. Given stop duration for one passenger is s=12, pick-up detour loss of route 1 is a + b - f + s = 172s, drop-off detour loss of route 1 is e + i - h + s = 92s, total detour loss of route 1 is 264s; Pick-up detour loss of route 2 is a + c - f + s = 252s, drop-off detour loss of route 2 is d + s = 32s and total detour loss of route 2 is 284s. Route 1 will be selected. Note that travel time is not estimated based on real-time travel situation, but with free-flow travel situation.

4.7.2 Pick-up detouring accepting constraints

Max travel time constraints and calculation of pick-up tolerance Pick-up detouring tolerance is how long the accepted passenger can wait for the coming vehicle. Pick-up detouring tolerance consists of max traveling time and waiting tolerance. The former parameter also decides the maximum accepted estimated traveling time from the current location of the vehicles to the location of the passengers.

Max travel time constraints:
Estimated travel time to pick up <= Max travel time

Figure 10: Illustration of pick-up tolerance

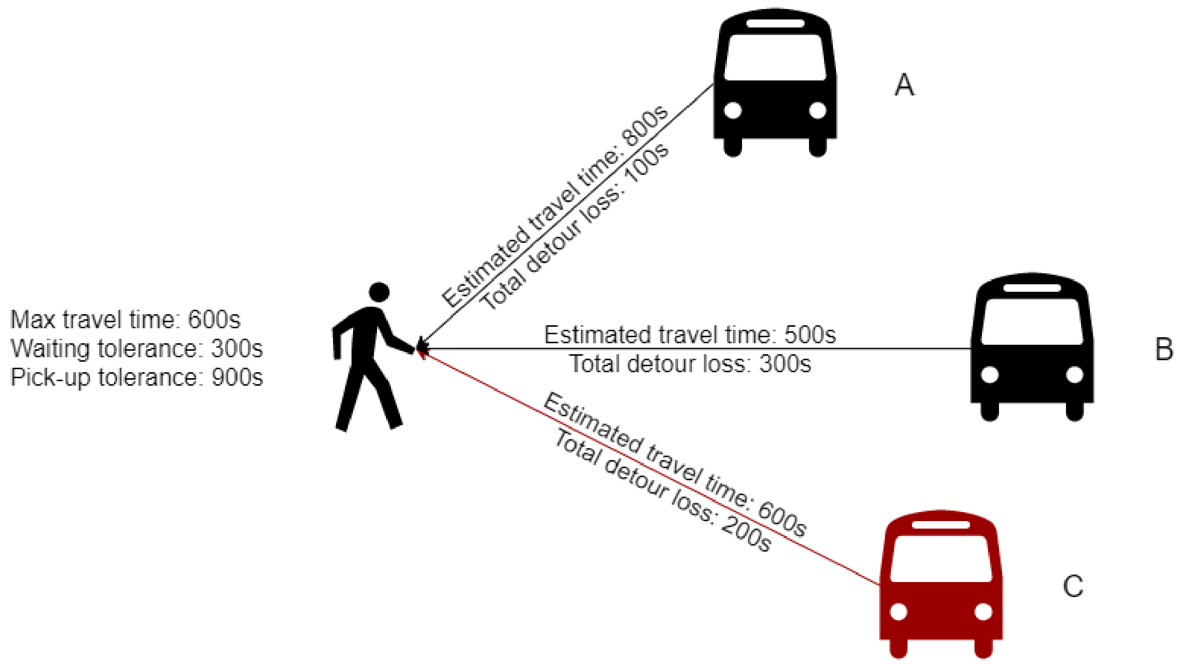


Fig. 10 exemplifies how the max traveling time and waiting tolerance function in the model.

Given that max traveling time is 600s, minibus A is excluded even though the real total detour loss of A is the minimum due to lower exceeding estimated travel time. B and C are all available in terms of max traveling time and C will be chosen for less total detour loss. The pick-up tolerance of passenger is max traveling time plus additional waiting tolerance, which means the time of detouring allowed is also related to estimated travel time. In the graph, since the maximum travel time is already reached, the passenger can only allow 300s delay for pick-up. If estimated travel time is less than max travel time, the tolerance will be the subtraction of estimated travel time from max travel time plus max travel time. In other words, the maximum waiting time for the accepted request, also known as pick-up tolerance, is the sum of max travel time and waiting tolerance. The latest departure time, which is the sum of current time and maximum waiting time, will be saved as latest departure time once the request is accepted.

Pick-up detouring accepting constraints:
For all pick-up tasks between new pick-up task and new drop-off task:
The estimated end time of the pick-up task + stop duration + pick-up detour loss <= The latest departure time of the task
For all pick-up tasks after both new pick-up task and new drop-off task:
The estimated end time of the pick-up task + stop duration + total detour loss <= The latest departure time of the task

Pick-up detouring tolerance constraints Once a new request coming, all tasks after the insertion of the new request will be influenced. All pick-up tasks after the insertion apply for pick-detouring tolerance constraints. As mentioned above, every accepted request has an attribute called *latest departure time*. All latest departure time of pick-up tasks after the new insertion will be compared to the departure time with new requests . The constraints will guarantee all waiting passengers can depart before latest departure time, otherwise the new request will be rejected with no exception.

As depicted in the Fig. 9, pick-up C is the only affected pick-up task. The influence of route 1 is 172s (pick-up detour loss) and of route 2 is 284s (total detour loss). Assuming current time is 0s and the latest departure time of pick-up C is 300s, then the estimated end time of the pick-up C is 40s and the stop duration for one passenger is 12s. For route 1, 40s + 172s = 212s < 300s, the insertion will be accepted by pick-up constraints; while for route 2, 40s + 284s = 336s > 300s, the insertion will be rejected.

4.7.3 Drop-off detouring accepting constraints

Similar to pick-up detouring accepting constraints, drop-off detouring accepting constraints note the tolerance of increasing travel time for in-vehicle passengers. Once a new request is accepted, all drop-off tasks after the insertion will be postponed. Every accepted requests have an attribute called *latest arrival time*. Latest arrival time is calculated by the function:

$$T_{latestArrival} = \underbrace{\alpha_{DetourTolerance} * T_{eatimatedTravel} + \beta_{DetourTolerance}}_{\textbf{Max ride-sharing travel time}} + T_{departureTime} \tag{3}$$

where $T_{latestArrival}$ is the latest arrival time, $T_{eatimatedTravel}$ is the estimated travel time from pick-up stop to drop-off stop and $T_{departureTime}$ is the time when the request is scheduled. Distinct from pick-up tolerance, drop-off tolerance is dependent on estimated travel time, which means a long-distance passenger can tolerate more detouring.

All latest arrival time of pick-up tasks after the new insertion will be compared to the arrival time with new requests. The constraints will guarantee all in-vehicle passengers can arrive at the destination before latest arrival time, otherwise the new request will be rejected with no exception.

Drop-off detouring accepting constraints:
For all drop-off tasks between new pick-up task and new drop-off task:
The estimated begin time of the drop-off task + stop duration + pick-up detour loss <= The latest arrival time of the task
For all drop-off tasks after both new pick-up task and new drop-off task:
The estimated end time of the task + stop duration + total detour loss <= The latest arrival time of the task

Table 2: Influence of request D of route 1

Passenger	Arrival time once request D is accepted	Latest arrival time
A	$a + s + b + s + e + s + i + g = 356s$	$(f + s + h + g) * \alpha + \beta = 338s$
B	$a + s + b + 12 + e + s + i + g + s + j = 468s$	$(f + s + h + g + s + j) * \alpha + \beta = 506s$
C	$e + s + i + g + s + j = 244s$	$(h + g + s + j) * \alpha + \beta = 578s$

As depicted in route 1 of Fig. 9, the drop-off of passenger A, B and C are all influenced by the new request D. Given current time is 0s, travel tolerance beta(β) is 200, and travel tolerance alpha(α) is 1.5, the influence of request D on drop of can be calculated as shown in Table 2. Although the drop-off constraints of passenger B and C is satisfied, the request is still rejected due to violating the constraints of passenger A.

5 Scenario

The simulation is executed on the scenario of the Sioux Falls(SD, USA), the population was scaled down to 10% of total population(8483 agents) for computational reasons, and the road capacity was scaled down accordingly. All agents executed exactly two legs during a day, home to work or secondary and work or secondary to home. Sioux Falls is a small city with 1810 nodes, 3359 links and 150 transit stops in the transport network. It is chosen to test the new ATOD implementation because it is simple enough for computation but also complete and realistic enough for reasonable results. Due to the purpose of the simulation is not to estimate or simulate the real situation of Sioux Falls but to understand and solve the fleet size and deployment problem, the transport mode of all plans are converted to *new DRT call* and *DRT request* mode in the input population file. It is noted that cost-based routing module will compare the cost of *new DRT call* or *DRT request* with the direct walk and the agents may turn to walk even though their original plan is *new DRT call* or *DRT request*. In addition, 10% of trips are randomly chosen as *new DRT call* and 90% are randomly chosen as *DRT request*. The initial proportion of *new DRT call* and *DRT request* are based on the assumption that one autonomous vehicle can replace around 10 conventional private vehicles.(Fagnant and Kockelman, 2014)

Table 3: Fixed parameters for simulation

Parameter	Value
Initial fleet size	0
Vehicle capacity	8 seats
Vehicle idle time	1800 seconds
Maximum passenger waiting time	3600 seconds
Marginal utility of waiting	-6 util/hour
Constant of DRT request	-1 util
Marginal utility of DRT request	-4 util/hour
Marginal utility of new DRT call	-4 util/hour
Marginal utility of walk	-5.8 util/hour
Detour tolerance alpha	1.5
Detour tolerance beta	600s
Max travel time	600s
Max waiting tolerance	300s

In the simulation, the values of some parameters can refer to previous research or can be fixed in the simulation, while some essential parameters may not have any reference values and need to be calibrated for robustness of the model and for a reasonable result. All fixed parameters are listed in the following Table 3. Initial zero fleet size and 8-seat vehicle are fixed for the purpose and assumption of the simulation. For the reason of improving computation time, the vehicle will disappear after being idle for more than 1800 seconds(half an hour), and the passenger will be labeled as "stuck and abort" after waiting for more than 3600 seconds (1 hour). It means that the vehicle will go back to the depot after being idle for 30 minutes and the experience of passengers who waits for more than 1 hour will be extremely terrible.

The marginal utility of waiting and walking is defined according to Wardman et al. (2016), where the marginal utility of walking is 1.45 times more than the marginal utility of in-vehicle traveling and the marginal utility of waiting is 1.5 times more. The constant of new DRT call is variable, while the constant of DRT request is fixed, -1. These scoring parameters are crucial for a trade-off of new DRT call, DRT request, and walk modes. Agents may obtain the same score of walking 10 min as waiting 9min and traveling with DRT 1min. Despite the relatively large negative constant, a new DRT caller whose vehicle serves more than the constant can get a higher score than DRT ride-sharing passengers with least waiting time, even a new DRT caller whose vehicle serves only one passenger can get a higher score than aborted DRT ride-sharing passengers. Calculating the score of each trade-off option, agents are forced to make a decision for an equilibrium where least and appropriate deployed vehicles can satisfy all the demand in the city. Loose detour tolerance can, on one hand, increase vehicle occupancy by accepting more passengers on the way, but on another hand, destroy the experience of passengers by long time waiting and detouring. Pick-up tolerance, including max travel time and max waiting tolerance, as well as drop-off tolerance, including detour tolerance alpha and beta, can affect the results collectively with different combinations. These parameters will be set fixed in the project and will be tested in future work.

Table 4: Variable parameters for simulation

Parameter	Value 1	Value 2	Value 3	Value 4	Value 5	Value 6	Value 7
Request update time	30s	300s	400s	500s	600s	700s	800s
Annealing	On	Off	-	-	-	-	
New DRT call constant	-10 util	-20 util	-30 util	-40 util	-50 util	-60 util	-

Apart from fixed parameters, some crucial parameters for ATOD will be tested, compared and analyzed from the perspective of the score, mode share, computational time, fleet size, the popularity of vehicles, ridership, fleet deployment and experience of passengers. The purpose is to compare the influence of these variables on the simulation and to figure out the better value in terms of fewer minibuses, higher occupancy, and better user experience. Since the following variables depict the model from totally different perspective, these variables can be seen as independent and identically distributed random variables. The influence of each variable can be explored individually.

Besides, the simulation will run 100 iterations with 10% agents change single trip mode each iteration for all scenarios. For first 60 iterations agents choose their plan randomly, and for the last 40 iterations, some agents begin to keep choosing the plan with the highest score in their memory. Simulation with annealing and without annealing will be compared in two different scenarios. Annealing means to disable random plan choosing gradually. In the simulation without annealing, above mentioned 10% agents will stop choosing plan randomly immediately at iteration 61; while in the simulation with annealing, 2.5% agents will begin to keeping choosing the best plan in their memory at iteration 61, 5% at iteration 71, 7.5% at iteration 81, and 10% at iteration 91. As the simulation relies on the balance between new DRT call and DRT request, sudden change of mode share may result in unstable simulation result. Therefore, a better and robust result is expected for the simulation with annealing.

5.1 Base scenario

Table 5: Configuration of base scenario

Parameter	Value
Request update time	600s
Annealing	With annealing
DRT constant	-30 util

Several groups of scenarios with different parameters will be run to test the robustness and the influence each parameter. Only one parameter is variable for each group. For example, for a group of *Request Update Time*, the parameter of annealing, and DRT constant are the same, the only variable is request update time. Obviously, there are in total three groups of parameters to be tested, namely group of request update time, annealing and DRT constant. For each group, except the tested parameter, all other parameters are control parameters, which should be consistent with the base scenario, a relatively reasonable and easy-computing scenario.

Aside from above-mentioned parameters, it is of great significance to validate the sensitivity and robustness of the model. Although the input initial population file is the same for all scenarios to control variables, it is essential to test whether the result of fleet size and deployment is independent of the randomly selected initial new DRT caller. Two different sensitivity analysis will be conducted in the project, 10% new DRT caller with six different random seeds and 10%, 20%, 30% new DRT caller, each with two different random seeds. The former analysis is to show the influence of initial random location of new DRT caller and the latter analysis is to test whether the initial number of new DRT caller will result in different optimal fleet size and deployment.

Table 6: Overview of sensitivity analysis: sensitivity of different random seeds and sensitivity of different initial proportion of new DRT caller

Sensitivity analysis	Random seed	Proportion of new DRT caller
Random generated new DRT caller	1-12	10%
	1-2	10%
Different proportion of new DRT caller	13-14	20%
	15-16	30%

5.2 Overview of all scenarios

Overall 12 different scenarios for sensitivity analysis and 12 scenarios for variables will be run in the simulation. For better computational performance, in the base scenario, a request will update every 10 minutes. Thus, it is to be figured out in Group A whether relatively long update interval will influence the result in terms of optimal fleet size, deployment strategy and the waiting time. The assumption that annealing can improve the stability of the system and reach a more reliable equilibrium will be tested in Group B. According to existing research, linear walk scoring with 800m limitation is chosen in the base scenario, but whether different walk scoring result in different output still needs to be tested in Group C. Bonus for ride-sharing passengers based on the maximum number of passengers share the ride together seems to be an efficient approach to improve vehicle occupancy. Group D is to explore the improvement. Request accepting constraints is highly related to vehicle occupancy. Imagining an extreme situation where all agents can wait and travel forever long and where no request will be rejected, vehicle occupancy should be always close to eight, but the wait time and travel time might be long. Therefore, in group E, the influence of different accepting constraints is to be figured out.

6 Data analysis

6.1 Sensitivity analysis

Ensuring the result of the simulation is independent of initial population files, sensitivity analysis is necessary for the model. The result should not rely on the locations of random generated new DRT caller at iteration 0 as well as should not depend on the initial ratio of new DRT caller and DRT request. Therefore two different sensitivity analysis is conducted to test the robustness of the model, sensitivity of different random seeds and sensitivity of different initial ratio. Sensitivity analysis is also of great significance to analyze the influence of parameters.

6.1.1 Overall performance

From Table 7, it is observed that the model is relatively robust in terms of avg. executed score, mode share, average access walking time, average in-vehicle time, number of passengers per vehicle, max fleet size despite different random seeds; while the model is varied from the perspective of avg. computation time, average waiting time, vehicle kilometers traveled, empty kilometers traveled and average vehicle occupancy. The range and deviation caused by random seed can be found in Table 9. The indicator with CV(Coefficient of variance) more than 0.1 is regarded as a sensitive indicator. Later the range and deviation of parameter analysis will be compared with the of sensitivity analysis. If the result is similar to sensitivity analysis, it notes that the variance may not come from the different parameters.

The result of Table 8 shares similarities with Table 7, but the variance of average computation time, average executed score, average wait time, vehicle kilometers traveled, empty kilometers traveled, number of passengers per vehicle is higher. It shows that different initial proportion of new DRT caller leads to the higher variance of indicators, but these sensitive indicators do not show any tendency with the increasing fraction of initial new DRT caller. It is noted that although the difference of average executed scores is only 3, it is considered as a huge gap due to the relatively small marginal disutility of traveling. For example, marginal disutility of DRT is only -4 util/hr, the average difference of 3 already means that each passenger travels 45 minutes more with DRT.

Table 7: Overall performance for sensitivity analysis of different initial ratio

Random seed		1	2	3	4	5	6
Avg. Computation time		619.80s	608.59s	625.92s	616.55s	619.79s	743.78s
Avg. executed score		104.28	103.85	103.18	103.36	103.99	104.33
Mode share	new DRT call	0.0352	0.0339	0.0342	0.0339	0.0356	0.0342
	DRT request	0.8510	0.8507	0.8431	0.8529	0.8503	0.8504
	Walk	0.1138	0.1153	0.1226	0.1132	0.1141	0.1154
Average time	Access walking	875.39s	874.42s	875.31s	875.14	874.21s	874.82s
	Waiting	306.49s	398.66s	538.46s	520.25s	352.91s	304.59s
	In-vehicle	291.08s	290.67s	290.84s	291.74s	294.77s	297.03s
Vehicle kilometers traveled		27352	32068	37131	35436	30672	27485
Empty kilometers traveled		1675	3180	4898	4214	2664	1669
No. of passengers per vehicle		25.18	26.06	25.62	26.17	24.88	25.88
Average vehicle occupancy		5.20	4.64	3.87	4.20	4.92	5.21
Max fleet size		252	297	294	304	276	255
Random seed		7	8	9	10	11	12
Avg. Computation time		637.82s	599.22s	630.80s	444.62s	605.80s	755.33s
Avg. executed score		103.24	103.21	103.64	103.68	104.15	104.14
Mode share	new DRT call	0.0325	0.0380	0.0345	0.0347	0.0329	0.0333
	DRT request	0.8516	0.8489	0.8502	0.8483	0.8496	0.8511
	Walk	0.1159	0.1132	0.1153	0.1170	0.1174	0.1155
Average time	Access walking	875.84s	873.88s	874.51s	874.13s	874.45s	875.60s
	Waiting	550.86s	488.83s	442.50s	428.03s	358.16s	351.29s
	In-vehicle	291.15s	302.75s	296.08s	293.93s	297.71s	294.14s
Vehicle kilometers traveled		36200	34527	33589	33073	29968	30503
Empty kilometers traveled		4627	3574	3516	3623	2395	2721
No. of passengers per vehicle		27.23	23.36	25.62	25.48	26.79	26.52
Average vehicle occupancy		4.07	4.30	4.37	4.48	4.93	4.93
Max fleet size		287	332	294	299	272	271

Table 8: Overall performance for sensitivity analysis of different initial ratio

Fraction of initial new DRT caller		10%		20%		30%	
Random seed		1	2	13	14	15	16
Avg. Computation time		619.83s	608.59s	643.87s	516.75s	707.64s	447.89s
Avg. executed score		104.28	103.85	102.89	103.80	104.30	101.19
Mode share	new DRT call	0.0352	0.0339	0.0335	0.0337	0.0318	0.0313
	DRT request	0.8510	0.8507	0.8448	0.8446	0.8491	0.8466
	Walk	0.1138	0.1153	0.1217	0.1217	0.1191	0.1221
Average time	Access walking	875.39s	874.42s	876.27s	876.36s	875.72s	876.23s
	Waiting	306.49s	398.66s	597.17s	422.32s	343.35s	813.93s
	In-vehicle	291.08s	290.67s	295.30s	298.89s	291.37s	281.05s
Vehicle kilometers traveled		27352	32068	38780	31583	29143	43195
Empty kilometers traveled		1675	3180	5417	2938	2379	8144
No. of passengers per vehicle		25.18	26.06	26.23	26.05	27.73	28.07
Average vehicle occupancy		5.20	4.64	3.70	4.61	5.10	3.17
Max fleet size		252	297	300	292	248	279

Table 9: Range and deviation of 16 random seeds

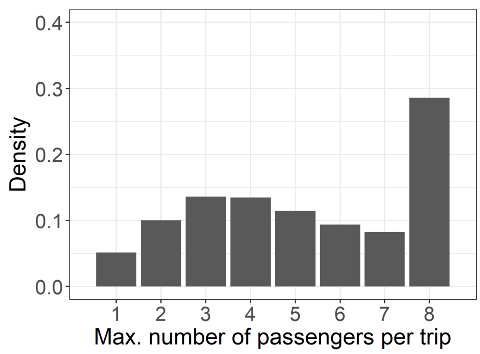
	Max	Min	Average	Std.	CV
Avg. computation time	755.33	444.62	625.67	77.02	0.12
Avg. executed score	104.33	103.18	103.75	0.43	0.00
Mode share - new DRT call	0.04	0.03	0.03	0.00	0.04
Mode share - DRT request	0.85	0.84	0.85	0.00	0.00
Mode share - Walk	0.12	0.11	0.12	0.00	0.02
Avg. access walk time	875.84	873.88	874.81	0.64	0.00
Avg. wait time	550.86	304.59	420.09	88.53	0.21
Avg. in-vehicle time	302.75	290.67	294.32	3.64	0.01
Vehicle kilometers traveled	37131.00	27352.00	32333.67	3228.10	0.10
Empty kilometers traveled	4898.00	1669.00	3229.67	1052.73	0.33
No. of passengers per vehicle	27.23	23.36	25.73	1.00	0.04
Avg. vehicle occupancy	5.21	3.87	4.59	0.45	0.10
Max fleet size	332.00	252.00	286.08	22.35	0.08

6.1.2 Ridership

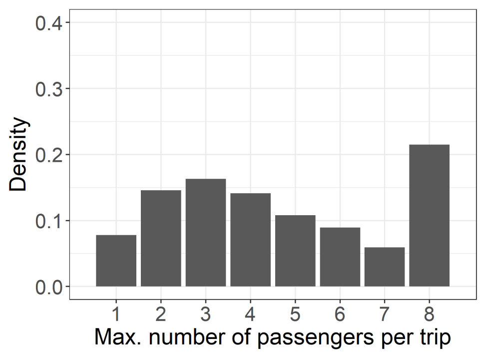
The distribution of ridership varies in the different group of random seeds, but all follow a distribution with the highest density in the middle and gradually decreasing tail on both sides. In all random seeds, there are more vehicles with 8 passengers than 7 passengers because the maximum number of passengers share ride is 8. If more than 8 passengers are willing to board, the exceeding passengers have to wait for the next minibus. It can be imagined that if the vehicle is infinity large, there will be probably a long tail on the right side of the graph. Besides, with the increasing initial proportion of new DRT caller, no specific patterns can be observed. The uncertainty of the distribution of vehicle occupancy may be from the dynamic passenger-vehicle matching. Due to the real-time matching and dynamic routing, randomness cannot be avoided in the system. A tiny difference in the initial matching may result in total different vehicle routing and vehicle occupancy. Besides, as a relatively small city, the traffic demand of Sioux Falls is probably limited for higher ridership and more incentives of ride-sharing are still needed to be tested.

Figure 11: Ridership during the day for sensitivity analysis of random seeds (Part A)

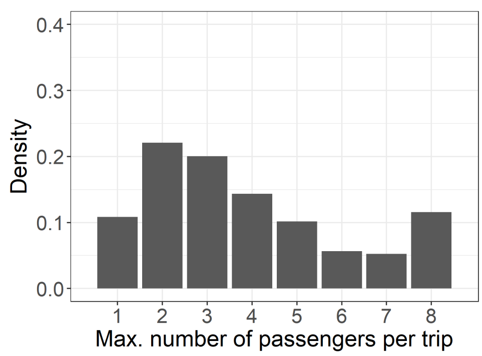
(a) Random seed 1



(b) Random seed 2



(c) Random seed 3



(d) Random seed 4

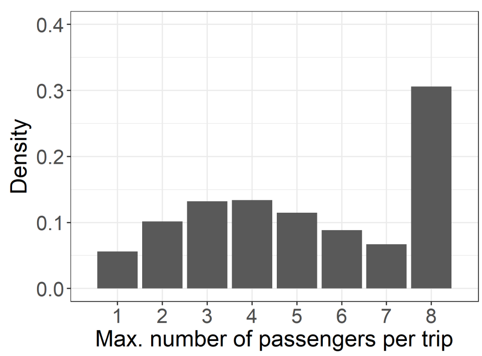
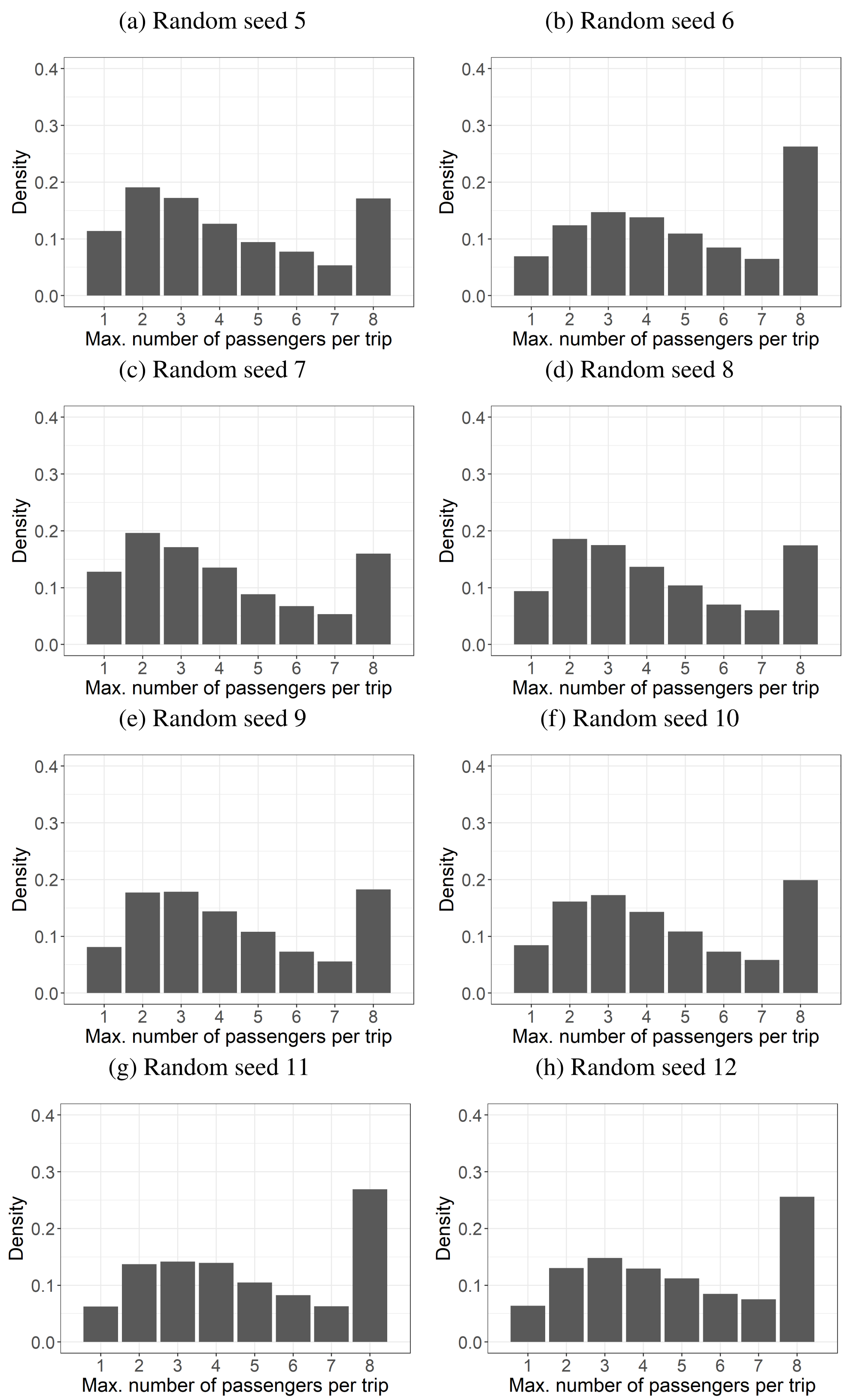
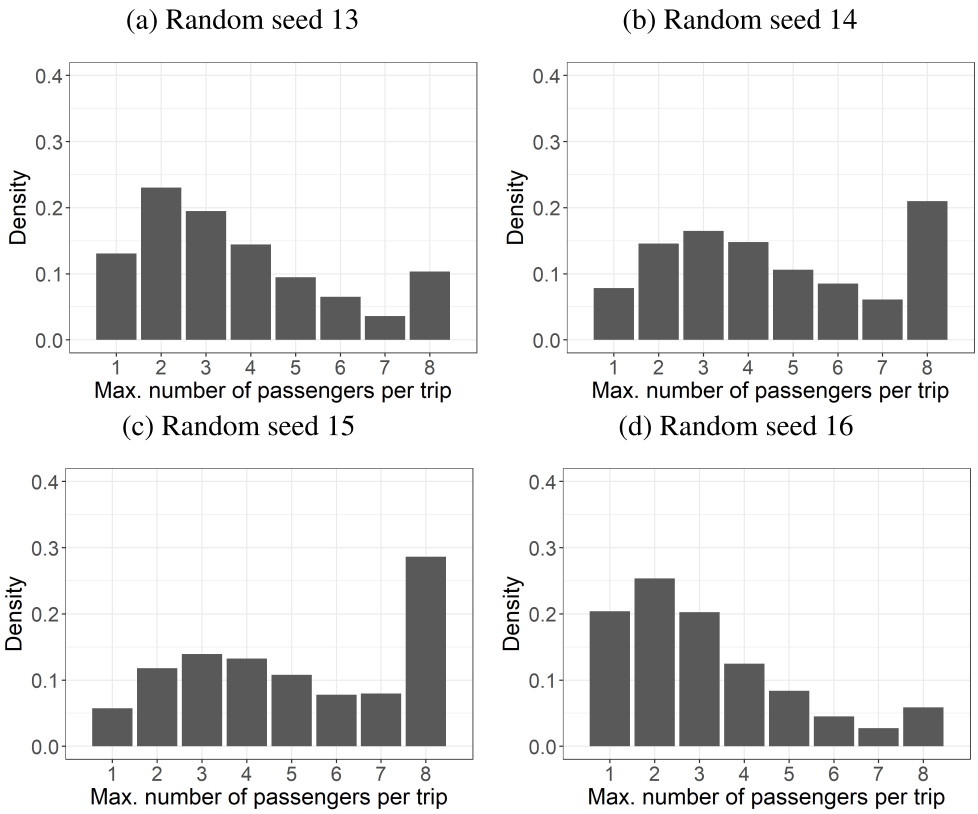


Figure 12: Ridership during the day for sensitivity analysis of random seeds (Part B)



(a) Random seed 5
(b) Random seed 6
(c) Random seed 7
(d) Random seed 8
(e) Random seed 9
(f) Random seed 10
(g) Random seed 11
(h) Random seed 12

Figure 13: Ridership during the day for sensitivity analysis of random seeds (Part C)

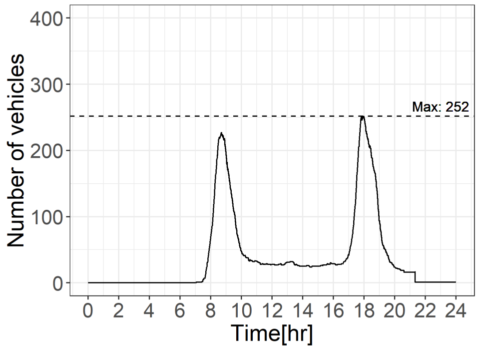


6.1.3 Fleet size

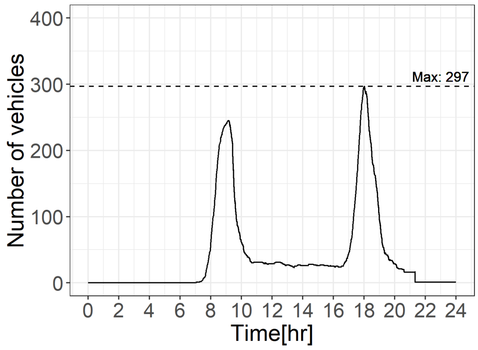
As shown in Figure 14 and Figure 15, the two peaks show the temporal distribution of fleet size is in line with the travel demand. During peak hour, around 200-300 vehicles are needed to satisfy the demand while during off-peak hour, only 20 - 50 vehicles are in use. In most random seeds, morning peak starts from 07:30 and ends at 09:30, while afternoon peak starts from 16:45 and ends at 18:45. In some random seeds with more flat peak, the above mentioned time may be postponed, for example in random seed 16, the end time of afternoon should be postponed to 19:45. In addition, in 15 out of 16 random seeds more vehicles observed in the afternoon peak than in the morning peak, and the exception is random seed 4. The maximum number of fleet size is around 250-300. The shape of the two peaks varies a lot among different random seeds.

Figure 14: Fleet size for sensitivity analysis (Part A)

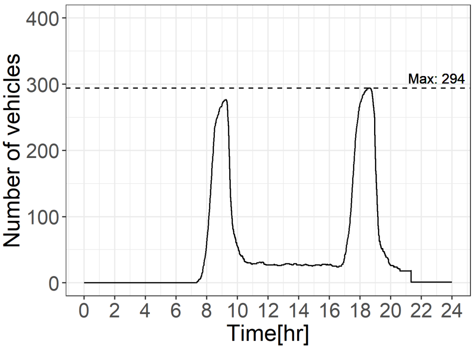
(a) Random seed 1



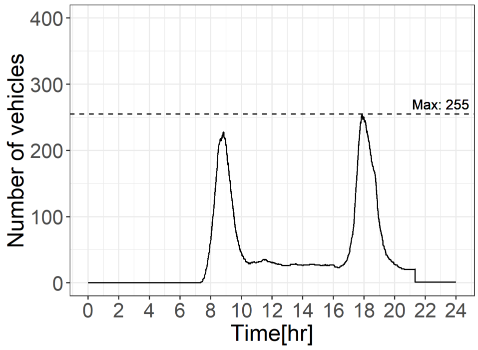
(b) Random seed 2



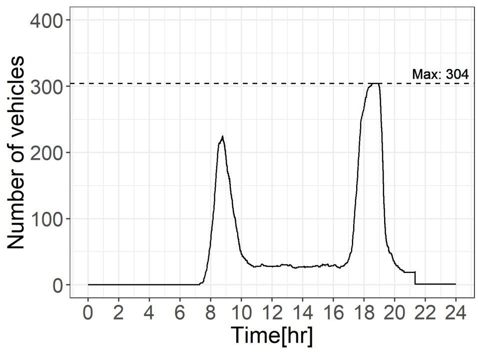
(c) Random seed 3



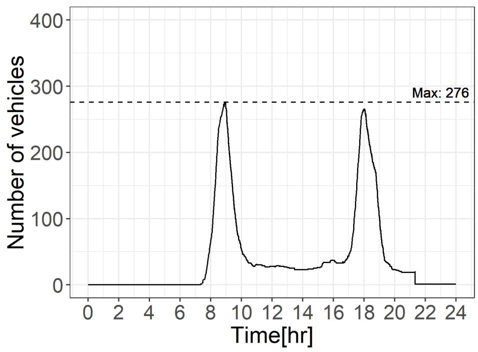
(d) Random seed 4



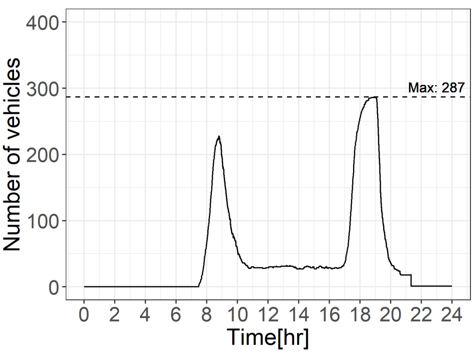
(e) Random seed 5



(f) Random seed 6



(g) Random seed 7



(h) Random seed 8

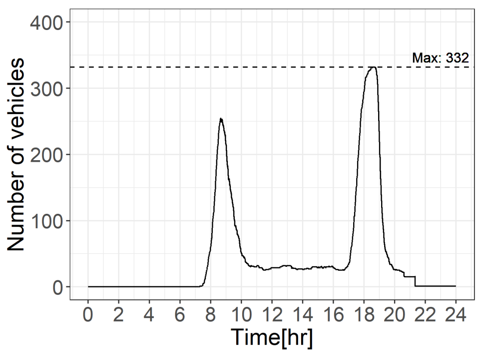
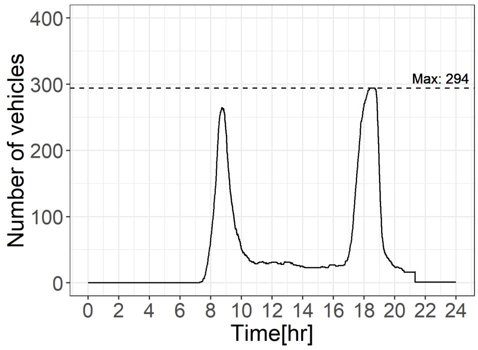
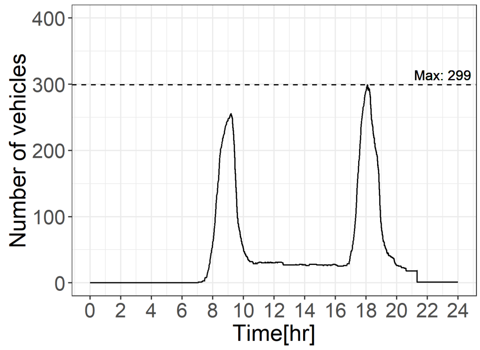


Figure 15: Fleet size for sensitivity analysis (Part B)

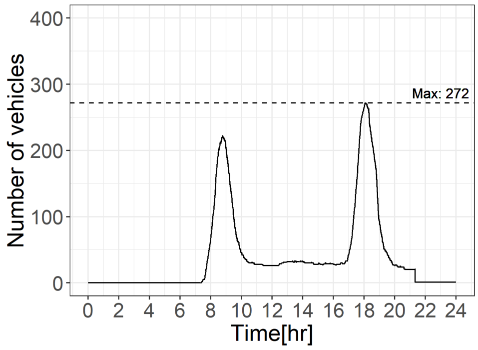
(a) Random seed 9



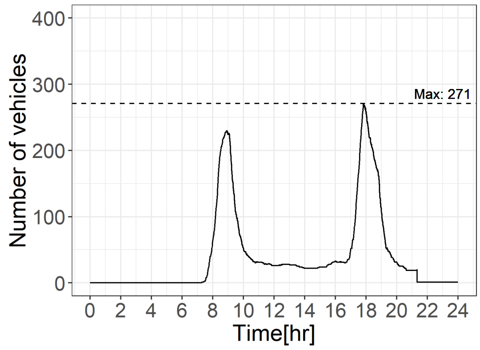
(b) Random seed 10



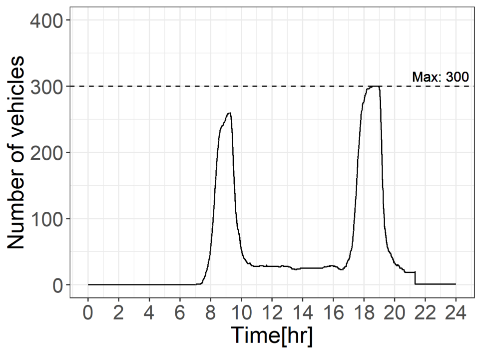
(c) Random seed 11



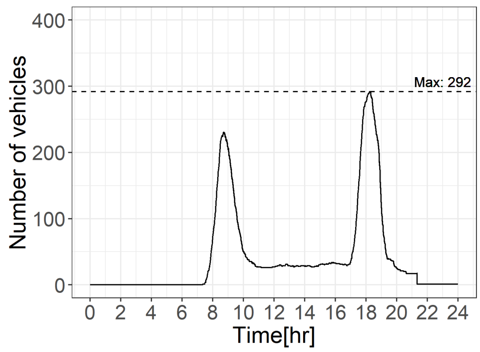
(d) Random seed 12



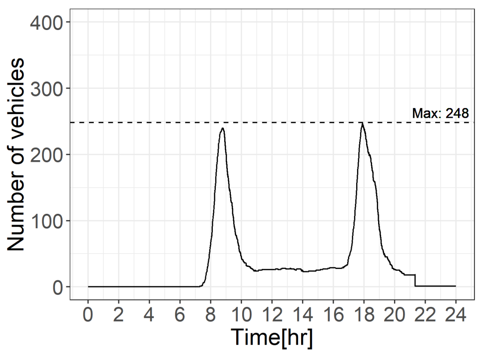
(e) Random seed 13



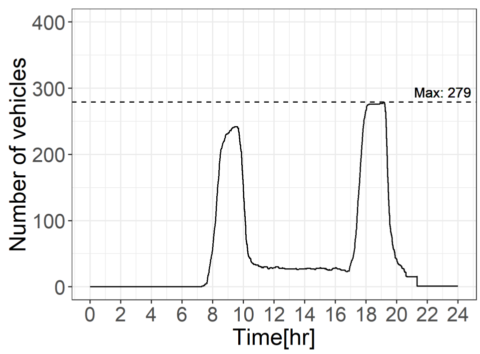
(f) Random seed 14



(g) Random seed 15



(h) Random seed 16

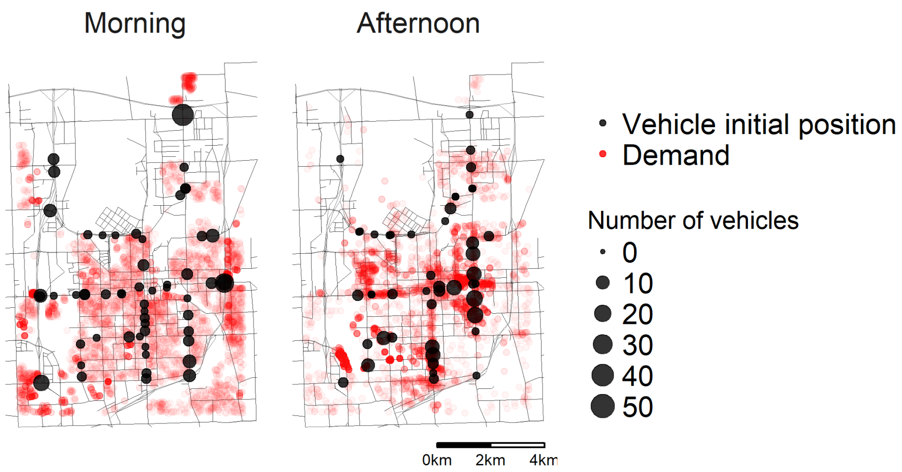


6.1.4 Fleet deployment

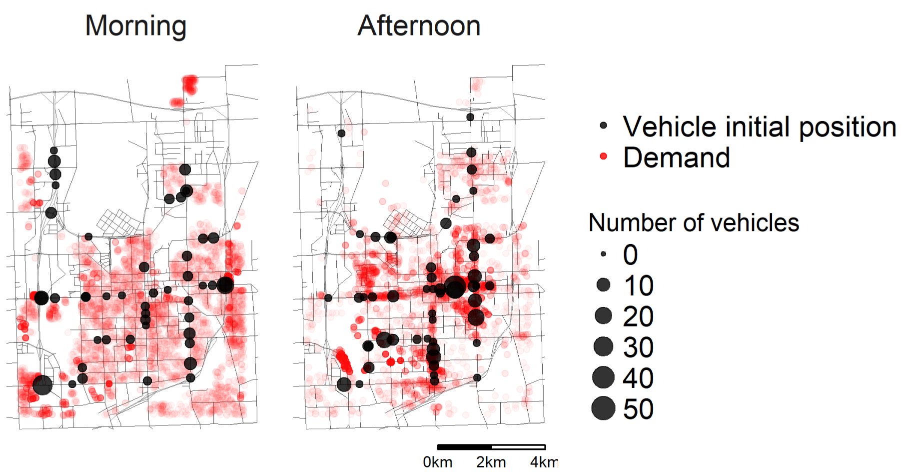
Fleet deployment shows the location where the vehicle is generated during morning and afternoon peak. In both morning and afternoon peak, the spatial distribution of initial vehicle position is close to travel demand. The most popular stops may attract more than 50 new calling vehicles around. During morning peak, some stops in the surroundings are more attractive than stops in the city center; during afternoon peak, some stops in the city center are more popular. Fleet deployment strategies of all random seeds are more or less similar, therefore only two typical graphs are selected in Fig. 16. The difference between these two typical graphs is whether huge amount of vehicles or no vehicle are generated in the northeast corner of the city.

Figure 16: Fleet deployment for sensitivity analysis of random seeds

(a) Random seed 1



(b) Random seed 3



6.2 Group of request update time

6.2.1 Overall performance

Similar to sensitivity analysis, it seems that mode share, average access walking time, average in-vehicle time are independent of request update time. For avg. computation time, it is plausible that the simulation of request update time of 30s is time-consuming, which reviews of all not accepted requests every 30 seconds, however, it is surprising that there is no much difference among simulations of other request update time. In addition, the result of the average executed score, average waiting time, vehicle kilometers traveled, empty kilometers traveled, max number of passengers per trip and max fleet size violates the expectation. With the decreasing request update time, the average waiting time increases dramatically but the vehicle occupancy(max number of passengers per trip) drops. As a result, more fleet size is needed to serve the same demand for a much more frequent updating scenario, which also causes the decrease of vehicle kilometers traveled and empty kilometers traveled as well as the drop of the average executed score. More exploration is needed to explain the weird result.

Table 10: Overall performance for group of request update time

Request update time		800s	700s	600s	500s	400s	300s	30s
Avg. Computation time		584.4s	675.9s	619.8s	528.0s	530.2s	644.7s	2852.1s
Avg. executed score		104.14	104.19	104.28	102.47	102.92	101.39	102.02
Mode share	new DRT call	0.0311	0.0309	0.0352	0.0326	0.0355	0.0355	0.0355
	DRT request	0.8488	0.8480	0.8510	0.8485	0.8451	0.8417	0.8405
	Walk	0.1201	0.1210	0.1138	0.1189	0.1179	0.1227	0.1240
Average time	Access walking	875.69s	878.04s	875.40s	875.13s	874.63	876.11	877.55
	Waiting	385.61s	373.48s	306.49s	683.67s	564.46	715.98	737.72
	In-vehicle	296.84s	294.98s	291.08s	284.26s	293.82	290.48	299.83
Vehicle kilometers traveled		31396	30432	27352	41351	37843	41766	44242
Empty kilometers traveled		2812	2489	1675	7101	5186	7125	8507
No. of passengers per vehicle		28.33	28.41	25.18	27.04	23.87	24.67	24.69
Average vehicle occupancy		4.72	4.82	5.20	3.38	3.80	3.35	3.20
Max fleet size		258	270	252	277	315	307	322

In order to eliminate the interference of uncertainty of the model itself, as the model is not robust in some parameters, range and deviation with sensitivity analysis is compared in Table 16. Compared to Table 9, the CVs of other indicators of different request update time are all higher than the of different random seeds, which means different update time does influence the performance. The average computation time is much higher with huge deviation due to the long computation time of request update time 30s.

The result can be divided into two groups, request update time less than 600s (the left group in the table) and request update time 600s and more (the right group in the table). Compared to the right group (high update frequency), the left group (low update frequency) has higher avg. executed score, lower wait time, shorter vehicle kilometers traveled and empty kilometers traveled, higher average vehicle occupancy and less fleet size.

Table 11: Range and deviation of different update time

	Max	Min	Average	Std.	CV
Avg. computation time	2852.10	528.00	919.30	854.08	0.93
Avg. executed score	104.28	101.39	103.06	1.17	0.01
Mode share - new DRT call	0.04	0.03	0.03	0.00	0.06
Mode share - DRT request	0.85	0.84	0.85	0.00	0.00
Mode share - Walk	0.12	0.11	0.12	0.00	0.03
Avg. access walk time	878.04	874.63	876.08	1.27	0.00
Avg. wait time	737.72	306.49	538.20	181.37	0.34
Avg. in-vehicle time	299.83	284.26	293.04	5.04	0.02
Vehicle kilometers traveled	44242.00	27352.00	36340.29	6575.00	0.18
Empty kilometers traveled	8507.00	1675.00	4985.00	2689.80	0.54
No. of passengers per vehicle	28.41	23.87	26.03	1.87	0.07
Avg. vehicle occupancy	5.20	3.20	4.07	0.83	0.20
Max fleet size	322.00	252.00	285.86	28.45	0.10

6.2.2 Experience of passengers

Total wait time is consists of two parts in the simulation: scheduled wait time and wait time after accept. The former wait time is the wait time for the request to be accepted, which is a multiple of request update time plus one reaction time; the latter wait time is the wait time for vehicle's arrival after accept, which is theoretically limited by pick-up tolerance (15 min in the simulation). Different request update time can only influence the former wait time, but contrary to expectation, less request update time result in more scheduled wait time as well as total wait time.

Figure 17: Experience of passengers for group of request update time

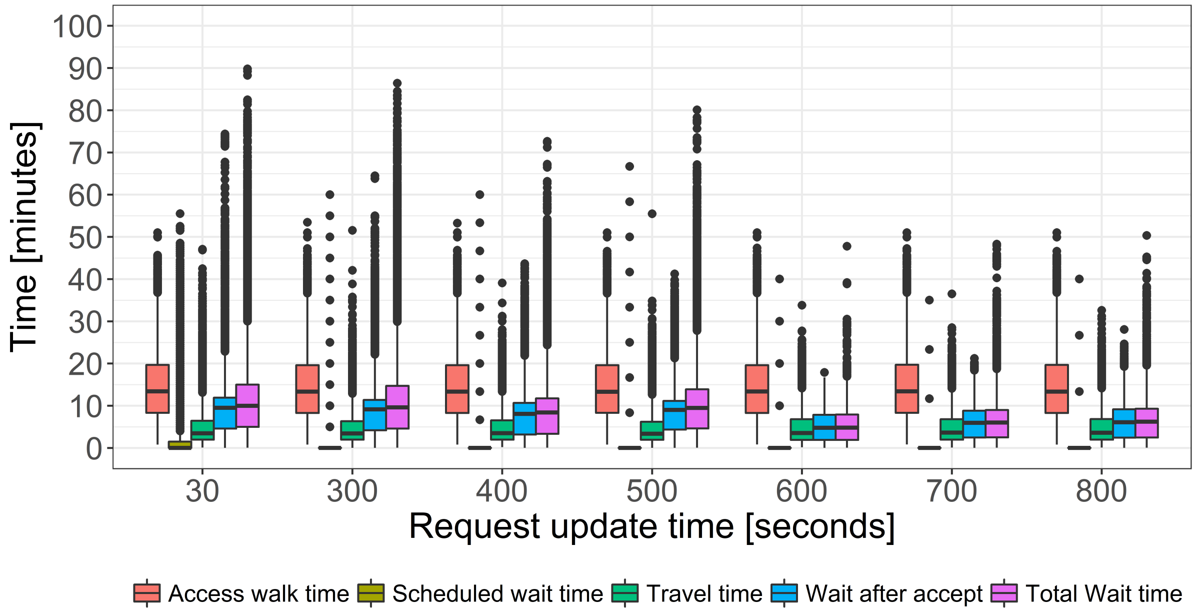


Table 12: Analysis of average wait time

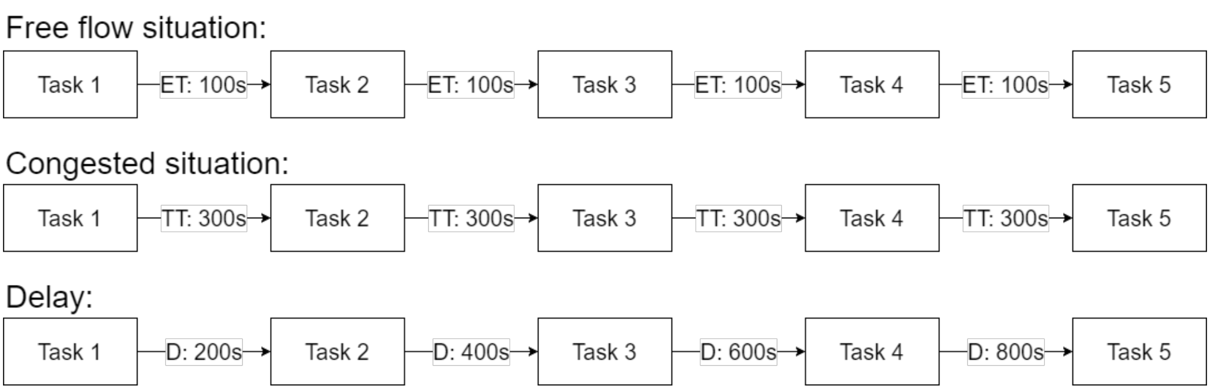
Request update time		800s	700s	600s	500s	400s	300s	30s
Wait time[s]		385.61	373.48	306.49	683.67	564.46	715.98	737.72
Scheduled wait time[s]		22.53	23.43	6.00	172.87	104.63	196.73	173.96
Wait time after accept[s]	Dwell time	55.86	56.15	54.84	57.44	56.27	56.47	56.45
	Others	307.22	293.90	245.65	453.35	403.55	462.80	507.30

It is possible that request update time 300-600s is so long that may result in a huge negative score, but request update 30s is much more acceptable. Thus, after 100 iterations, agents with request update time 300-600s tend to avoid request update, but agents with request update time 30s still update, which result in longer scheduled wait time. Besides the various total wait time is also caused by wait time after accept. Although the maximum wait time after accept should not exceed 15 min, the long tail of wait time after accept in Fig. 17 shows that some trips violate pick-up tolerance. Actually, the rule is still strictly followed, because it can only guarantee that no more detour is accepted but cannot guarantee the vehicle will arrive at the stop in time.

One possible explanation is that the long wait time after accept is from the consolidation of agents, which means maybe in some scenarios passengers prefer popular transport hub and board in some stops together which save the boarding time. Dwell time in the Table 12 describes the boarding time of other passengers between the request accepted and the arrival of the minibus. If consolidation exists, the average dwell time should be shorter, because, for example, the average dwell time for 8 passengers boarding in the same stop is (10 + 2 * 8) / 8 = 3.25s but the average dwell time for 8 passengers boarding in the different stop is (10 + 2) * 8 / 8 = 12s. However, the dwell time is similar to all different scenarios, the consolidation is not observed.

Another possible explanation is that the time loss due to congestion can still enlarge the wait time after accept. At the very beginning, if a lot of requests are accepted by a vehicle before congestion appears, when congestion appears, all the request will be delayed, the wait time of the last request in the list may be postponed a multiple of delays. As shown in Fig. 18, if five tasks come at the very beginning, they will be accepted because the max wait time of the last passenger is only 500s. However, later with the congestion, the travel time will be three times more, the delay of the last passenger will be (300 - 100) * 4 = 800s. As an accepted request cannot submit their request to other vehicles, the delay of a passenger will be multiplied by the number of tasks prior to picking up the passenger. With frequent update time, it is more likely for a vehicle to accept lots of request at the beginning. When congestion begins, these requests already lose the chance to submit to better vehicles. Besides, after only 30s of rejection, probably the request only satisfy the upper or lower boundary of the constraints; while after 300s of rejection, probably the request has more options to compare and can choose a better vehicle. Above mentioned explanations are just some guesses, more exploration is needed to explain the result.

Figure 18: Delay caused by congestion, ET: Estimated travel time, TT: Travel time, D: Delay



6.2.3 Vehicle kilometers traveled

In spite of the fact that vehicle kilometers traveled(VKT) and empty vehicle kilometers traveled(EVKT) drops a lot with increasing request update time, passenger kilometers traveled(PKT) and passenger estimated kilometers(PEKT) both go up. Longer PKT and PEKT means more detouring given that origins and destinations are same for all scenarios. To be concluded, more detouring in the scenario of request update time 600s encourage high vehicle occupancy, which gives rise to low fleet size, less congestion and less VKT. Less congestion compensates for the time loss of detouring and result in less waiting.

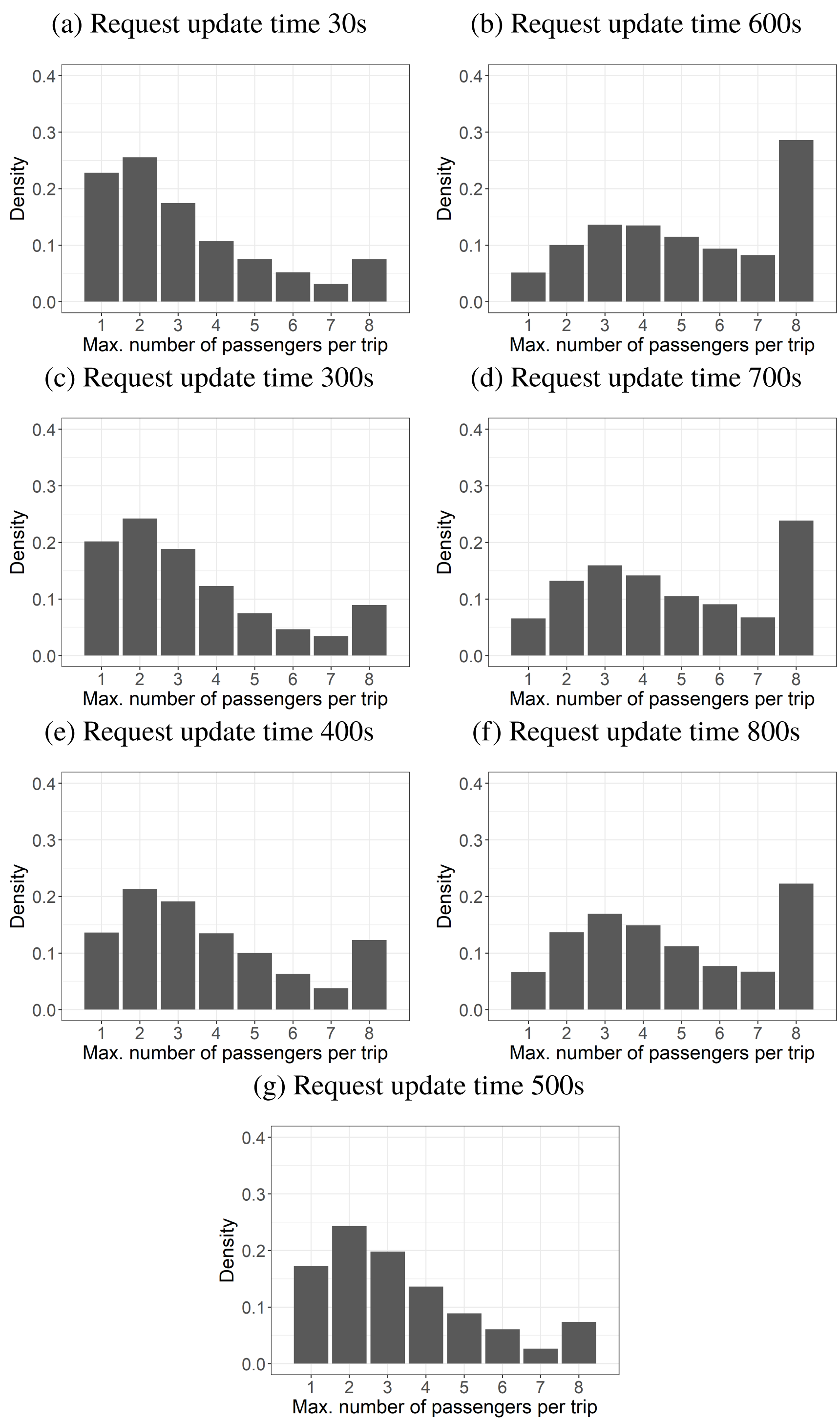
Table 13: Analysis of kilometers traveled for group of request update time, VKT: Vehicle kilometers traveled, EVKT: Empty vehicle kilometers traveled, PKT: Passenger kilometers traveled, PEKT: Passenger estimated kilometers traveled

Request update time	800s	700s	600s	500s	400s	300s	30s
VKT[km]	31396	30432	27352	41351	37843	41766	44242
EVKT[km]	2811.63	2489.34	1674.65	7101.10	5185.52	7124.78	8507.21
PKT[km]	67335	67127	67969	62140	64075 26	61738	61535
PEKT[km]	38706	38593	38793	38672	38694	38620	38575

6.2.4 Ridership

The distribution of ridership shares similar pattern within the group with high update frequency and the group with low update frequency. Most trips in the group with high update frequency are shared with 8 passengers, while the group with low update frequency has the most trips with only 1 or 2 passengers, which is the reason of the relatively high fleet size and high vehicle kilometers traveled.

Figure 19: Ridership during the day for group of request update time



(a) Request update time 30s

(b) Request update time 600s

(c) Request update time 300s

(d) Request update time 700s

(e) Request update time 400s

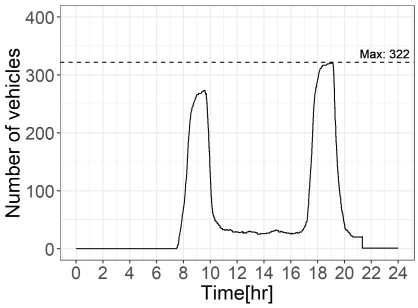
(f) Request update time 800s

(g) Request update time 500s

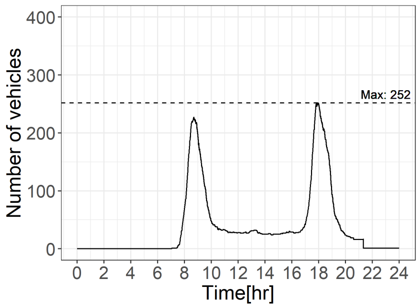
6.2.5 Fleet size

Figure 20: Fleet size for group of request update time

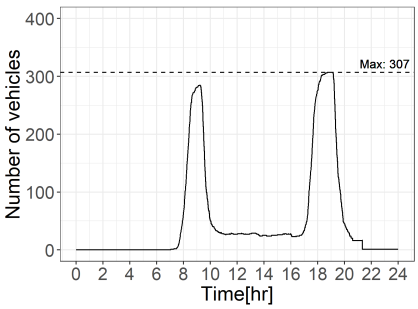
(a) Request update time 30s



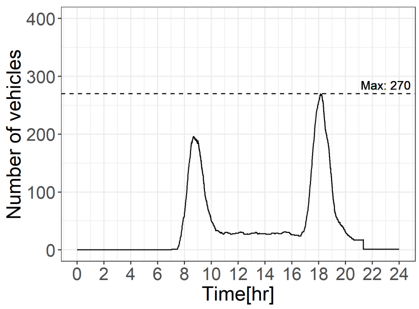
(b) Request update time 600s



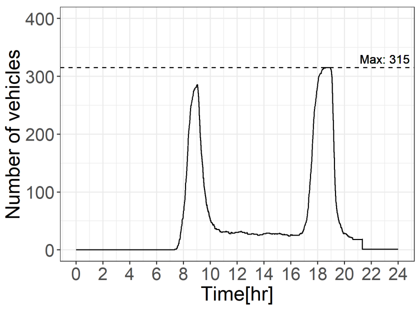
(c) Request update time 300s



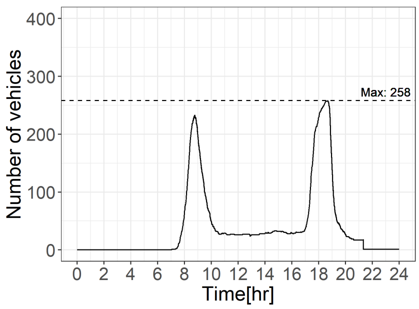
(d) Request update time 700s



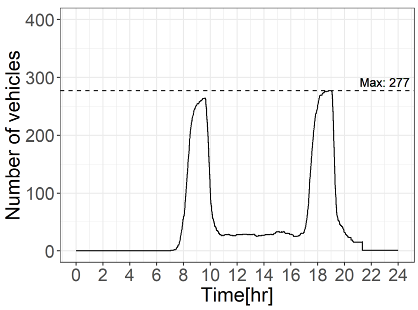
(e) Request update time 400s



(f) Request update time 800s



(g) Request update time 500s



Similar to the result of sensitivity analysis, there are more minibuses in the system during afternoon peak than during morning peak. The peak of the group with low update frequency is more flat while the peak of the group with high update frequency is sharper. Despite the different values of peaks, the number of vehicles during the off-peak hour is all round 50.

6.2.6 Fleet deployment

The result of fleet deployment does not show any pattern with the increasing request update time. The spatial distribution of the initial location of vehicles is similar to the result of sensitivity analysis.

6.3 Group of annealing

6.3.1 Overall performance

Table 14: Overall performance for group of annealing

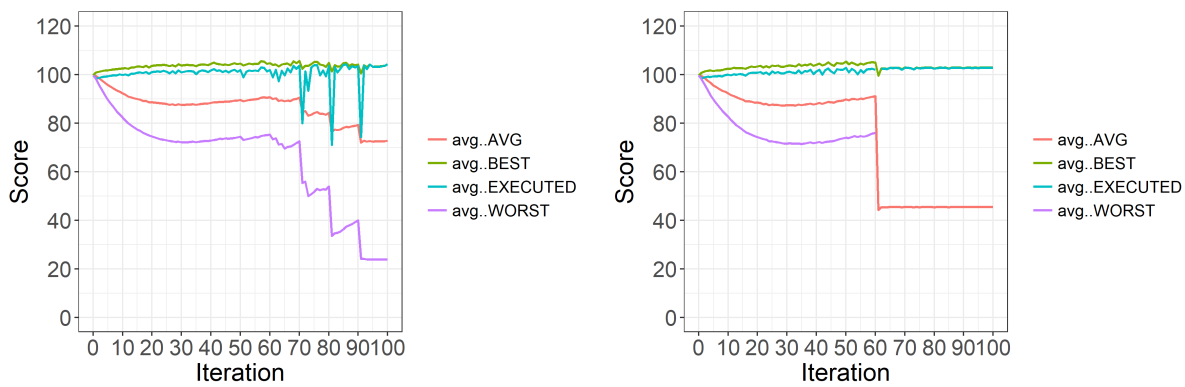
Annealing		Yes	No
Avg. Computation time		619.8s	573.63s
Avg. executed score		104.28	102.83
Mode share	new DRT call	0.0352	0.0636
	DRT request	0.8510	0.8334
	Walk	0.1138	0.1030
Average time	Access walking	875.40s	872.96s
	Waiting	306.49s	261.97s
	In-vehicle	291.08s	294.20s
Vehicle kilometers traveled		27352.33	26890.22
Empty kilometers traveled		1674.65	1253.81
No. of passengers per vehicle		25.18	14.11
Average vehicle occupancy		5.20	5.26
Max fleet size		252	356

The result of the group of annealing is opposite to both sensitivity analysis and the group of request update time. Once annealing is turned off, the mode share changes dramatically, the number of passengers per vehicle is halved and the number of fleet size skyrockets. Although both empty kilometers traveled and average wait time decreases without annealing, but the score also drops. Probably oversupply can explain the result.

6.3.2 Score and mode share

Figure 21: Score for group of annealing

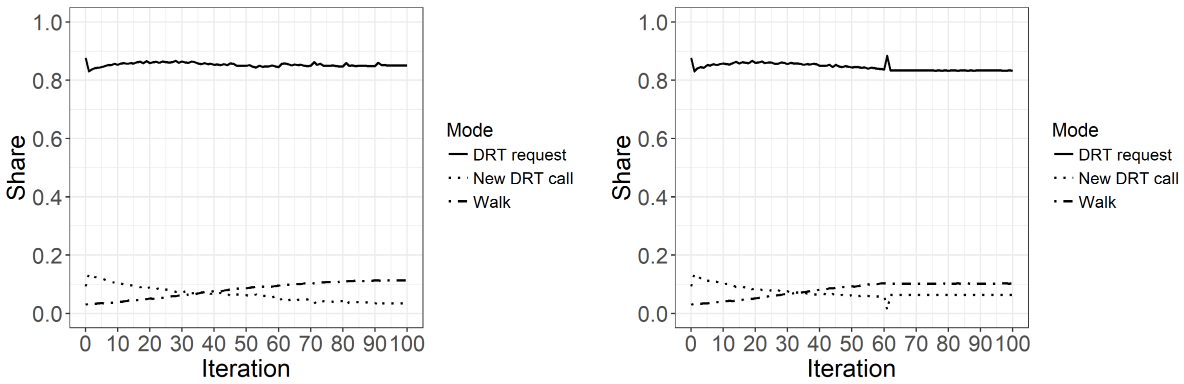
(a) With annealing (b) No annealing



From **??**, it is obvious that the score will hit the bottom whenever turning off the innovation but recover soon afterwards. With annealing, the score goes down four times at iteration 60, 70, 80, 90(the one at iteration 60 is not very visible); while the score only drops once at iteration 60 without annealing. From Figure 22, the balance between DRT request and new DRT call changes slightly whenever turning off the innovation but the ratio remains almost the same. However, without annealing, a peak can be captured at iteration 60 for both DRT request and new DRT call, because DRT request is more attractive in general. If the demand of DRT, which is the sum of DRT request and new DRT call, surpasses the supply, which is the new DRT call, the balance of the system will be destroyed and the exceeding part of demand will be aborted. Although the system recovers immediately at iteration 61, the terrible experience of DRT request at iteration 60 is already stored in the memory. Therefore, it may result in a relatively higher mode share of new DRT call. The extra vehicles will give rise to higher fleet size and lower number of passengers per vehicle.

Figure 22: Mode share for group of annealing

(a) With annealing (b) No annealing



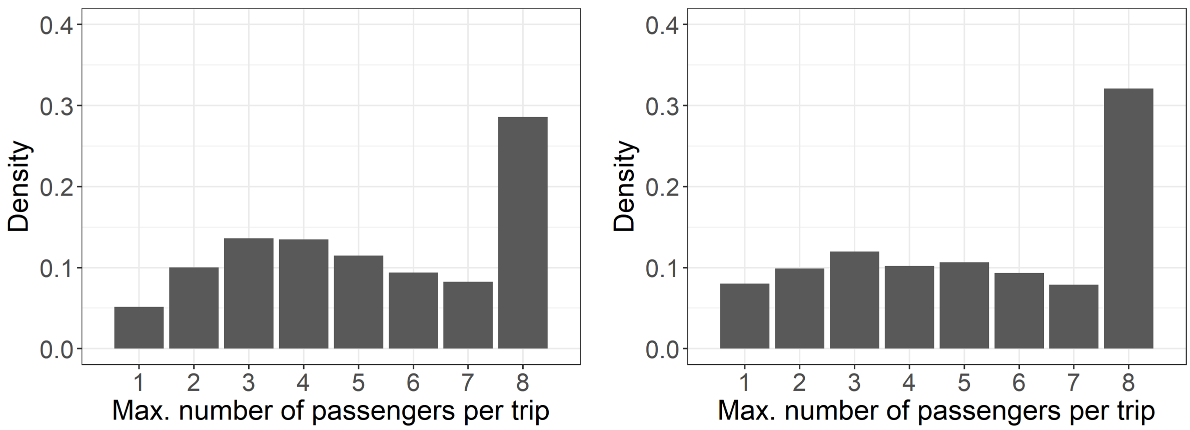
6.3.3 Number of passengers per vehicle

6.3.4 Ridership

The distribution of vehicle occupancy is similar and the distribution without annealing is more evenly distributed. It is also proved that annealing has few impact on ridership.

Figure 23: Ridership during the day for group of annealing

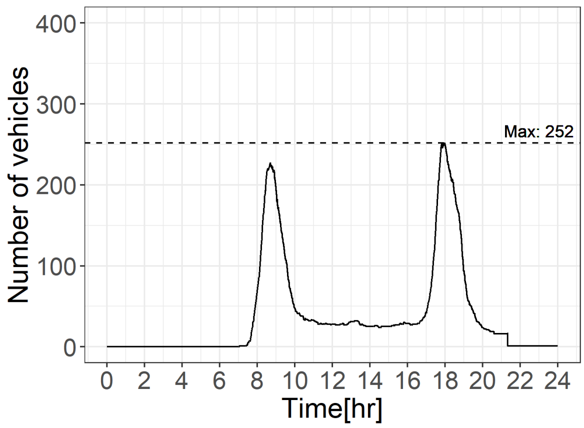
(a) With annealing (b) No annealing



6.3.5 Fleet size and deployment

Figure 24: Fleet size for group of annealing

(a) With annealing



(b) No annealing

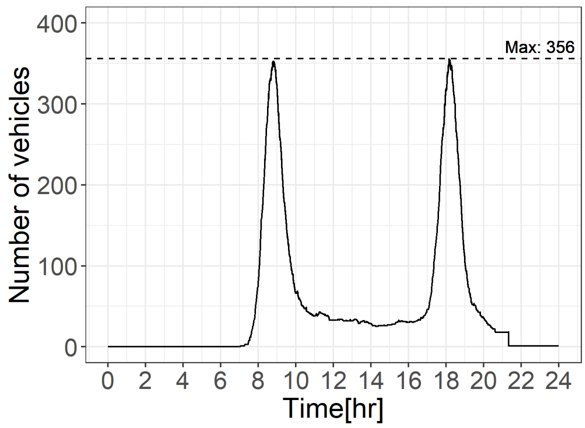
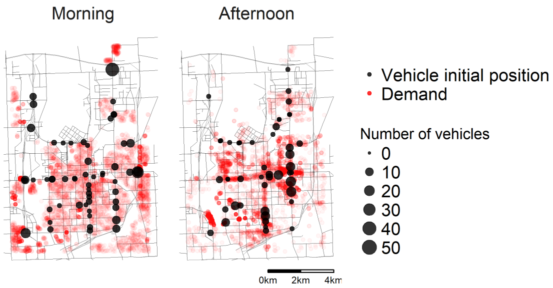
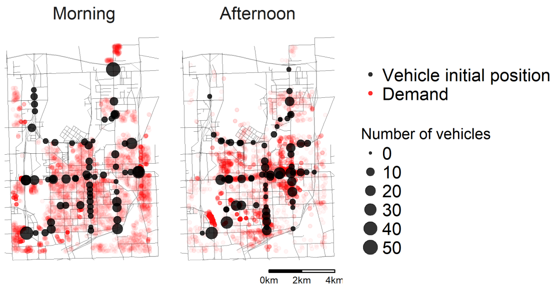


Figure 25: Fleet deployment for group of annealing

(a) With annealing



(b) No annealing



Given that the average vehicle occupancy remains the same and the slight decrease of average wait time, the increase of fleet size without annealing means that extra vehicles share the passengers. From Figure 25, it illustrates that the spatial distribution of the initial location of vehicles with and without annealing is almost the same, and the difference is just more vehicles are generated in some places. These vehicles share the request of the vehicles in the scenario of with annealing.

6.4 Group of new DRT call constant

6.4.1 Overall performance

Table 15: Overall performance for group of new DRT call constant

New DRT call constant		-10	-20	-30	-40	-50	-60
Avg. Computation time		524.46s	439.58s	619.80s	567.02s	621.37s	654.66s
Avg. executed score		103.38	102.60	104.28	103.50	101.37	100.40
Mode share	new DRT call	0.0329	0.0321	0.0352	0.0339	0.0346	0.0346
	DRT request	0.8663	0.8543	0.8510	0.8519	0.8512	0.8497
	Walk	0.1007	0.1136	0.1138	0.1141	0.1142	0.1157
Average time	Access walking	869.30s	873.05s	875.40s	874.39s	874.99s	875.58s
	Waiting	728.59s	688.5s	306.49s	340.97s	539.45s	605.05s
	In-vehicle	283.35s	289.49s	291.08s	295.15s	295.85s	291.56s
Vehicle kilometers traveled		42674	41692	27352	29228	36078	36790
Empty kilometers traveled		7848	6879	1675	2052	4949	4788
No. of passengers per vehicle		27.29	27.62	25.18	26.10	25.68	25.54
Avg. vehicle occupancy		3.35	3.40	5.20	5.00	4.13	4.00
Max fleet size		319	273	252	286	303	296

The result of the group of DRT constant is similar to the result of the group of request update time, average executed score, average wait time, vehicle kilometers traveled, empty kilometers traveled and average vehicle occupancy and max fleet size increases first and decreases afterwards with the increasing new DRT call constant. Compared to sensitivity analysis, the CV of all above mentioned indicators is much higher, which means that the variance is not only from randomness, but also influenced by new DRT call constant. The performance of the scenarios can be divided into two groups, groups with new DRT call constant -30 and -40 and group with other new DRT call constant.The number of passengers per vehicle is around 25-30 for almost all simulated scenarios, which also proves that the optimal value of new DRT call constant should around 25-30.

Table 16: A Tables long Caption

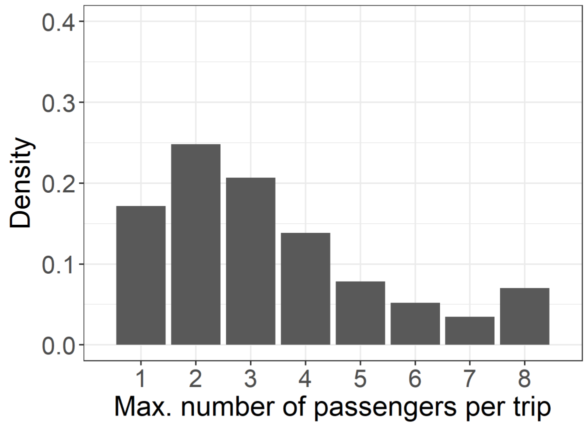
	Max	Min	Average	Std.	CV
Avg. computation time	654.66	439.58	571.15	79.21	0.14
Avg. executed score	104.28	100.40	102.59	1.46	0.01
Mode share - new DRT call	0.04	0.03	0.03	0.00	0.03
Mode share - DRT request	0.87	0.85	0.85	0.01	0.01
Mode share - Walk	0.12	0.10	0.11	0.01	0.05
Avg. access walk time	875.58	869.30	873.78	2.38	0.00
Avg. wait time	728.59	306.49	534.84	176.54	0.33
Avg. in-vehicle time	295.85	283.35	291.08	4.51	0.02
Vehicle kilometers traveled	42674.17	27352.33	35635.76	6283.49	0.18
Empty kilometers traveled	7847.57	1674.65	4698.32	2485.46	0.53
No. of passengers per vehicle	27.62	25.18	26.23	1.00	0.04
Avg. vehicle occupancy	5.20	3.35	4.18	0.78	0.19
Max fleet size	319.00	252.00	288.17	23.56	0.08

6.4.2 Ridership

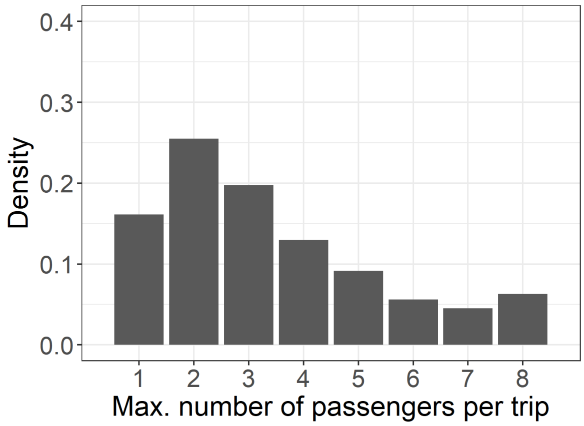
Still similar to the group of request update time, the vehicle occupancy of the well-performed group with value -30 and -40 is similar and with the highest density of 8 passengers; while the vehicle occupancy of the group with not so good result is also similar and with the highest density of 2 passengers. The shape of the distribution seems to be almost same for new DRT call constant -10 and -20, -30 and -40, and -50 and -60.

Figure 26: Ridership during the day for group of new DRT call constant

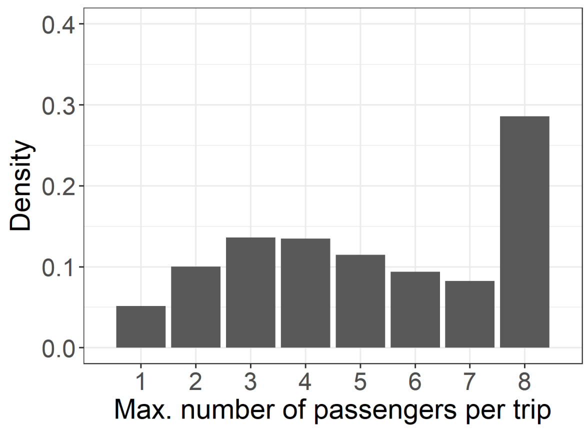
(a) New DRT call constant -10



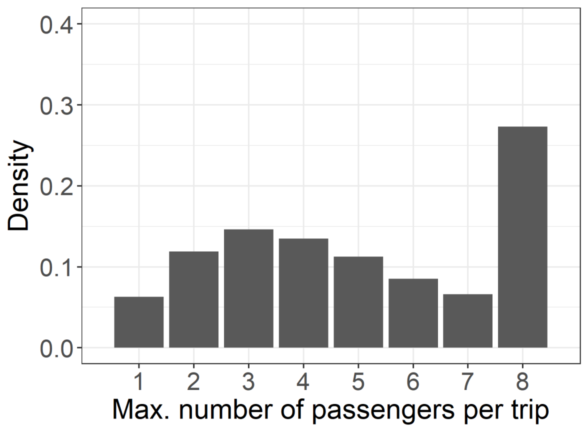
(b) New DRT call constant -20



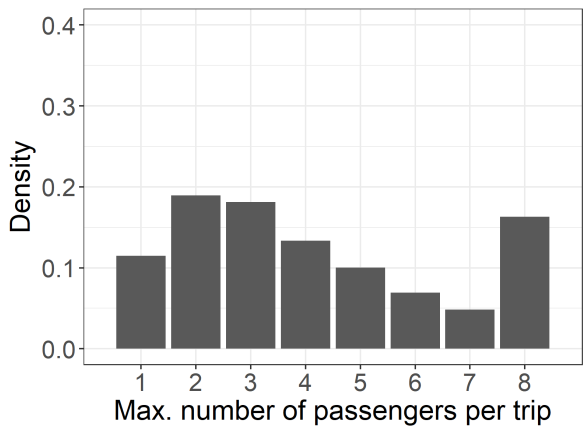
(c) New DRT call constant -30



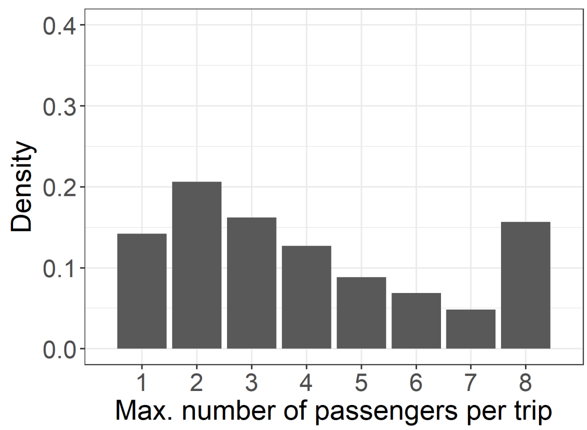
(d) New DRT call constant -40



(e) New DRT call constant -50



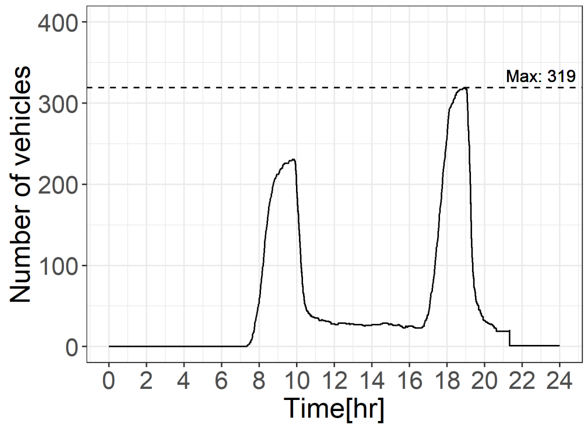
(f) New DRT call constant -60



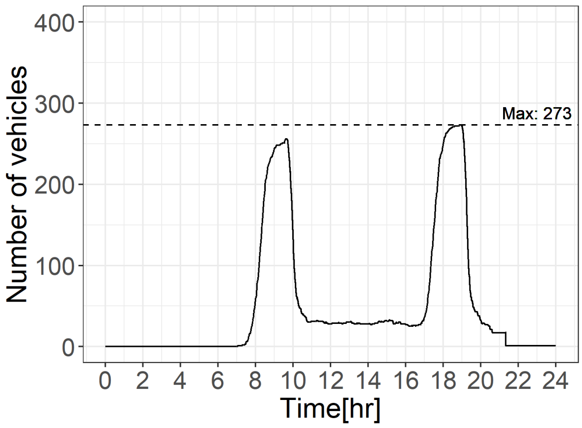
6.4.3 Fleet size

Figure 27: Fleet size for group of new DRT call constant

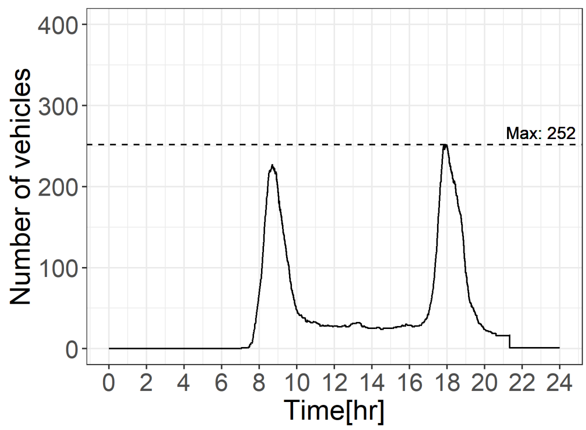
(a) New DRT call constant -10



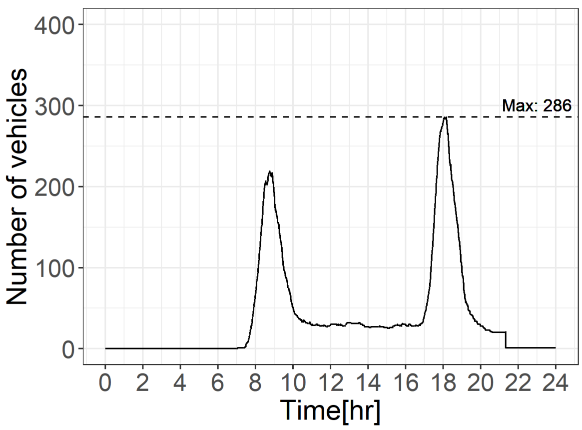
(b) New DRT call constant -20



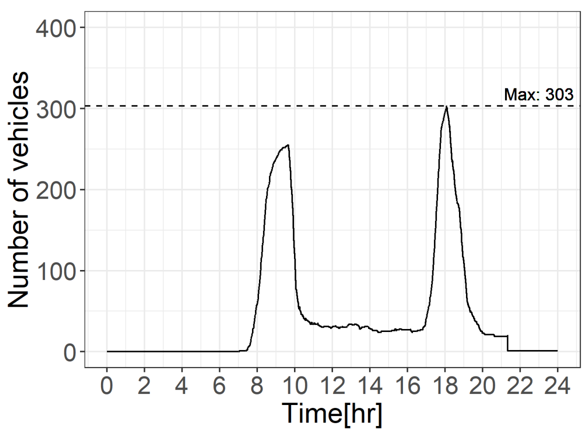
(c) New DRT call constant -30



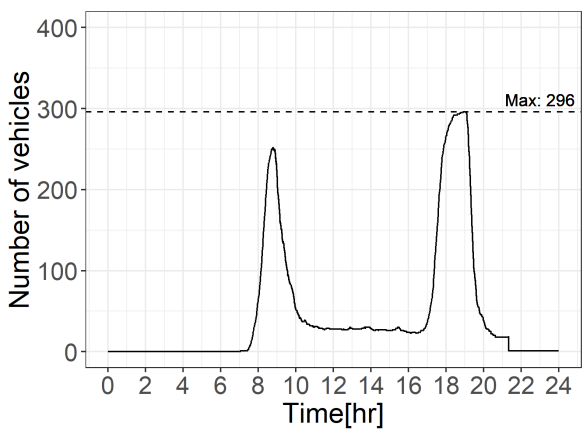
(d) New DRT call constant -40



(e) New DRT call constant -50



(f) New DRT call constant -60



From the result of fleet size, also similar conclusion can be drawn that the group with good performance has a sharper peak while the group with bad performance has a more flat peak. It is interesting that for the graph of new DRT call constant -50, the afternoon peak is sharp and the morning peak is flat; while the result of new DRT call constant -60 is totally the opposite. The result of fleet deployment of this group is similar to previous simulations, therefore the analysis of fleet deployment is neglected.

7 Conclusion and discussion

7.1 Conclusion

The paper shows the possibilities of the application of agent-based modeling in areas other than transport simulation. The purpose of the simulation is not only to model the real ATOD system but solve a complex transport optimization problem with the evolutionary algorithm. After the equilibrium, the system can indicate the appropriate fleet size and deployment on demand with high vehicle occupancy. The result confirms that with an appropriately designed trade-off, the simulation can output a reliable and meaningful result, which can help policy-makers decide how many minibuses is needed to satisfy the demand and how to deploy minibuses.

From data analysis, on one hand, it is an exciting discovery that indicators such as mode share, average executed score, access walk time, in-vehicle travel and number of passengers per vehicle do not change a lot with different parameters and input population file, which shows the reliability and robustness of the model under the circumstance of dynamic ride-sharing. Although the fleet size varies among all the simulated scenarios, the range of optimal fleet size can be determined, which should be around 250-300 in peak hour and 50 in off-peak hour. In addition, the fleet deployment is similar among ll simulation, which follows the pattern that the initial location of minibuses are around suburb in the morning, and gathers in some area of the city center in the afternoon.

On another hand, the variance of some indicators of the model is still large in the sensitivity analysis, such as wait time, vehicle kilometers traveled, empty kilometers traveled and average vehicle occupancy. The uncertainty of the model is probably from dynamic vehicle occupancy. Since the model takes the dynamic matching problem into account and routing reacts to the demand, a tiny difference in vehicle-passenger matching may change the route of many vehicles and result in total different vehicle occupancy as well as other indicators. Furthermore, better performance of these parameters is observed with the increasing request update time and the performance goes up and comes down with increasing new DRT call constant. The best-performed request update time should be more than 500s, and the most appropriate new DRT call constant should be around -30. More accurate best request update time and new DRT call constant should be determined through more simulation.

The framework of ATOD system enables people to explore the influence of specific parameters on the transport performance, fleet size, vehicle occupancy and fleet deployment under the constraints such as request accepting constraints, frequently-used vehicle incentive, vehicle capacity constraints. The result proves the successful implementation of the system under some assumptions and simplifications such as, both positioning and repositioning time is ignored in the system, people can only choose DRT for their journey, preference of people on ride sharing is ignored, etc.. The current result shows that under above-mentioned assumption, it is possible to introduce a new ATOD system to satisfy transport demand with relatively high vehicle occupancy and reliable service. The thesis offers a framework to solve the optimization problem with MATSim, and the framework is very extendable and flexible, other optimization with different constraints and objective function can easily adapt to the ATOD system.

7.2 Future work

The thesis offers a simulation framework to solve the optimization problem of ATOD. There are still dozens of future work can be done to further improve the model as well as apply the model to solve existing transport problem and to support policy-making. This thesis is just a start point of exploring the possibilities of MATSim and the potential of AV. It can be imagined that with the realization of AV, there will be increasingly more discussion related to AV from the perspective of transport in the future.

7.2.1 The improvement of the model

First, the dynamic vehicle routing and matching in the simulation result in some variance in sensitivity analysis. Further exploration is needed to figure out what influences vehicle occupancy and how to reduce the variances in sensitivity analysis.

Second, the relationship of request update time and vehicle occupancy is not very clear. Although some explanations are given in Section 6, more simulation and analysis are needed to confirm the explanations. If scenarios with more request update time are simulated in the future, the curve of vehicle occupancy with the changing request update time can be determined. In addition, the curve of scheduled wait time and wait after accept time for every iteration should be determined to observe the influence of increasing request update time in details.

It is also important to input the output fleet size and deployment strategies into the simulation with other 10% of the total population for validation. The validation result can test whether the model is overfitting.

As vehicle capacity in the model limits the vehicle occupancy, it is interesting to simulate the model with various vehicle size to reach a better performance in vehicle occupancy. Besides, different operators compete with different vehicle size will also be a good next step. The model can be modified to analyze ATOD from operator's perspective

7.2.2 The application of the model

Despite that ATOD is not launching yet, it is still possible to apply this model to improve the existing public transit system. It will be super interesting to compare the performance of ATOD with PT, from both operator and passenger perspective. Same scenarios and configuration in the thesis can be repeated with only PT mode. The result can show us whether the introduction of future ATOD really can improve the passengers' experience of public transit service as well as reduce the operation cost. In addition, It is also interesting to have a look at the final routing of each minibuses and compare it with the public transit lines. Probably the routing from ATOD can inspire policy makers and planners for a better public transport network design.

Besides, although Sioux Falls is a completed and realistic scenario, more complicated and huge scenarios, such as Zurich, Singapore, etc., are still needed to be tested for a more meaningful result. Then the result can be located on a real map and compare with site analysis. The comparison result can be a good guide and reference for authorities to evaluate both current and future transport situations. Furthermore, more detailed policies regarding AV and PT in cities such as Zurich and Singapore can be implemented into the ATOD system for a more reasonable result and a more complicated analysis.

Acknowledgement

This thesis is written under the supervision of Engaging Mobility group of Future cities laboratory and Institut für Verkehrsplanung und Transportsysteme in ETH Zurich. I want to express my sincere gratitude to Professor Dr. Kay Axhausen, Dr. Pieter Fourie, Dr. Sergio Ordonez and Sebastian Hörl for supervising me on such interesting project and offering me the opportunity to have such an exciting and fulfilling experience in Future Cities Lab.

Professor Dr. Kay Axhausen inspires me with many interesting ideas during the meeting and helps me a lot with my thesis writing, Dr. Pieter Fourie offers me many useful suggestions on both coding and writing, Dr. Sergio Ordonez always encourages me and sits together with me to solve every problem I have, and Sebastian Hörl supports me a lot for my technical and programming issues. Besides, I also want to thank Prof. Kai Nagel for offering me a lot of information on existing capability of MATSim and giving me useful suggestions.

8 References

Azevedo, C. L., K. Marczuk, S. Raveau, H. Soh, M. Adnan, K. Basak, H. Loganathan, N. Deshmunkh, D.-H. Lee, E. Frazzoli et al. (2016) Microsimulation of demand and supply of autonomous mobility on demand, *Transportation Research Record: Journal of the Transportation Research Board*, (2564) 21–30.

Bischoff, J. and M. Maciejewski (2016) Simulation of city-wide replacement of private cars with autonomous taxis in berlin, *Procedia computer science*, **83**, 237–244.

Bischoff, J., N. Soeffker and M. Maciejewski (2016) A framework for agent based simulation of demand responsive transport systems.

Boesch, P. M., F. Ciari and K. W. Axhausen (2016) Autonomous vehicle fleet sizes required to serve different levels of demand, *Transportation Research Record: Journal of the Transportation Research Board*, (2542) 111–119.

Bösch, P. M., F. Ciari and K. W. Axhausen (2017) Transport policy optimization with avs, *Arbeitsberichte Verkehrs-und Raumplanung*, **1269**.

Brownell, C. and A. Kornhauser (2014) A driverless alternative: fleet size and cost requirements for a statewide autonomous taxi network in new jersey, *Transportation Research Record: Journal of the Transportation Research Board*, (2416) 73–81.

Carlino, D., M. Depinet, P. Khandelwal and P. Stone (2012) Approximately orchestrated routing and transportation analyzer: Large-scale traffic simulation for autonomous vehicles, paper presented at the *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, 334–339.

Chebbi, O. and J. Chaouachi (2015) Modeling on-demand transit transportation system using an agent-based approach, paper presented at the *IFIP International Conference on Computer Information Systems and Industrial Management*, 316–326.

Fagnant, D. J. and K. Kockelman (2015a) Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations, *Transportation Research Part A: Policy and Practice*, **77**, 167–181.

Fagnant, D. J. and K. M. Kockelman (2014) The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios, *Transportation Research Part C: Emerging Technologies*, **40**, 1–13.

Fagnant, D. J. and K. M. Kockelman (2015b) Dynamic ride-sharing and optimal fleet sizing for a system of shared autonomous vehicles, paper presented at the *Transportation Research Board 94th Annual Meeting*, no. 15-1962.

Fagnant, D. J., K. M. Kockelman and P. Bansal (2015) Operations of shared autonomous vehicle fleet for austin, texas, market, *Transportation Research Record: Journal of the Transportation Research Board*, (2536) 98–106.

Hörl, S., C. Ruch, F. Becker, E. Frazzoli and K. W. Axhausen (2017) Fleet control algorithms for automated mobility: A simulation assessment for zurich, *Arbeitsberichte Verkehrs-und Raumplanung*, **1270**.

Horni, A., K. Nagel and K. W. Axhausen (2016) *The multi-agent transport simulation MATSim*, Ubiquity Press London.

Levin, M. W., T. Li, S. D. Boyles and K. M. Kockelman (2016) A general framework for modeling shared autonomous vehicles, paper presented at the *95th Annual Meeting of the Transportation Research Board*.

Li, J., Y. S. Chen, H. Li, I. Andreasson and H. van Zuylen (2010) Optimizing the fleet size of a personal rapid transit system: A case study in port of rotterdam, paper presented at the *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, 301–305.

Liu, J., K. M. Kockelman, P. M. Boesch and F. Ciari (2017a) Tracking a system of shared autonomous vehicles across the austin, texas network using agent-based simulation, *Transportation*, **44** (6) 1261–1278.

Liu, X., Y. Yang, M. Meng and A. Rau (2017b) Impact of different bus stop designs on bus operating time components, *Journal of Public Transportation*, **20** (1) 6.

Maciejewski, M. and J. Bischoff (2017) Congestion effects of autonomous taxi fleets, *Transport*, 1–10.

Maciejewski, M. and K. Nagel (2011) Towards multi-agent simulation of the dynamic vehicle routing problem in matsim, paper presented at the *International Conference on Parallel Processing and Applied Mathematics*, 551–560.

Ordóñez Medina, S. A. (2016) New dynamic events-based public transport router, in *The Multi-Agent Transport Simulation MATSim*, 123–132, Ubiquity Press.

SAE, O.-R. A. V. S. C. (2014) Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems, *SAE Standard J3016*, 01–16.

Spieser, K., S. Samaranayake, W. Gruel and E. Frazzoli (2016) Shared-vehicle mobility-on-demand systems: a fleet operator?s guide to rebalancing empty vehicles, paper presented at the *Transportation Research Board 95th Annual Meeting*, no. 16-5987.

Sun, Z., G. Bebis and R. Miller (2006) On-road vehicle detection: A review, *IEEE transactions on pattern analysis and machine intelligence*, **28** (5) 694–711.

Wardman, M., V. P. K. Chintakayala and G. de Jong (2016) Values of travel time in europe: Review and meta-analysis, *Transportation Research Part A: Policy and Practice*, **94**, 93–111.

Zhu, S. and A. L. Kornhauser (2017) The interplay between fleet size, level-of-service and empty vehicle repositioning strategies in large-scale, shared-ride autonomous taxi mobility-on-demand scenarios.

A Configuration of basic scenario

```
<?xml version="1.0"?>

<!DOCTYPE config SYSTEM "http://www.matsim.org/files/dtd/config_v2.dtd">

-<config>


-<module name="drt">

<!-- If true, the startLink is changed to last link in the current
  schedule, so the taxi starts the next day at the link where it stopped
  operating the day before. False by default. -->


<param name="changeStartLinkToLastLinkInSchedule" value="false"/>

<!-- Beeline distance factor for DRT. Used in analyis and in plans file.
  The default value is 1.3. -->


<param name="estimatedBeelineDistanceFactor" value="1.3"/>

<!-- Beeline-speed estimate for DRT. Used in analysis, optimisation
  constraints and in plans file, [m/s]. The default value is 25 km/h -->


<param name="estimatedDrtSpeed" value="8.333333333333334"/>

<!-- Defines the slope of the maxTravelTime estimation function
  (optimisation constraint), i.e. maxTravelTimeAlpha *
  estimated_drt_travel_time + maxTravelTimeBeta. Alpha should not be
  smaller than 1. -->


<param name="maxTravelTimeAlpha" value="1.5"/>

<!-- Defines the shift of the maxTravelTime estimation function
  (optimisation constraint), i.e. maxTravelTimeAlpha *
  estimated_drt_travel_time + maxTravelTimeBeta. Beta should not be
  smaller than 0. -->


<param name="maxTravelTimeBeta" value="600.0"/>
```

```xml
<!-- Max travel time from vehicle to drt passenger (optimisation
  constraint). -->

<param name="maxWaitTime" value="600.0"/>

<!-- Maximum walk distance to next stop location in stationbased system.
  -->

<param name="maxWalkDistance" value="10000000000.0"/>

<!-- Number of threads used for parallel evaluation of request insertion
  into existing schedules. If unset, the number of threads is equal to the
  number of logical cores available to JVM. -->

<param name="numberOfThreads" value="15"/>

<!-- Operational Scheme, either door2door or stationbased. door2door by
  default -->

<param name="operationalScheme" value="stationbased"/>

<!-- Bus stop dwell time per passenger. -->

<param name="stopDurationBeta" value="2.0"/>

<!-- Bus stop acceleration and deceleration time. -->

<param name="stopDurationConstant" value="10.0"/>

<!-- Stop locations file (transit schedule format, but without lines) for
  DRT stops. Used only for the stationbased mode -->

<param name="transitStopFile" value="schedule.xml"/>

<!-- Choose whether input vehicles from files -->

<param name="inputVehicleFile" value="false"/>
```

```xml
<!-- If generate vehicles in the simulation, please input the capacity of
  vehicles -->

<param name="capacity" value="8"/>

<!-- An XML file specifying the vehicle fleet. The file format according
  to dvrp_vehicles_v1.dtd -->

<param name="vehiclesFile" value="null"/>

<!-- Writes out detailed DRT customer stats in each iteration. True by
  default. -->

<param name="writeDetailedCustomerStats" value="true"/>

<!-- Writes out detailed vehicle stats in each iteration. Creates one file
  per vehicle and iteration. False by default. -->

<param name="writeDetailedVehicleStats" value="false"/>

<param name="initialFleetSize" value="0"/>

<!-- Maximum extra waiting time for passenger whose request is accepted.
  Max waiting time equals detouIdx + maxWaitTime-->

<param name="detourIdx" value="300.0"/>

<!-- request update time interval-->

<param name="requestUpdateTime" value="30"/>

<!-- If a vehicle is idle for more than killing time, it will disappear
  from the system-->

<param name="killingTime" value="1800"/>

<!-- If a passenger waits for more than 2 hours, it will be labeled as
  abort...-->
```

```
<param name="abortTime" value="3600"/>

</module>

-<module name="dvrp">

<!-- Mode which will be handled by PassengerEngine and VrpOptimizer
  (passengers'/customers' perspective) -->

<param name="mode" value="drt"/>

<!-- Mode of which the network will be used for routing vehicles,
  calculating trave times, etc. (fleet operator's perspective). If null,
  no mode filtering is done; the standard network (Scenario.getNetwork())
  is used -->

<param name="networkMode" value="null"/>

<!-- Used for estimation of travel times for VrpOptimizer by means of the
  exponential moving average. The weighting decrease, alpha, must be in
  (0,1]. We suggest small values of alpha, e.g. 0.05. The averaging starts
  from the initial travel time estimates. If not provided, the free-speed
  TTs is used as the initial estimates For more info see comments in:
  VrpTravelTimeEstimator, VrpTravelTimeModules, DvrpModule. -->

<param name="travelTimeEstimationAlpha" value="0.05"/>

</module>

-<module name="controler">

<param name="outputDirectory"
  value="output/walkScoreLinear800_withAnnealing_requestUpdate30_noRideSharingBonus_detc

<param name="firstIteration" value="0"/>

<param name="lastIteration" value="100"/>

<param name="eventsFileFormat" value="xml"/>
```

```
<param name="mobsim" value="qsim"/>

<param name="overwriteFiles" value="deleteDirectoryIfExists"/>

<param name="writeEventsInterval" value="10"/>

<param name="writePlansInterval" value="10"/>

</module>


-<module name="plans">

<param name="inputPlansFile" value="population_10prct_90drt_new.xml.gz"/>

</module>


-<module name="network">

<param name="inputNetworkFile" value="network.xml"/>

</module>


-<module name="global">

<param name="numberOfThreads" value="15"/>

</module>


-<module name="qsim">

<param name="startTime" value="00:00:00"/>

<param name="endTime" value="24:00:00"/>

<param name="simStarttimeInterpretation" value="onlyUseStarttime"/>

<param name="flowCapacityFactor" value="0.1"/>

<param name="storageCapacityFactor" value="0.3"/>

</module>
```

```
-<module name="planCalcScore">
<param name="writeExperiencedPlans" value="true"/>

-<parameterset type="scoringParameters">
<param name="marginalUtilityOfMoney" value="1.0"/>
<param name="performing" value="6.0"/>
<param name="utilityOfLineSwitch" value="-1.0"/>
<param name="waiting" value="-6.0"/>
<param name="waitingPt" value="-6.0"/>

-<parameterset type="activityParams">
<param name="activityType" value="home"/>
<param name="typicalDuration" value="08:00:00"/>
</parameterset>

-<parameterset type="activityParams">
<param name="activityType" value="work"/>
<param name="typicalDuration" value="09:00:00"/>
</parameterset>

-<parameterset type="activityParams">
<param name="activityType" value="secondary"/>
<param name="typicalDuration" value="01:00:00"/>
</parameterset>
```

```
-<parameterset type="modeParams">

<param name="constant" value="-1.0"/>

<param name="marginalUtilityOfDistance_util_m" value="0.0"/>

<param name="marginalUtilityOfTraveling_util_hr" value="-4.0"/>

<param name="mode" value="drt"/>

<param name="monetaryDistanceRate" value="0.0"/>

</parameterset>


-<parameterset type="modeParams">

<param name="constant" value="-1.0"/>

<param name="marginalUtilityOfDistance_util_m" value="0.0"/>

<param name="marginalUtilityOfTraveling_util_hr" value="-4.0"/>

<param name="mode" value="pt"/>

<param name="monetaryDistanceRate" value="0.0"/>

</parameterset>


-<parameterset type="modeParams">

<param name="constant" value="-1.0"/>

<param name="marginalUtilityOfDistance_util_m" value="0.0"/>

<param name="marginalUtilityOfTraveling_util_hr" value="-4.0"/>

<param name="mode" value="car"/>

<param name="monetaryDistanceRate" value="0.0"/>

</parameterset>
```

```
-<parameterset type="modeParams">
<param name="constant" value="-30.0"/>
<param name="marginalUtilityOfDistance_util_m" value="0.0"/>
<param name="marginalUtilityOfTraveling_util_hr" value="-4.0"/>
<param name="mode" value="drt creation"/>
<param name="monetaryDistanceRate" value="0.0"/>
</parameterset>

-<parameterset type="modeParams">
<param name="constant" value="-1.0"/>
<param name="marginalUtilityOfDistance_util_m" value="0.0"/>
<param name="marginalUtilityOfTraveling_util_hr" value="-5.8"/>
<param name="mode" value="walk"/>
<param name="monetaryDistanceRate" value="0.0"/>
</parameterset>
</parameterset>
</module>

-<module name="strategy">
<param name="maxAgentPlanMemorySize" value="4"/>
<!-- 0 means unlimited -->

<param name="fractionOfIterationsToDisableInnovation" value="1"/>

-<parameterset type="strategysettings">
```

```
<!-- iteration after which strategy will be disabled. most useful for
  ``innovative'' strategies (new routes, new times, ...). Normally, better
  use fractionOfIterationsToDisableInnovation -->

<param name="disableAfterIteration" value="-1"/>

<!-- strategyName of strategy. Possible default names:
  SelectRandomBestScoreKeepLastSelectedChangeExpBetaSelectExpBetaSelectPathSizeLogit
  (selectors), ReRoute TimeAllocationMutator ChangeLegMode
  TimeAllocationMutator_ReRoute ChangeSingleLegMode ChangeSingleTripMode
  SubtourModeChoice ChangeTripMode TripSubtourModeChoice (innovative
  strategies). -->

<param name="strategyName" value="SelectExpBeta"/>

<!-- weight of a strategy: for each agent, a strategy will be selected
  with a probability proportional to its weight -->

<param name="weight" value="0.9"/>

</parameterset>

-<parameterset type="strategysettings">

<!-- iteration after which strategy will be disabled. most useful for
  ``innovative'' strategies (new routes, new times, ...). Normally, better
  use fractionOfIterationsToDisableInnovation -->

<param name="disableAfterIteration" value="60"/>

<!-- strategyName of strategy. Possible default names:
  SelectRandomBestScoreKeepLastSelectedChangeExpBetaSelectExpBetaSelectPathSizeLogit
  (selectors), ReRoute TimeAllocationMutator ChangeLegMode
  TimeAllocationMutator_ReRoute ChangeSingleLegMode ChangeSingleTripMode
  SubtourModeChoice ChangeTripMode TripSubtourModeChoice (innovative
  strategies). -->

<param name="strategyName" value="ChangeSingleTripMode"/>

<!-- weight of a strategy: for each agent, a strategy will be selected
```

```
  with a probability proportional to its weight -->

<param name="weight" value="0.027"/>
</parameterset>

-<parameterset type="strategysettings">
<!-- iteration after which strategy will be disabled. most useful for
  ``innovative'' strategies (new routes, new times, ...). Normally, better
  use fractionOfIterationsToDisableInnovation -->

<param name="disableAfterIteration" value="70"/>
<!-- strategyName of strategy. Possible default names:
  SelectRandomBestScoreKeepLastSelectedChangeExpBetaSelectExpBetaSelectPathSizeLogit
  (selectors), ReRoute TimeAllocationMutator ChangeLegMode
  TimeAllocationMutator_ReRoute ChangeSingleLegMode ChangeSingleTripMode
  SubtourModeChoice ChangeTripMode TripSubtourModeChoice (innovative
  strategies). -->

<param name="strategyName" value="ChangeSingleTripMode"/>
<!-- weight of a strategy: for each agent, a strategy will be selected
  with a probability proportional to its weight -->

<param name="weight" value="0.026"/>
</parameterset>

-<parameterset type="strategysettings">
<!-- iteration after which strategy will be disabled. most useful for
  ``innovative'' strategies (new routes, new times, ...). Normally, better
  use fractionOfIterationsToDisableInnovation -->

<param name="disableAfterIteration" value="80"/>
<!-- strategyName of strategy. Possible default names:
```

```
  SelectRandomBestScoreKeepLastSelectedChangeExpBetaSelectExpBetaSelectPathSizeLogit
  (selectors), ReRoute TimeAllocationMutator ChangeLegMode
  TimeAllocationMutator_ReRoute ChangeSingleLegMode ChangeSingleTripMode
  SubtourModeChoice ChangeTripMode TripSubtourModeChoice (innovative
  strategies). -->

<param name="strategyName" value="ChangeSingleTripMode"/>

<!-- weight of a strategy: for each agent, a strategy will be selected
  with a probability proportional to its weight -->

<param name="weight" value="0.024"/>

</parameterset>

-<parameterset type="strategysettings">

<!-- iteration after which strategy will be disabled. most useful for
  ``innovative'' strategies (new routes, new times, ...). Normally, better
  use fractionOfIterationsToDisableInnovation -->

<param name="disableAfterIteration" value="90"/>

<!-- strategyName of strategy. Possible default names:
  SelectRandomBestScoreKeepLastSelectedChangeExpBetaSelectExpBetaSelectPathSizeLogit
  (selectors), ReRoute TimeAllocationMutator ChangeLegMode
  TimeAllocationMutator_ReRoute ChangeSingleLegMode ChangeSingleTripMode
  SubtourModeChoice ChangeTripMode TripSubtourModeChoice (innovative
  strategies). -->

<param name="strategyName" value="ChangeSingleTripMode"/>

<!-- weight of a strategy: for each agent, a strategy will be selected
  with a probability proportional to its weight -->

<param name="weight" value="0.023"/>

</parameterset>

</module>
```

```
-<module name="facilities">

<param name="inputFacilitiesFile" value="Siouxfalls_facilities.xml"/>

</module>


-<module name="changeMode">

<!-- Defines all the modes available, including chain-based modes,
  seperated by commas -->


<param name="modes" value="drt, drt creation"/>

</module>

</config>
```